Ivan Begtin



Data Catalogs

How open data portals, geoportals and repositories work, and how to choose your own

October 2026 edition

Data Catalogs. How open data portals, geoportals and repositories work, and how to choose your own

© 2026 Ivan Begtin. October 2026 edition.

The text is available under the Creative Commons Attribution 4.0 International licence (CC BY 4.0): creativecommons.org/licenses/by/4.0.



The book and the data catalog knowledge base are supported by Dateno, a search engine for datasets.

Online edition: https://datacatalogs.begtin.tech/en/

Source text: https://github.com/ivbeg/data-catalogs-book

PDF built automatically on 23 October 2026 from source revision 2b56130. The latest edition and the interactive route map are on the book's website.

Contents

About This Book 6

I. What Is a Data Catalog 12

What Is a Data Catalog 13

Why Catalogs Exist and Who Uses Them 16

Anatomy of a Catalog 19

The World of Catalogs in Numbers 27

II. What Data Gets Published 31

What Catalogs Publish 32

Units of Publication and Formats 40

III. Types of Catalogs 44

How to Classify Catalogs 45

Open Data Portals 56

Geoportals 66

Research Data Repositories 79

Indicator Catalogs 92

Microdata Catalogs 105

ML Catalogs 118

Metadata Catalogs and Registries 130

Search Engines and Aggregators 144

API Catalogs, Marketplaces and Dataset Lists 158

Comparing Types and Hybrids 171

IV. Catalog Capabilities 182

Capability Map 183

Metadata Export 193

Metadata Completeness and Depth 203

APIs and Bulk Download 215

Built-in Storage 229

Persistent Identifiers, DOIs and Versions 241

Citation Export 254

Access, Roles and Workflows 266

Deployment Speed and Maintenance 282

Other Capabilities 286

V. Platforms and Extensions 296
Classes of Software 297
Extension Ecosystems 300
CKAN: Core and Extensions 307
Platform Profiles 311

VI. Choosing a Catalog 329
Decision Route Map 330
Selecting Features 335
From Features to a Platform 343
Justifying a Decision 347
Build, Open Source or SaaS 355
Typical Scenarios 361
Requirements Checklist 370
Worked Examples 373

VII. Publishing Data 380
The Publishing Process 381
Metadata Standards 388
Federation and Harvesting 397
Licences and Access Conditions 402
Discoverability 411

VIII. Publishing Without a Portal 416
When You Don't Need a Portal 417
Static Sites 424
Git Repositories 432
Cloud Storage and Shared Hubs 440
Making a Publication Findable 448

IX. Practices 456
Good Practices 457
Anti-patterns 463
Catalog Lifecycle 468
Archiving and Rescuing Catalogs 474

X. Catalog Quality 479
Quality Levels 480
Quality Metrics 485
Methodologies and Rankings 491
Catalog Self-Audit 498

XI. Trends 505

AI, Data Spaces and Regulation 506

What Innovative Catalogs Can Already Do 516

Appendices 525

Glossary 526

Using the Registry 543

Platform Comparison Tables 567

Checklists 572

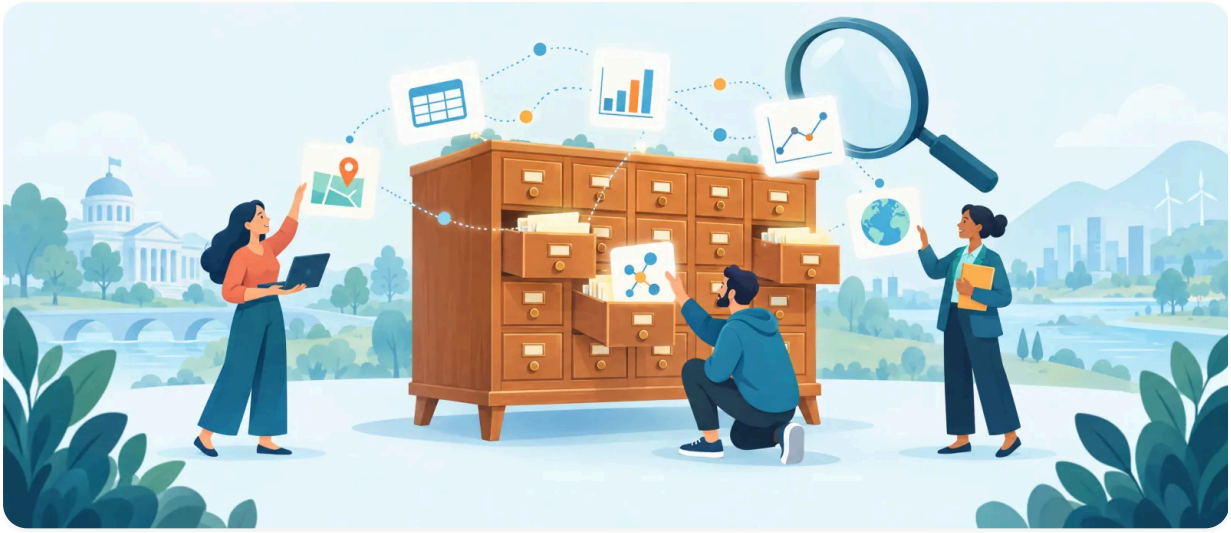
Bibliography and Links 580

Stable Addresses for Links 585

About the Author 586

About This Book

How open data portals, geoportals and repositories work, and how to choose your own.





Author: Ivan Begtin, founder and director of the non-profit Infoculture ("Information Culture"), founder and CTO of the Dateno dataset search engine. He has worked on open data since 2009 and on catalogs since 2010: from the community Hub of Open Data to the worldwide catalog registry from which this book's figures are calculated. More in the chapter "About the Author".

This book and the data catalog knowledge base are supported by Dateno, a search engine for datasets. Dateno collects dataset descriptions from thousands of catalogs around the world (open data portals, geoportals, research repositories and statistical offices) and lets you search them in one place. As of 16 October 2026 it holds 22 million datasets from 5,236 sources.

October 2026 edition. The text is available under a CC BY 4.0 licence. This is the English translation of the Russian original.

Download the book as PDF: in English · in Russian. The Russian version of the site is in the language switcher in the menu.

A **data catalog** is a service that describes datasets and helps people find, understand and obtain them. This book is about public catalogs: open data portals, geoportals, research repositories, statistics catalogs and machine learning dataset catalogs. It explains what kinds of catalogs exist and what they can do, how to choose features and a platform and justify the choice, and how to publish data, on a portal or without one.

Enterprise catalogs that describe data inside a company (Collibra, Alation, DataHub) are mentioned only for comparison.

If you are looking for data rather than a catalog, start by searching Dateno. How it relates to the book's figures is explained in "Where the numbers come from".

Where to start

You do not need to read the book in order. Pick your task: under it are 4–6 chapters in reading order and what you will have at the end. If none of the tasks fits, go through the route map: eight questions, about five minutes.

You have been asked to launch an open data portal: a government body, region or city · 6 chapters, about 1 h 15 min

1. Decision Route Map: a first pointer to the catalog type and platform.
2. Typical Scenarios: find your row: central government, region or municipality.
3. When You Don't Need a Portal: perhaps a regional or national portal already covers the task.
4. Selecting Features: a feature list built from tasks, not from marketing.
5. Requirements Checklist: what to put in the specification.
6. Justifying a Decision: a decision record for your manager and for the next person in the job.

At the end you have: an answer to whether you need your own portal, a feature list for the specification and a decision record.

Choose a platform or write a specification · 6 chapters, about 1 h 45 min

1. Capability Map: which capabilities exist and what is typical for your catalog type.
2. Selecting Features: every feature gets five notes: whose task it is, how mandatory, why, how to check it, when to review.
3. From Features to a Platform: how to turn a feature list into a platform shortlist.
4. Extension Ecosystems: how to check an extension before the platform gets on the list.
5. Platform Profiles: what each platform has in the core, what in extensions, and its limitations.
6. Requirements Checklist.

At the end you have: a platform shortlist marked "core", "extension" and "no", and requirements for the specification.

Publish data for a paper and get a DOI · 5 chapters, about 1 h 35 min

1. Research Data Repositories: Zenodo, a discipline repository or a university repository.
2. Persistent Identifiers, DOIs and Versions: how a version DOI differs from a concept DOI.
3. Citation Export: how the dataset will be cited and why you need a `CITATION.cff` file.
4. Licences and Access Conditions: which licence to choose and what it allows.
5. Making a Publication Findable: so that search engines and aggregators find the dataset.

At the end you have: a chosen repository, a DOI, a licence and a citation file.

Publish a few datasets without your own portal · 4 chapters, about 50 min

1. When You Don't Need a Portal: five questions before installing anything.
2. Static Sites or Git Repositories: choose one approach.
3. Making a Publication Findable: a dataset page, a description file and links from other catalogs.
4. The "Minimum publication without a portal" checklist and file templates: README, `datapackage.json`, `CITATION.cff`, schema.org markup.

At the end you have: a page or repository with dataset descriptions that search engines and aggregators can find.

Collect descriptions from many catalogs: a search engine or aggregator developer · 5 chapters, about 1 h 50 min

1. Dataset Search Engines and Aggregators: how a search engine differs from an aggregator.
2. Metadata Export: DCAT, schema.org and what platforms actually serve.
3. APIs and Bulk Download: a full dump versus a search API.
4. Federation and Harvesting: how descriptions are collected and where records get lost.
5. Metadata Standards: which description language each recipient reads.

At the end you have: which protocols and formats to take from each catalog type and what losses to expect.

A statistics office: indicators and microdata · 5 chapters, about 2 h

1. The worked example of a statistics office: why these are two catalogs, not one.
2. Indicator Catalogs: time series, SDMX, PxWeb and .Stat Suite.
3. Microdata Catalogs: anonymised records, DDI, NADA and access on request.
4. Platform Profiles: PxWeb, .Stat Suite and NADA.
5. The checklists for an indicator catalog and a microdata catalog.

At the end you have: two catalogs with different standards and a platform shortlist for each.

Check and improve your own catalog · 4 chapters, about 45 min

1. Quality Levels: which level an assessment touches and who is responsible for it.
2. Catalog Self-Audit: how to go through your catalog in one pass.
3. Anti-patterns: typical failures and where they come from.
4. Good Practices: six things without which a dataset cannot be found or used.

To check a live site, use the live portal checklist.

At the end you have: a list of problems by quality level and the order in which to fix them.

The book in nine theses

1. **A catalog is descriptions, not a file store.** Without a description, a link to a file is not yet a catalog. → What Is a Data Catalog
2. **The catalog type sets what is published in it.** The type determines the description standard and the range of suitable platforms. → How to Classify Catalogs
3. **You do not always need your own portal.** While there are few datasets and publishers, data is published in someone else's catalog, on your own site, in a repository or on a shared hub. → When You Don't Need a Portal
4. **Features are chosen by task**, not by a platform's marketing or someone else's specification. → Selecting Features
5. **Every feature has an address:** the platform core, a named extension, or nowhere. If a mandatory feature depends on an extension that is not tested on your core version, the platform is struck off. → Extension Ecosystems
6. **The build model is chosen before the brand.** A platform is useless if nobody can update it. → Build, Open Source or SaaS
7. **The decision is written down.** A decision record explains to the next person in the job which commitment the platform fulfils and when to review the choice. → Justifying a Decision
8. **A dataset is published when an outsider can use it:** they find the page, open the file, read the licence and know who is responsible for the data. → The Publishing Process
9. **Quality has four levels.** A high score on one does not fix the others. → Quality Levels

Contents

Times are estimates for reading straight through, without code examples. About 17 hours in total.

- **I. What Is a Data Catalog** · 4 chapters, 30 min: the definition, how a catalog is built, the world of catalogs in numbers.
- **II. What Data Gets Published** · 2 chapters, 25 min: what is published, units of publication and formats.
- **III. Types of Catalogs** · 11 chapters, 4 h 30 min: open data portals, geoportals, research repositories, statistics, ML, metadata, aggregators and hybrids.
- **IV. Catalog Capabilities** · 10 chapters, 3 h 40 min: metadata export, APIs, storage, DOIs, citation, access and roles.
- **V. Platforms and Extensions** · 4 chapters, 1 h 10 min: software classes, core and extensions, CKAN, platform profiles.
- **VI. Choosing a Catalog** · 8 chapters, 1 h 40 min: the route map, choosing features and a platform, justification, scenarios and examples.
- **VII. Publishing Data** · 5 chapters, 1 h 10 min: the process, metadata standards, federation, licences, discoverability.
- **VIII. Publishing Without a Portal** · 5 chapters, 1 h: static sites, Git, cloud storage and hubs.

- **IX. Practices** · 4 chapters, 50 min: good practices, anti-patterns, lifecycle, archiving.
- **X. Catalog Quality** · 4 chapters, 50 min: levels, metrics, methodologies and indices, self-audit.
- **XI. Trends** · 2 chapters, 40 min: AI, data spaces and regulation; what leading catalogs can already do.
- **Appendices** · reference: glossary, using the registry, platform tables, checklists, bibliography.
- **About the Author**: who wrote the book and how his projects relate to its figures.

How chapters are built

- **Conclusion first.** A chapter opens with its main conclusion; details and examples follow.
- **Core, extension or no.** In the chapters on capabilities and in the platform profiles, every feature says where it lives: in the platform itself, in a separate module with its own author and releases, or nowhere.
- **Decision record**: one or two pages about a choice: context, options, criteria, decision, rationale, when to review. The template is in the chapter "Justifying a Decision".
- **Dates on facts.** Software versions and the state of websites change, so facts are recorded "as of October 2026". The last section of every chapter is "Sources".
- **An unfamiliar word**: look it up in the glossary.

Ready-made materials: checklists, platform comparison tables, file templates and catalog registry queries.

Where the numbers come from

Figures on the number of catalogs and platforms were calculated from the open dataportals-registry of data catalogs, release 1.27.0 of 15 October 2026. It describes 43,614 catalogs and 790 platforms. The main queries behind the book's figures are collected in the appendix on the registry: on this release they give the same numbers. The registry is updated every few days, and newer releases will give different numbers.

The registry was originally built for the Dateno dataset search engine, and Dateno still indexes portals from their registry descriptions: from a record it learns the catalog's platform and API addresses, and from those which harvester to use for collecting datasets. As of 16 October 2026 Dateno holds 22 million datasets from 5,236 sources. Why some catalogs are easy to harvest while the harvester never reaches others is explained in the chapters "Search Engines and Aggregators" and "Discoverability".

Software versions and platform capabilities were checked against primary sources in October 2026.

About this edition

Author: Ivan Begtin. **Edition:** October 2026. This English edition is a translation of the Russian original.

Support. The book and the knowledge base are supported by the Dateno project, a search engine for datasets. The author is the founder and CTO of Dateno; how this affects the book's sources is explained in the chapter "About the Author".

Licence. The text is distributed under the Creative Commons Attribution 4.0 International (CC BY 4.0) licence. You may copy it, translate it and use it in your own materials, including commercial ones, as long as you credit the author and link to the book.

How to cite.

> Begtin I. Data Catalogs: How Open Data Portals, Geoportals and Repositories Work, and How to Choose Your Own. October 2026 edition. URL: https://datacatalogs.begtin.tech/en/ (accessed: DD Month YYYY).

PART I.

What Is a Data Catalog

Definitions, the boundaries of the concept, who uses catalogs, how a catalog is built, and the overall picture from the registry data.

1. What Is a Data Catalog
2. Why Catalogs Exist and Who Uses Them
3. Anatomy of a Catalog
4. The World of Catalogs in Numbers

What Is a Data Catalog

A data catalog is a service that describes datasets and helps people find, understand and obtain them. The description is called metadata: the title, who published it, the licence, the time period, the format, the address of the file or service.

The file itself may sit in the catalog or be linked from somewhere else. This is the main technical distinction, and the book returns to it in the chapter on storage. Without a description, a link to a file is not yet a catalog.

Portal, repository, aggregator

Related words are often confused.

A **portal** is a catalog plus everything around it: news, reuse examples, discussions, a feedback form. data.gouv.fr is a portal: a dataset has a page with reuse examples and a discussion. A page listing ten CSV files is not one yet, even though it also publishes data. When such a list is enough is covered in the part Publishing Without a Portal.

A **repository** is a catalog that stores the files itself and usually issues a persistent identifier so that the data can be cited in a paper. Zenodo is a typical example.

A **geoportal** is a catalog of layers and map services, often together with a map in the browser.

An **aggregator or dataset search engine** is a catalog of catalogs. It does not publish datasets of its own but collects descriptions from others. Google Dataset Search and Dateno work this way.

A **metadata registry** describes structures rather than datasets: vocabularies, schemas, other catalogs. It belongs on a different shelf, even though the word "catalog" may appear in its name.

A **data warehouse or data lake** is where the bytes live inside an organisation. A catalog answers a different question: how to find that data from the outside.

Three big families

Leaving details aside, data catalogs fall into three families. This division was proposed back in 2020, and it still helps you quickly understand what you are dealing with.

	Open data portals	Research repositories	Enterprise catalogs
Who can see it	everyone	everyone, sometimes part of the data on request	employees only
Control over sources	almost none: agencies publish, the catalog accepts what it is given	sets requirements for the description, but not for the data itself	full: the catalog knows every table
What matters most in the description	licence and terms of use	persistent identifier and citation	owner, lineage, business glossary

	Open data portals	Research repositories	Enterprise catalogs
How they connect to neighbours	harvesting descriptions via DCAT, the CKAN API and similar protocols	harvesting via OAI-PMH, issuing DOIs	integration with the company's storage and tools

The book covers the first two families and their relatives: geoportals, indicator and microdata catalogs, ML catalogs. A detailed classification by type is in Part III.

What this book does not cover

Enterprise catalogs such as Collibra, Alation and DataHub manage data inside a company: who owns a table, where it came from, who is allowed to see it. The only thing they share with a public portal is the word "catalog". The audience, openness and standards are different. A public catalog describes files and services that can be downloaded. An enterprise catalog describes database tables, processing pipelines and reports that can only be accessed from the inside. Companies deploy one to find where personal data is stored, to help analysts find the right table and to help engineers understand where its values come from. Such catalogs hardly use common description standards like DCAT: integration with the company's tools matters more to them. The rest of the book does not discuss them.

The book also does not look for a specific dataset such as "the Moscow budget for 2024". For that, you open the portal itself or a dataset search engine. This book is about how such portals work and how to choose your own.

A short history

- 2007: the eight principles of open government data and the EU INSPIRE Directive on spatial data.
- 2009–2010: the national portals data.gov and data.gov.uk; CKAN spreads.
- 2010–2011: civic portals appear next to government ones, built by activists who do not wait for the authorities. In Russia, this is how the Open Data Hub, hubofdata.ru, appeared on CKAN, earlier than the government portals.
- 2012: re3data, the registry of research repositories.
- 2014: the DCAT vocabulary becomes a W3C Recommendation. 2016: the FAIR principles on the findability and reuse of research data.
- 2019: the EU Open Data Directive. 2023: the implementing regulation on high-value datasets, which public bodies must provide free of charge and in machine-readable form.
- 2020s: there are so many catalogs that catalogs of catalogs and general search engines appear. In 2021 the list of Russian catalogs datacatalogs.ru was launched; in 2023, the open registry of catalogs worldwide, dataportals-registry (started within the Common Data Index project); and in 2024, built on it, the dataset search engine Dateno. Alongside them, programs that read catalogs without a human are growing.
- 2026: the flagships of the first wave are changing. data.gov is moving from CKAN to a custom build; data.gov.uk now shows curated collections on its home page instead of harvesting everything, and since the summer the home page has been called the National Data Library.

The previous catalog still works.

The types that grew out of this are covered in Part III.

Sources

- INSPIRE Directive 2007/2/EC
- DCAT 3
- FAIR principles
- re3data
- GDS, What's changing on data.gov.uk and why, 25 March 2026; the data.gov.uk home page in the Web Archive on 1 April 2026, not yet named National Data Library, and on 2 July 2026, already with that name
- I. Begtin, Telegram (in Russian): three categories of data catalogs, 14 October 2020; relaunch of hubofdata.ru, created in 2010–2011, 14 April 2021; hubofdata.ru on CKAN appeared earlier than the government portals, 12 February 2022; launch of datacatalogs.ru, 3 November 2021; Common Data Index and the catalog registry, 10 April 2023; launch of Dateno, 13 March 2024
- dataportals-registry: repository created in March 2023, checked 12 October 2026; Dateno
- I. Begtin, Data discovery in the corporate sector. Part 1 (in Russian), July 2023: why companies deploy data catalogs

Why Catalogs Exist and Who Uses Them

Almost everything else depends on why a catalog was set up and who will use it: the type, the feature set and the platform. A catalog built "to have one" usually becomes a showcase that nobody updates.

Why they are built

- **Law and reporting.** A public body is required to publish open data. A statistical office releases indicators. A geoportal is part of a spatial data infrastructure.
- **Transparency.** Budgets, procurement and elections should be findable without a written request to an office.
- **Grant or journal conditions.** Research data is published with a persistent identifier so that it can be cited.
- **Reuse.** One agency stops begging another for a spreadsheet by email.
- **Sales and services.** The description is open; the file or API is paid or available under a contract.
- **Showcase.** A portal is launched for a date or a project. The motive is not bad in itself, but without updates and machine access it leads to anti-patterns.

If there are two motives, there are often two catalogs. A statistical office needs both indicators and survey files. One site rarely does both jobs well. See the statistical office example.

Three jobs of a portal

The motive answers the question "why was it set up". It also helps to ask what job the portal does for its users. Government open data portals usually have three, and a good portal does at least one of them seriously.

- **Search engine and aggregator.** The portal collects descriptions from dozens or hundreds of systems where the data is actually published and offers a single search. This is how the national portals of large countries work: data.gov harvests the catalogs of federal agencies, and the European data.europa.eu harvests national portals. The Russian data.gov.ru was also conceived as a harvester of lists from agency websites; how that turned out is covered in anti-patterns.
- **Community.** The portal engages users: dataset discussions, reuse examples, data requests. data.gouv.fr is the model. It is not only agencies that publish there: Open Food Facts, a civic project and open food products database, has its own organisation page with datasets.
- **Infrastructure for reference data.** The portal is part of the government's digital infrastructure and serves data that other systems and businesses run on. In France, the Digital Republic Act singled out "reference data": as of 2026 there are nine datasets, among them the national address database, the Sirene business register, the official geographic reference and

the cadastral map. Quality and availability obligations apply to them, and the list is published on data.gouv.fr. In Russia, separate reference data portals play a similar role; they are covered in the chapter on metadata catalogs.

The features depend on which job is the main one. A search engine needs harvesting and good search, a community needs discussions and user accounts, infrastructure needs stable addresses, bulk download and APIs that others can rely on.

Who uses them

- People: residents, journalists, researchers, analysts, staff of other agencies.
- Developers of services and apps.
- Other catalogs that harvest descriptions on a schedule.
- Dataset search engines: Google Dataset Search, Dateno, OpenAIRE.
- Programs and AI agents that look for data through APIs. More on this in the chapter on trends.

A person and a program are different customers. A site without a metadata export loses search engines and higher-level portals. A catalog only for programs, without readable pages, fails the transparency job: a resident is not obliged to read JSON.

What different people expect

The people in the list above expect different things from a catalog. It is convenient to divide them into three main groups and several subgroups. One person may belong to several at once.

Group	What they expect
Analysts	fresh data on the main topics, above all official statistics; familiar formats such as CSV and XLSX; the ability to open the data in their own tools without programming
Researchers	data from their discipline, open or with clear access rules; DOIs for datasets; the formats and tools accepted in their field; a clear provenance and methodology
Developers and data engineers	APIs; bulk download of the whole catalog; schema and structure descriptions; modern formats such as compressed CSV and Parquet; predictable addresses so they can build downloads into their pipelines
Journalists	the same as analysts, but with less technical depth; what matters most is that the data is available at all
ML engineers	the same as developers, plus large volumes for training and links to platforms such as Hugging Face
Statisticians	substantive metadata and specialised standards: SDMX for indicators, DDI for surveys
Geospatial analysts and GIS specialists	search by territory, standard ArcGIS and OGC services, download in common geospatial formats

The table works as a test. Look at your catalog through the eyes of each group: can they solve their task? A portal that is useless to all seven will not be saved by design or by the number of datasets.

Business stands apart. It needs specific data rather than a portal: registers of organisations, procurement, indicators, base maps. Having found the original source once, a company connects to it directly and from then on expects a stable channel and a person to write to if something

breaks. For business it matters more that the agency publishes the needed data at all and answers requests than which portal the data sits on.

What follows for the choice

Write down one main user group and one obligation. Features follow from them. A search engine needs schema.org markup. A higher-level portal needs an export in its DCAT profile. A journal needs a DOI and a citation line. A resident needs a clear title, a licence and a download button.

A feature without such a note is a candidate for rejection. How to record this is described in the chapter on selecting features.

Sources

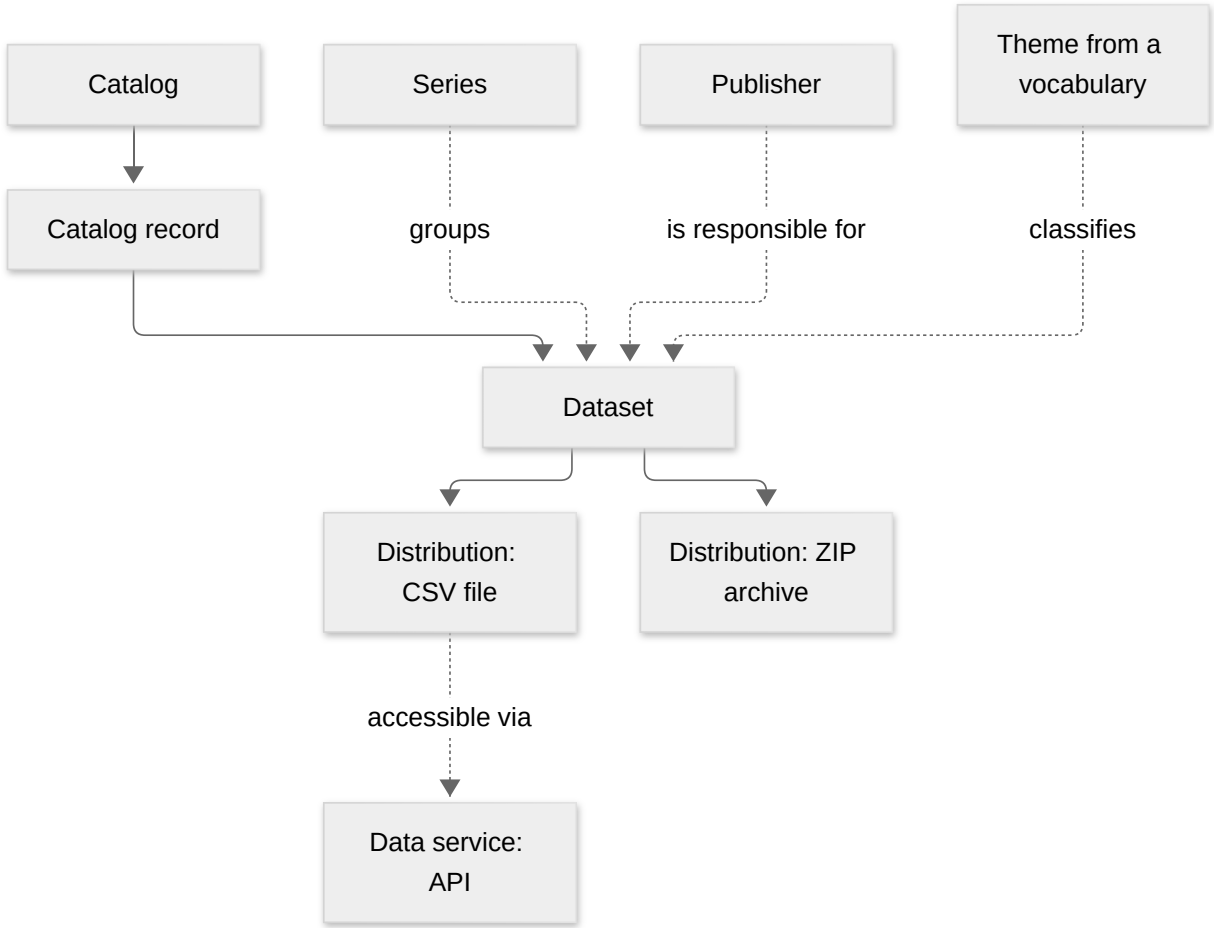
- Capability map
- Selecting features
- data.gouv.fr: Service public de la donnée, the list of nine reference datasets, checked 12 October 2026; Open Food Facts organisation page
- I. Begtin, Telegram (in Russian): three functions of open data portals and their audience, 27 March 2023; three groups of data users and their expectations, 7 December 2025

Anatomy of a Catalog

A CKAN portal, a research repository and a geoportal call their parts by different names, but they are built the same way. A catalog holds records about datasets. A dataset has a description, someone responsible for it and one or more ways to get the data itself. Around the datasets there are organisations, themes and groups; inside the software there is a metadata database, a search index and file storage.

If you can tell these parts apart, you can compare any two catalogs, read their programming interface and see what your own is missing. This chapter describes the structure in the words of DCAT, the W3C vocabulary for describing catalogs. Most open data portals are built on it, and the other standards map onto it easily. At the end, one live dataset page is taken apart: almost all the parts are visible in it, along with the typical breakages between them.

The overall picture



Read the diagram from top to bottom: the catalog keeps records, a record describes a dataset, and a dataset is served in one or more ways. Dashed lines are links that not every catalog has.

Dataset

A **dataset** is what the user sees as one page and uses as one whole: a city budget, a cadastral layer, an indicator series, the data behind a paper. What counts as a dataset is decided by the catalog owner, and the decisions vary a lot: for a central bank a dataset is a time series, in a satellite imagery catalog it is a collection of thousands of files. See the chapter on units of publication.

A dataset page answers six questions. They make it easy to read any dataset page, even in an unfamiliar language.

Question	Fields	What cannot be missing
What is it	title, description, keywords, theme	a title and a clear description
Who is responsible	publisher, contact, author	the publisher and a way to get in touch
What it covers and for what time	period, territory, update frequency	the period, for any data that changes
Under what terms	licence, access restrictions	the licence
How to get it	list of distributions	at least one that works
How to cite it	persistent identifier, version, citation line	a persistent address

Which fields are mandatory for everyone and which are needed only in specific domains is covered in the chapter on metadata completeness.

A theme and a tag are different things. A theme comes from a shared vocabulary: for European portals it is the list of themes of the EU Publications Office, for geoportals it is the ISO 19115 topic categories. A tag is a free word typed in by the publisher. By theme, datasets from different portals line up on one shelf of a shared portal. Tags do not: "transport", "Transport" and "транспорт" are three different words for a program.

A publisher and an organisation in the software are not the same either. The **publisher** is the party responsible for the data in substance and legally. An organisation in the software is an account inside which staff edit datasets. Usually they coincide. They diverge when the portal operator creates datasets on behalf of an agency, or when the catalog harvests other people's descriptions: the organisation is then the "harvester", and the publisher is the original agency.

The dataset record and two pairs of dates

A **catalog record** holds information not about the data but about how the dataset got into the catalog: when it was added and when the description was last edited. DCAT has a separate class for this, and the specification explains it precisely through dates. The dataset's issue date is when the publisher first made the data available. The record's date is when this particular catalog learned about the dataset.

Most software does not distinguish these pairs of dates. Then the "updated" field shows the date of the last edit to the description, not to the data. Fix a typo in the title, and a five-year-old dataset looks fresh. Harvest descriptions from another portal, and every dataset gets today's date. That is why the freshness of data is checked against the file itself and the period field, not the date on the page.

Distribution

A **distribution** is a specific file, link, API or map service through which a dataset is served. DCAT calls it a distribution. One dataset may have several: CSV for a person, Parquet for an analyst, an API for a program.

A distribution has its own fields: format, size, checksum, sometimes its own licence and its own modification date. DCAT draws a fundamental distinction between two links:

- a **download URL** (`downloadURL`): a program gets the data with a single request;
- an **access URL** (`accessURL`): a page or service through which the data can be obtained, though not necessarily with a single request.

If the download URL field holds a page on the agency's website, a person will figure it out, but a program will download HTML instead of a table. This is one of the common reasons why a dataset search engine shows a dataset but cannot determine its format and size.

A **data service** is a programming interface through which a dataset's data is obtained piece by piece: filter rows, request features on a map. In DCAT 2 it became a separate entity: a service has its own address and description, and a distribution refers to it. This makes it convenient to describe a map service that serves a hundred layers, or a table API shared by the whole portal.

Series and versions

Catalogs most often confuse two links between datasets.

A **series** groups datasets of the same kind that are released one after another: budget execution for each month, a census every ten years. A separate class for series appeared only in DCAT 3 (2024), so most portals do not know about series and create each release as a separate dataset page. On the Greek national portal data.gov.gr, in October 2026, state budget execution is 131 separate datasets, one per month since 2014, with no common group. A person finds the whole series only by searching by title. Where the line runs between a series and inflating the counter is covered in the chapter on anti-patterns.

A **version** is a fixed state of a dataset that can be cited. Research repositories need versions without fail: a paper cites the data the authors had, not the latest export. InvenioRDM, which runs Zenodo, has three levels for this. A parent record groups all versions and holds what they share: access rights, community membership. Each version has its own record with its own files and its own identifier. Minor edits to the description within a version are saved as revisions and do not create a new version. Open data portals more often just replace the file, and the previous state disappears. More on this in the chapter on identifiers and versions.

Organisations, groups and the catalog itself

Catalogs build one more level on top of datasets.

- **Organisations** answer the question "who is responsible". In CKAN a dataset usually belongs to exactly one organisation, and editing rights depend on it.

- **Groups, collections, communities** answer the question "what belongs together". In CKAN a dataset can be in several groups. In Dataverse collections nest inside each other: university, faculty, lab. In InvenioRDM a community curates records: it accepts or rejects them.
- The **catalog** is also an entity with its own description: who runs it, which theme vocabulary it uses, on what terms the descriptions themselves can be reused. A catalog can include other catalogs. Portal federations work this way, and so does STAC, where catalogs nest inside each other like folders.

The structure of organisations and groups is worth thinking through before launch. Their short names often become part of page addresses. The Dataverse documentation warns explicitly: if you change a collection's short name, old links to it will break.

The same parts under different names

Where	Dataset	Distribution	How things are grouped	What to know
DCAT	Dataset; series is DatasetSeries	Distribution; service is DataService	catalog, theme from a vocabulary	the common language for exchanging descriptions between portals
CKAN	dataset, in the API a package	resource: an uploaded file or a link	organisation, groups, tags	the word package is left over from early versions and means the same as dataset
Dataverse	dataset	file	nested collections	a collection can contain both datasets and other collections
InvenioRDM, Zenodo	record	file	communities	each version has its own record, plus a common parent for all versions
Geoportal on ISO 19115	metadata record about a layer or service	online resource in the distribution section	keywords, topic category	one layer often has three distributions: a WMS image, WFS features, a file
STAC	collection, containing items	asset: an image file or another file	catalogs and subcatalogs	according to the STAC specification, a collection corresponds to a DCAT dataset

Indicator and microdata catalogs fit the same scheme, but with a different main object: an indicator with breakdowns instead of a file, a survey with variables instead of a table. They are covered in the chapters on indicator catalogs and on microdata.

Where the bytes live

A catalog can store the file itself, only link to the file at the source, or harvest other people's descriptions and send the person to the original portal. A link is cheaper but breaks more often. A research paper needs a copy in a repository, not the hope that the agency's website will not move.

"The file is in the catalog" does not mean "on its server". Files uploaded to data.gov.gr open at the portal's address but are served from Microsoft Azure cloud storage. For the user there is no difference. For the owner it is a budget line and a question of who is responsible for backups. Storage options and their support in platforms are covered in the chapter on storage.

What is inside the software

From the outside a catalog is a website. Inside, almost any server platform has the same parts:

- a **metadata database**: dataset pages, users, permissions; usually PostgreSQL;
- a **search index**: a separate copy of the descriptions, built for fast search and filters. If the index drifts from the database, the dataset exists but cannot be found. After migrations and upgrades the index is rebuilt;
- **file storage**: the server's disk or cloud storage;
- **table storage**: a separate database into which table rows are loaded so they can be shown in the browser and served via an API;
- **background jobs**: harvesting other catalogs, loading tables, checking links, building archives. They are queued and run separately from the website;
- the **web interface and API**: what people and programs see.

Platform	Metadata database	Search index	Files	What else
CKAN 2.12	PostgreSQL	Solr 9 or 8	server disk, cloud via an extension	Redis; the DataStore table storage is optional, a separate PostgreSQL database
Dataverse	PostgreSQL	Solr	disk or cloud storage	Payara application server
InvenioRDM	PostgreSQL 15 and later	OpenSearch 2.12 and later	network disk, S3 and others	Redis or memcached for caching, RabbitMQ or Redis for the job queue
Static site	none	none, or a small index downloaded into the browser	files next to the pages	a generator that builds the site

Each part is a separate program that has to be upgraded, backed up and monitored. The number of parts is a good first estimate of how much maintenance costs. More in the chapter on deployment and maintenance.

Two entrances

A person needs search, filters, a dataset page and a "Download" button. A program needs the same in machine form. The same machine entrance is used by other catalogs that harvest descriptions on a schedule, and by dataset search engines. A good catalog serves each part both ways.

A person sees	A program reads	How CKAN does it
search and filters	a search API	core: `/api/3/action/package_search`
a dataset page	a record in JSON or DCAT	core: `package_show`; DCAT is the ckanext-dcat extension, the dataset page address ending in `.ttl`, `.rdf` or `.jsonld`
a "Download" button	a direct link to the file	core: the resource URL
a table with a preview	an API to table rows	core, DataStore plugin: `datastore_search`
a list of all datasets	a file with all descriptions	the ckanext-dcat extension: `/catalog.ttl`
the dataset in search engine results	schema.org markup on the page, a sitemap	the markup is also the ckanext-dcat extension

Both entrances should say the same thing, but they often diverge: the page has a file that is missing from the export, or the export lacks a licence that is visible on the page. A person sees one thing, an aggregator another. Which programmatic entrances exist and how to check that they are open is covered in the chapter APIs and Bulk Download.

Taking apart a live dataset page

Take the dataset "Meteorological data of the Parnitha station" on data.gov.gr, the national open data portal of Greece on CKAN 2.11 (checked 12 October 2026). The page is simple, yet almost all the parts from this chapter are visible in it.

- **Dataset.** The publisher is the Hellenic National Meteorological Service; in the software the dataset belongs to the organisation of the same name. The theme is "Environment" from the EU theme vocabulary, there are eight tags in Greek and English. The licence is CC BY 4.0, the update frequency is "continuous". The dataset is marked as a high-value dataset under EU Regulation 2023/138, category "Meteorological".
- **Two distributions.** The first is an 18 MB CSV file with measurements every five minutes. It is loaded into table storage, so the rows can be queried through the API. The second is an "All data" ZIP archive that the portal builds itself with the ckanext-downloadall extension: inside are the same file and the dataset description `datapackage.json`.
- **Data service.** In the DCAT export, the CSV file has an access service: the table storage API with its own address. This is exactly the "distribution — data service" pairing.
- **Dates.** In the DCAT export the dataset's issue date is 13 May 2026 and its modification date is 11 October. Both are taken from the catalog record: when the page was created and when it was last changed.

If you compare the parts with each other, five discrepancies turn up.

1. **The two entrances say different things.** The page has two distributions, the DCAT export has one: the archive is missing. A person will see the ZIP, the European portal's harvester will not.
2. **The derived file lags behind.** As of 11 October 2026 the CSV contains measurements up to that day, but the "All data" archive only up to 14 August. The reason is described in the extension's documentation: a small archive is rebuilt when the dataset page changes, not when the file is updated. Anyone who downloaded "everything in one archive" got two-month-old data.
3. **The description and the contents disagree.** On the page the period starts on 7 April 2026, but the first row of the file is 1 April.
4. **There is no data dictionary.** In table storage all 57 columns are stored as text, including temperature and pressure, and none has a description. Names like `Temp_Dry_5min` make sense to a meteorologist but not to anyone else. Missing values are marked with slashes `/////`, and this is not mentioned anywhere.
5. **There is nothing for search engines.** The page has no schema.org markup, and the portal has no sitemap.

None of these discrepancies is visible on the page. All five turn up within ten minutes if you open the machine version of the same dataset page.

How to take apart any catalog

Take one dataset and look at it in three ways: the page, the API record, the metadata export. For CKAN, substitute the portal address and the dataset name from its page address:

```
# the dataset record in JSON: fields, resources, organisation
curl "https://portal.example/api/3/action/package_show?id=dataset-name"
# the same dataset in DCAT, if the ckanext-dcat extension is installed
curl "https://portal.example/dataset/dataset-name.ttl"
# the first rows and column types, if the file is loaded into DataStore
curl "https://portal.example/api/3/action/datastore_search?resource_id=RESOURCE-ID&limit=5"
```

Other platforms keep the dataset record at similar addresses: Dataverse at `/api/datasets/:persistentId/?persistentId=doi:...`, InvenioRDM and Zenodo at `/api/records/RECORD-ID`. A STAC catalog itself consists of JSON files and can be read right in the browser.

Then answer five questions:

1. How many distributions are there, and do they all open?
2. Does the download URL lead to a file or to a page?
3. Do the page and the export contain the same things?
4. Which date is on the page: the data's or the record's?
5. Is there a data dictionary, and does the period in the description match the contents?

What follows for your own catalog

- Decide what is a dataset and what is a series before creating the first pages. Converting hundreds of separate releases into a series later is expensive.
- Check whether the platform distinguishes a download URL from an access page, and data dates from record dates. If not, agree on what each field means and write it into the publishing rules.
- Think through organisations and groups in advance: their names become part of page addresses.
- Check derived files (all-in-one archives, previews, copies in table storage) the same way as the originals. They go stale unnoticed.
- Remember that each internal part of the software is a separate maintenance cost.

Sources

- W3C, DCAT 3, Recommendation of 22 August 2024: catalog, catalog record, dataset, series, distribution, data service, download URL and access URL
- CKAN: user guide: datasets, resources, organisations, the word package; installing from source: PostgreSQL, Solr, Redis, DataStore; checked 12 October 2026

- Dataverse: collection management: nested collections, changing the short name breaks links; installation prerequisites: Payara, PostgreSQL, Solr
- InvenioRDM: records model: parent record, versions, drafts, revisions; installation requirements: PostgreSQL, OpenSearch, Redis, RabbitMQ, storage
- STAC, specification overview: catalog, collection, item, asset; how a collection corresponds to a DCAT dataset
- ckanext-downloadall: when the "All data" archive is rebuilt
- data.gov.gr, Meteorological data of the Parnitha station: API record, DCAT export; checked 12 October 2026
- Built-in data storage, APIs and bulk download

The World of Catalogs in Numbers

The figures below come from an export of the dataportals-registry (release 1.27.0 of 15 October 2026). The registry describes catalogs, not the datasets inside them. The queries used to get the numbers are in the appendix.

The main finding: there are more records of the "geoportal" type than of all other types combined, 24,752 against roughly 18,800. This number includes both catalogs where a layer is found by its description and servers that serve maps without such a catalog. The CKAN open data portal is the smaller part of this world.

By type

Catalog type	Catalogs	Share
Geoportal	24,752	57%
Research data repository	8,404	19%
Open data portal	5,967	14%
Indicator catalog	3,405	8%
Microdata catalog	326	under 1%
Metadata catalog	172	under 1%
Dataset search engine	124	under 1%
API catalog	115	under 1%
Dataset list	107	under 1%
ML catalog	100	under 1%
Data marketplace	56	under 1%

A few dozen more records fall outside these types: generic research repositories and unsorted entries. They do not affect the conclusions.

Who owns them

- Local government: about 40% (17,600). Of these, 14,400 are geoportals.
- Universities and research organisations: 22%.
- Central government: 14%, regional government: 12%.
- Businesses, non-profits, international organisations and communities: about 10% together.

The long tail is a municipal geoportal on a standard platform, not the national portal that reviews write about.

What they run on

- Custom builds: 16% (7,000). Among geoportals only 4% are custom, among indicator catalogs 44%.
- ArcGIS Hub: about 3,500 geoportals; these are catalogs of layer descriptions. ArcGIS Server (about 4,700) and GeoServer (about 1,500) serve layers and are not metadata catalogs themselves. Together such servers account for about 6,100 records, almost the entire margin by which geoportals outnumber the other types. More in the chapter on geoportals.
- CKAN: about 1,900 open data portals. The Polish e-mapa: 1,900 geoportals.
- Next by number of installations come DSpace, GeoNode, GeoNetwork, OpenDataSoft, Gosweb, WordPress, Drupal and Pure.

A municipality rarely chooses a platform on its own. Sometimes the state decides for it: the Russian Gosweb is the government's platform for public authority websites. More often the market decides. e-mapa is a product of the Polish company Geo-System, and more than half of the country's gminas use it. A town in Poland ends up on e-mapa not because it compared it with GeoNetwork, but because its neighbours did so and it makes connecting to the national registers easier. In most countries, municipal geoportals are shared among a few local GIS vendors rather than global platforms.

Where there are more of them

The United States has the most records (about 8,000), followed by Poland, Germany, Italy, France, the Czech Republic, Canada, Japan, Spain, the United Kingdom and Russia. Another 1,800 catalogs are not tied to a single country.

The order of countries reflects both the density of portals and where the registry searched more thoroughly. It is not an openness ranking.

The number of catalogs in a country depends most on its income, then on its population and on how integrated it is into global science and the economy. The political regime matters less than one might expect: the Netherlands has hundreds of catalogs, Myanmar almost none, and China publishes a lot, especially research data and through provincial portals. The countries with almost no catalogs of their own are the poorest, small island states and those in crisis. Data about them still exists, but in the catalogs of international organisations rather than national ones. National portals keep appearing: in 2026, for example, Bhutan, Jordan, Oman, Monaco and Liechtenstein opened theirs.

Machine access and live sites

- A programming interface is marked for 65% of catalogs and not marked for 34%.
- 868 catalogs do not open as working sites, 98 have been decommissioned.
- 613 catalogs are flagged as national: the official catalog of its type in the country, not just any ministry website.

- The "DOI" flag is set for 2,000 catalogs: 1,400 research repositories and almost 600 geoportals. The flag is set by platform: it means the software can issue DOIs, not that the catalog issues them. In practice only a handful of geoportals issue DOIs; see the chapter on identifiers.

The share without an interface is no small matter for a federation harvester or for a program that needs a single file with all descriptions. A search engine, however, can read the page if it has schema.org markup and a sitemap.

What the numbers do not show

One record is one catalog, whatever its size. Hugging Face and the website of a small town weigh the same. The registry is uneven across countries and types: where the search went on longer, there are more records. Before citing a figure in a report, check it against a newer registry release.

The registry itself shows how much the number depends on search effort. In August 2026 it held about 14,400 catalogs, by early September 29,800, and by release 1.27.0 of 15 October 2026 more than 43,000. The world did not change in two months: catalogs began to be found by URL patterns in the Common Crawl web archive (an address with `/rest/services` almost always means an ArcGIS server, one with `/geoserver/` means GeoServer) and through internet device search engines such as Censys, and AI agents helped describe them. That is why the growth came mostly from geoportals, research repositories and regional catalogs that simply had not been seen before. Earlier totals (10,000 catalogs in 2024, 1,700 in 2023) say how many could be found, not how many existed.

The number of datasets inside catalogs is even harder: everything depends on what counts as a dataset. In the DataCite index, where DOIs for research data are registered, there were about 43 million "datasets" in the summer of 2025. Of these, 19.8 million were experiments of a single Japanese fusion research institute, each assigned its own DOI, and about 3.5 million more were individual species occurrences from GBIF, although GBIF itself holds about 115,000 datasets. By October 2026 DataCite had 76.4 million "datasets", 42.8 million of them from the same institute. In 2024 Mendeley Data showed 2.35 million datasets "from GBIF" against 91,000 real ones. ScienceBase, the research repository of the US Geological Survey, counts 18.7 million records, but most of them are slices of a few large databases by area. The same "number of datasets" field means things that differ by orders of magnitude from catalog to catalog. More on the unit of counting in the chapter on units of publication.

Sources

- dataportals-registry export, release 1.27.0 of 15 October 2026. The queries are in the appendix.
- Geo-System, summary of e-Gmina deployments in 2025 (in Polish): e-mapa is used by more than half of Poland's gminas.
- I. Begtin, distribution of geoportals by country, Telegram (in Russian), 27 September 2026: in most countries local GIS vendors lead.
- I. Begtin, Telegram (in Russian), on the growth of the registry: the start of the registry, 1,700 records, 10 April 2023; 10,000 catalogs, 26 August 2024; finding catalogs by URL patterns in Common Crawl, 22 May 2025; 29,800 catalogs, 3 September 2026

- I. Begtin, Telegram (in Russian), on counting datasets: DataCite and the fusion research institute, 21 July 2025; Mendeley Data and GBIF, 11 January 2024; ScienceBase, 1 November 2024; on the link between the number of catalogs and a country's income, post, 9 April 2024; on the new national portals of 2026, post, 16 August 2026

PART II.

What Data Gets Published

What the data in catalogs is about, where it comes from, how detailed it is, and in which units and formats it is published.

1. What Catalogs Publish
2. Units of Publication and Formats

What Catalogs Publish

The short answer: almost everything that governments, science, companies and communities know about the world and write down. A list of schools and every payment from the budget. A bus timetable and where that bus is right now. A population census and a raw satellite signal. A bacterial genome and an archival finding aid.

The catalog type from Part III tells you where to look for what: layers on a geoportal, data behind a paper in a research repository. But the content is broader than any type. A city portal holds the budget, the street trees and committee decisions. A general research repository holds tables, interview recordings, 3D models of archaeological finds and trained neural networks.

It is convenient to analyse the content with five questions. The answers determine which catalog you need.

Question	Example answers
What the data is about	budget, transport, health, climate, culture
Where it came from	agency records, statistics, sensors, research, a community
What kind of object it is	a reference list, a register, event records, an observation series, text, an image
How detailed it is	a national total, by district, an individual record, a raw signal
How often it changes	once, once a year, every day, continuously

What it is about: themes

There is no single list of themes. There are several classifications, each made for its own purpose.

- **DCAT-AP themes**: 13 headings that European portals use to tag datasets so they can be searched together: agriculture, fisheries, forestry and food; economy and finance; education, culture and sport; energy; environment; government and public sector; health; international issues; justice, legal system and public safety; regions and cities; population and society; science and technology; transport. The vocabulary is maintained by the EU Publications Office.
- **ISO 19115 topic categories**: headings for geodata, from elevation and inland waters to cadastre and communications. The 2003 version has 19 of them; ISO 19115-1 adds two more, extraterrestrial and disaster.
- **INSPIRE themes**: 34 spatial data themes that EU countries must describe and serve: addresses, cadastral parcels, hydrography, buildings, soil, species distribution, natural risk zones and others.
- **G8 Open Data Charter categories** of 2013: 14 "high-value" areas, from company registers and crime to genomes and transport timetables.
- **ODIN categories**: 22 areas of official statistics that the Open Data Watch project uses to assess national statistical offices: population, education, health, prices, trade, energy, pollution and others.

A classification lets a person narrow a search and lets an aggregator put datasets from different countries on one shelf. You can introduce your own headings, but it is better to map them to the common ones. EU national portals do exactly this: they have their own classification, but DCAT-AP themes are what goes out. More on vocabularies in the chapter on metadata standards.

If you combine these classifications and look at live portals, a map of themes emerges. It is not exhaustive, but it shows the breadth.

Government and law. The structure of bodies, contacts, officials. Legal acts and drafts. Parliamentary votes and transcripts. Court decisions. Election results by polling station. Citizens' petitions. Inspection plans and results.

Public money. The budget, planned and executed. Procurement and contracts. Grants and subsidies. Public property and debt. Officials' salaries and asset declarations. International aid.

Economy and companies. Registers of legal entities and their owners. Licences and permits. Financial statements. Bankruptcies. Prices, foreign trade, the labour market. Patents and trademarks.

Population and social affairs. Census and demography. Migration. Income, poverty, benefits. Housing. Schools and universities, exam results. Hospitals and pharmacies, the medicines register, disease incidence, clinical trial registries.

Safety. Crime, road accidents, fires, emergencies.

Territory. Addresses, administrative boundaries, cadastre, urban plans, buildings, place names, elevation, aerial imagery.

Nature and climate. Air, water and soil quality. Emissions and waste. Protected areas. Species observations. Weather, forecasts, climate series, satellite imagery.

Transport and infrastructure. Routes and timetables, real-time vehicle positions, parking, roads and roadworks, airports and railways. Power grids, telecommunications, water supply.

Energy and resources. Energy production and consumption, mineral resources, forests, fish.

Science. Results of experiments and observations, gene sequences and protein structures, sky surveys, questionnaires, field notebooks, code and models.

Culture and heritage. Museum collections, library catalogs, archival finding aids, digitised manuscripts and photographs, heritage sites, text corpora.

Reference data. Codes for countries, regions, economic activities, currencies. On their own they interest few people, but without them datasets from different agencies cannot be combined.

Where data comes from

Origin matters as much as the theme. It determines completeness, methodology, update frequency and what needs to be explained in the dataset description.

Administrative data. A by-product of an agency's work: a register of licences exists because licences are issued, a register of contracts because contracts are signed. This is the backbone of open data portals. Coverage is complete: everyone who received a licence is in the register. But the fields were designed for record-keeping, not for analysis, and they change along with the regulations.

Official statistics. Collected on purpose, according to a methodology: censuses, sample surveys, business reporting. Aggregate indicators are published, anonymised microdata less often. Without the methodology the figures are easy to misread.

Observations and measurements. Weather stations, air quality sensors, river gauges, meters, satellites. Large volumes, long series, often a real-time stream.

Research. An experiment, modelling, fieldwork, a survey. Reproducibility is what matters here: the data should come with its version, methodology and processing code.

Texts and documents. Laws, court decisions, transcripts, minutes. A document on its own is not yet data. Structure makes it data: markup of articles, links to other acts, fields such as "date, court, parties".

Communities. OpenStreetMap, Wikidata, observations by amateur naturalists that reach GBIF through iNaturalist. Coverage can be wider than the government's, quality is uneven, and the licence is their own: OpenStreetMap, for example, uses ODbL.

Companies. They disclose some information by law: issuer reporting, emissions. Some they publish themselves when it pays: a transport operator publishes its timetable so that navigation apps show it. The rest is available under contract, for a fee or not at all.

Derived data. Forecasts, models and reanalyses, for example the ERA5 climate reanalysis, which combines observations and a model into continuous series for the whole planet. This also includes indices, rankings and trained machine learning models. They are published together with a note on which source data they were derived from.

What kind of object it is

Objects of different nature sit within one theme. Their nature determines how to store, update and display them.

- **Reference lists and classifications**: codes and names. They change rarely and other datasets refer to them, so the version and date matter.
- **Registers of objects**: companies, schools, licences, buildings. They answer the question "what exists now". They change every day and need stable record identifiers and a change history.
- **Records of events and transactions**: contracts, payments, road accidents, ambulance calls. They accumulate endlessly and are published in batches by period.
- **Observations and time series**: temperature, water level, pollutant concentration by the hour.
- **Indicators**: aggregate figures by territory and year with breakdowns.
- **Microdata**: records about individual people, households or firms after anonymisation.
- **Spatial data**: vector features (boundaries, roads, buildings) and rasters (imagery, elevation).

- **Multidimensional arrays**: climate models and satellite products, with a grid, time and dozens of variables in one file.
- **Texts, images, audio, video, 3D**: corpora, digitised manuscripts, interview recordings, scans of museum objects.
- **Sequences and spectra**: genomes, proteins, mass spectrometry results. There are dedicated global archives for them.
- **Code, models and protocols**: without them a research result cannot be reproduced, so they are published next to the data.
- **Descriptions of other data**: vocabularies, schemas, ontologies, catalogs of catalogs. They are covered in the chapter on metadata catalogs.

The files and services in which all this is served are covered in the next chapter.

Depth: from total to record

The same subject can be published at different depths. Here are the steps for three themes.

Theme	Total	More detail	Individual record	Deepest
Budget	the budget law	execution by line item and month	every payment	a contract with stages and documents
Transport	a list of stops	routes and timetables	where each bus is now	anonymised trips by travel card
Health	disease incidence by region per year	by hospital and month	anonymised case records	medical records, only in a secure environment

In the Earth sciences the steps are formalised. NASA divides satellite products into levels 0 to 4. Level 0 is the raw instrument signal. Level 1 is the same signal referenced to time and place and calibrated. Level 2 is physical quantities, for example sea surface temperature, at the observation points. Level 3 is the same quantities on a regular grid, per day or per month. Level 4 is the output of a model that combines several measurements. An analyst usually needs level 3, someone checking an algorithm needs level 1.

The deeper the step, the more questions you can ask of the data, but the higher the risks and requirements. At depth, personal data and trade secrets appear, volumes grow by orders of magnitude, and you cannot do without an API and a data dictionary. A portal that publishes only the top step looks complete but does not answer the questions people look for data to answer: who was paid, where the bus is, which hospital.

How often it changes

- **Once.** A study's results, an archival collection. It needs an immutable version and a persistent identifier.
- **Periodically.** An annual report, monthly statistics. It needs one series with a consistent structure, not a new dataset every year.
- **Constantly.** A register that is edited every day. It needs snapshots by date and an export of changes.

- **Continuously.** Transport, air sensors, power grid load. This cannot be published as a file: it needs a stream or an API, and the catalog needs a description of that stream.

Domain standards

When many publishers publish on the same theme, a common format appears. It describes not a file in general but the subject: what a contract, a stop or a species observation is. Programs and aggregators understand data in such a format immediately: map services pick up timetables in GTFS, GBIF harvests observations in Darwin Core.

Theme	Standard	What it describes
Procurement and contracts	OCDS	the whole contracting process: planning, tender, award, implementation
Company ownership	BODS	who ultimately owns a company and through which chains
International aid	IATI	who gives aid to whom and for what
Foundation grants	360Giving	who gave how much to whom and for what
Budget	Fiscal Data Package	budget lines with classifications
Public transport	GTFS and GTFS Realtime, NeTEx in Europe	stops, routes, timetables, vehicle positions
City service requests	Open311 GeoReport v2	reports of problems in the city and their status
Social services	HSDS (Open Referral)	organisations, services, addresses, opening hours
Legislation	Akoma Ntoso	the structure of an act, amendments, links to other acts
Company reporting	XBRL	reporting lines with machine-readable tags
Statistics	SDMX	indicators with breakdowns
Surveys	DDI	questionnaire, variables, sample
Biodiversity	Darwin Core	observations and specimens: species, place, date
Climate, ocean, atmosphere	NetCDF with the CF conventions	gridded variables with units and coordinates
Cultural heritage	the Europeana Data Model (EDM), IIIF	description of an object, display of high-resolution images
Astronomy	FITS and IVOA standards	images, spectra, catalogs of sky objects
DNA sequences	INSDC rules	genomes and reads in shared global archives
Machine learning	Croissant	the files, fields and splits of a dataset

There are two conclusions for a catalog. First, if a standard exists for the theme, publish in it and offer CSV in addition. Second, a catalog does not have to understand every standard from the inside, but it must be able to say which standard a file conforms to. DCAT has the `dct:conformsTo` field for this. Specialised platforms go further and check the file with a validator. More in the chapter on metadata completeness.

What must be published

Mandatory lists set the minimum. They show what the state considers key.

- **EU.** Regulation 2023/138 on high-value datasets has applied since 9 June 2024. It has six categories: geospatial, earth observation and environment, meteorological, statistics, companies and company ownership, mobility. They are provided free of charge, in machine-readable form, via APIs and, where relevant, as bulk download. Earlier, the INSPIRE Directive required 34 spatial data themes to be described and served through services.
- **United States.** The OPEN Government Data Act is Title II of the Foundations for Evidence-Based Policymaking Act of 2018, signed in January 2019. It makes federal agency data open by default and requires agencies to maintain a comprehensive inventory of their data. The federal catalog data.gov harvests the descriptions.
- **Russia.** Government Order No. 1187-r of 10 July 2013 (as amended in 2022) contains three lists. Federal bodies publish their subordinate organisations, inspection plans and results, statistics, vacancies and registers of licences. Regions publish registers of licences for educational, medical and pharmaceutical activities, and the forest and hunting registers. The third list names information from the registers of Rosprirodnadzor and Roshydromet: waste, facilities that pollute the environment, emission permits, weather forecasts, pollution data.

International benchmarks are broader than obligations. The 2013 G8 Charter named four groups first (national statistics, maps, elections and budgets), followed by 14 high-value areas. The ODIN index checks 22 areas of official statistics not only for availability but also for detail, by year and territory.

The lists have a weak spot. It is easier to put into them things that are simple to define: names of bodies, lists of licences. A list can be fulfilled formally without publishing either budget payments or timetables. So a list is a lower bound, not a map of what people need.

What is usually missing

Information about individuals. Medical records, citizens' tax data, records of benefit recipients. Instead, aggregates are published, anonymised microdata on request, or synthetic data, which are artificial records with the same statistical properties. Another way is to allow work with the records in a secure environment without handing them out. This is covered in the chapters on microdata catalogs and on access and roles.

Secrets and anything related to security. Trade secrets, detailed network diagrams, critical infrastructure sites, some precise geodata.

What is charged for. In many countries the cadastre, meteorology and company registers used to be sold. The EU regulation made six high-value categories free, but outside the EU paid access to such data is common.

Raw and fast data. Primary records and daily series stay in the record-keeping system and are released as aggregates.

Company data on customers and services. It is published rarely and usually as aggregates. Exchange happens under contracts, increasingly in data spaces; see the chapter on AI and data spaces.

Supporting material. A data dictionary, the methodology, the change history, a list of known errors. Without a data dictionary a file is formally open and practically mute.

Failed and intermediate research results. What made it into the paper goes into the repository. Negative results and draft exports usually stay with the author.

What was withdrawn from publication. A dataset can disappear by the owner's decision, after a change of system, management or policy. Then only copies in web archives and at aggregators remain. This is an argument for storing the data itself, not just a link to someone else's site, and for never changing addresses without a redirect.

What the registry shows

The dataportals-registry describes catalogs, not datasets, and marks their content roughly: what kind of material is in the catalog. According to the dataportals-registry export (release 1.27.0 of 15 October 2026), the picture is as follows. One catalog can appear in several rows.

Content	Catalogs
Datasets	42,297
Map layers	24,655
Indicators	2,876
Publications	457
Microdata	151
Documents	120

Layers almost match the number of geoportals. Indicators are fewer than indicator catalogs: not every statistics showcase has its content tagged. Dataset themes (budget, transport, climate) cannot be counted from a catalog registry. For that you need dataset search engines such as Dateno, or the data.europa.eu catalog, where datasets carry DCAT-AP themes.

What this means for a catalog

Before choosing a platform, it is useful to draw up an inventory: for each group of data, write down the theme, origin, kind of object, depth, update frequency, domain standard and access restrictions. The inventory shows which capabilities are mandatory.

If you publish	What you will need	Where to read more
a real-time stream	an API or a stream and its description in the catalog	access interfaces
large arrays, imagery, models	storage outside the catalog, download in parts	storage
a study's results	an immutable version, a DOI, citation export	identifiers, citation
a register edited every day	snapshots by date, versions, stable record identifiers	identifiers and versions
microdata and records about people	access on request, roles, an access log	access and roles
data in a domain standard	a conformance statement, file validation	metadata completeness
map layers	map services, a layer catalog	geoportals

If you publish	What you will need	Where to read more
tables, texts, images	a preview on the dataset page	other capabilities
many themes and agencies	a shared theme classification and a mapping to your own	metadata standards

How to get from these requirements to features and a platform is covered in the chapter on selecting features.

Sources

- The EU Publications Office Data theme vocabulary, on which the DCAT-AP themes are based.
- ISO 19115 code lists, section MD_TopicCategoryCode.
- INSPIRE themes.
- G8 Open Data Charter and Technical Annex, 2013.
- Open Data Inventory: categories and methodology, 2024/25 edition.
- NASA EOSDIS data processing levels.
- Commission Implementing Regulation (EU) 2023/138 on high-value datasets.
- US Public Law 115-435, Title II, the OPEN Government Data Act.
- Russian Government Order No. 1187-r of 10 July 2013 (in Russian), as amended on 10 November 2022.
- Domain standards: OCDS, BODS, IATI, 360Giving, Fiscal Data Package, GTFS, NeTEx, Open311 GeoReport v2, HSDS, Akoma Ntoso, XBRL, Darwin Core, CF conventions, IIIF, IVOA, INSDC.
- ERA5 in the Copernicus Climate Data Store.
- Catalog content composition: dataportals-registry export, release 1.27.0 of 15 October 2026.

Units of Publication and Formats

"Publishing data" means different things. In one catalog it is a file. In another it is a map service. In a third it is an indicator for which the user picks a year and a region. Confusing these units is harmful: they determine both the number of "datasets" in a report and the requirements for the platform.

What counts as one publication

- **A file or a batch of files:** CSV, JSON, XML, Parquet, sometimes XLSX.
- **An API:** a program retrieves the same data with a request, without downloading the whole archive.
- **A map service:** an image of a layer (WMS, WMTS) or the features themselves (WFS, OGC API Features). An image without features is viewing, not downloading.
- **An indicator or a cube:** a table broken down by time and territory. Formats are SDMX, JSON-stat, PC-Axis.
- **A survey:** a description of the variables plus the record files. The description is usually in DDI.
- **A dataset for training a model:** files plus a dataset card, increasingly in the Croissant format.

One subject can be published as a dozen files "by year and district" or as one dataset with an API. For a person, a dozen files is sometimes clearer. For a report saying "we have 10,000 datasets", it is a way to inflate the number. More on this in the chapter on anti-patterns.

What counts as a dataset

The word "dataset" means different things in different catalogs, and when comparing catalogs this is the first thing to find out.

- **A time series.** Central bank portals, such as those of the ECB and the Bank for International Settlements, have few datasets and they are huge, but each series, say inflation in one country, has its own persistent address. A user usually looks for the series for their country, not the whole database. That is why dataset search engines show series as separate entries: in Dateno in 2024 about 30% of records were organised this way.
- **A map layer.** Most often this is not a file but the address of an ArcGIS or OGC service. It is customary to count it as a dataset, but it is more honest to call it a layer.
- **An imagery collection.** In STAC catalogs one collection is a series of images of a territory on different dates, with thousands or millions of files inside. In Open Data Cube the word dataset refers precisely to one image on one date, and what other catalogs call a dataset is a product. The licence is set on the product.
- **An observation record.** The global biodiversity database GBIF harvests about a hundred thousand datasets, and inside them there are billions of species observations. If each observation is given an identifier and called a "dataset", the counter grows by orders of magnitude. This is exactly what some aggregators do; see the chapter on numbers.

- **A document and a feed.** Are RSS and ATOM data or not? In 2024 the European portal data.europa.eu had more than 141,000 datasets that were actually ATOM feeds from geoportals.

A working criterion is reuse. A program's configuration file is machine-readable, but it is not a dataset: nobody takes it for their own task. A collection of texts or photographs is not machine-readable in the usual sense, but as a training set it is a dataset. A dataset is whatever a user searches for, downloads and uses as one whole.

Dataset or data product

A dataset is what has been published. A data product is what has been published for specific users and is maintained. The distinction comes from enterprise data work, but it is increasingly applied to open data: in 2025–2026 it was discussed, for example, by the creators of the open data cloud storage Source Cooperative. The signs of a data product:

1. It is known who uses it and for what: there are main use cases, even if the product is free.
2. There is someone responsible who understands both the data and the users, answers questions and makes changes based on feedback.
3. There is documentation: the data schema, code examples, limitations.
4. Access is provided in several ways at once: an API, bulk download, individual slices. Access can also be controlled, by registration or in a cloud database.
5. Data quality is described in the documentation and checked continuously, not once before publication.
6. There are versions and an update schedule.
7. A dashboard or interactive showcase complements the data rather than replacing it.

The World Bank API fits this description, as do the Amazon Registry of Open Data (each entry there is a documented collection with usage examples) and the Russian address register FIAS with full exports and updates. Most datasets on typical open data portals do not, nor does a dashboard with a "download to Excel" button. By I. Begtin's estimate (August 2026), on well-known open data portals at best 5% of what is published is regularly updated; the rest was uploaded once.

Catalogs are built for one or the other. CKAN, DKAN, GeoNode and GeoNetwork count datasets and the files attached to them. Huwise (formerly OpenDataSoft) stores data as tables and provides an API and exports for every dataset, so it is closer to a product catalog. Source Cooperative is even simpler: it is file storage where the product page is a README file with a schema description, code examples and coverage. Each product has its own access method, but large collections, up to hundreds of terabytes, are well described because the authors curate them themselves.

There is an open specification for describing a data product, ODPS (Open Data Product Specification, version 4.1): it describes a product with several access interfaces, terms and a price. It is still rarely used. A practical list to check your own dataset against is in the section Dataset as a product of the chapter on good practices.

When a file is usable by a program

Tim Berners-Lee's "five stars" scale (2010) goes from a file under any licence to a machine-readable open format and further to links to shared identifiers. In practice, three questions are enough.

1. The file opens without paid software. CSV, JSON, XML, Parquet, GeoPackage: yes. PDF and scans: no.
2. The table has a header row and no cells merged "for looks". An XLSX file with a three-row header breaks a program, even though the format is formally tabular.
3. The columns are named and explained. A schema such as Frictionless Table Schema or CSV on the Web says that a column holds a number, not text.

The fifth star, linked data with links to shared reference lists, is needed when datasets from different organisations should join on their own. For a city budget it is not the first priority. The first three questions are.

Which file to publish

The format is chosen by three questions: what users will work with the data in, what you can convert what you have into, and how much disk space and bandwidth you have. For most cases the answer fits a short rule.

- **A flat table**: CSV in UTF-8, comma as the delimiter, the first row as headers. Any program can read such a file. GitHub renders it as a table right in the browser if the file is no larger than 512 KB, while a file with semicolons is shown as plain text.
- **Records with nested lists** (an organisation has several addresses, a contract has several stages): JSON Lines, one JSON record per line. A program reads such a file line by line. Ordinary libraries load one big JSON or XML file into memory whole, and a file of several gigabytes becomes a problem.
- **A large table for analysts**: Parquet in addition. It stores column types, so a program does not have to guess where a number is and where text is, and it compresses data by column.
- **Domain data**: in the domain format, GeoJSON or GeoPackage for geodata, NetCDF for measurements, SDMX for indicators. A CSV can sit next to it, but not instead of it.
- **XLSX**: as an extra copy for people who will work in a spreadsheet. Any spreadsheet can import a comma-delimited CSV.

In practice portals publish differently. In July 2025 the Dateno search engine counted 6.7 million files and links in datasets from open data portals. CSV made up 15%, XLSX, XML, JSON and ZIP 7–8% each, but another 7% were PDF and the same share were links to HTML pages, that is, reports and other people's websites instead of data. Parquet occurred 1,652 times, less than 0.025%. In the same search engine's experience, for every CSV that can be read without manual fixing there are two with export errors: a different delimiter, encoding, extra rows above the header.

The difference in size is noticeable. The same 3.5 GB CSV took 8.7 GB after conversion to JSON Lines and 0.93 GB in Parquet (an example from the Dateno search engine's practice, 2024). JSON Lines costs more space, but preserves what does not fit into a table.

Compression

A text file larger than 20–30 MB is worth compressing. The archiver is chosen by who will read the file and how.

- **gzip or zstd**: for one large file that programs will read. Many tools open such a file without unpacking: DuckDB, for example, recognises `.csv.gz` and `.csv.zst` by the extension.
- **ZIP**: for a batch of files that are downloaded together. Zenodo lists the files inside a ZIP right on the record page. Analytical software reads ZIP less well: DuckDB will not open it without the zipfs extension.
- **RAR and 7z**: better not. They are popular archivers among users, but few libraries can stream a file from inside such archives without unpacking it to disk.

What to choose

Publish the unit that people use. An official statistics indicator is not replaced with an annual CSV "to tick the box" if international exchange happens in SDMX: the CSV can be an additional file. A cadastral layer is not replaced with a picture of a map if a person needs to download the boundaries. Research data behind a paper is fixed as a file version, not as "always the latest export".

A format without a field description and without a licence remains a file, not a publication.

Sources

- 5-star open data
- Frictionless Table Schema
- JSON Lines, Apache Parquet
- Rendering CSV and TSV data on GitHub: delimiters, the 512 KB limit, checked 12 October 2026
- Reading CSV in DuckDB: the `compression` parameter, gzip and zstd; the zipfs extension for ZIP
- I. Begtin, Good and bad data publishing practices. Metadata and file formats (in Russian), November 2024: choosing formats and compression, the file size example
- I. Begtin, Telegram (in Russian), on formats in the Dateno index: July 2025, 9 July 2025; Parquet and CSV with errors, 20 March 2024
- I. Begtin, Telegram (in Russian), on what counts as a dataset: reuse as the criterion, 22 September 2024; time series, layers, ATOM feeds, 25 November 2024; Open Data Cube terminology, 18 May 2023
- I. Begtin, Telegram (in Russian), on data products: dataset and data product, 5 August 2026; signs of a data product and dataset catalogs and product catalogs, 12 August 2026; Source Cooperative, 25 August 2026; ODPS 4.1, 8 April 2026
- Radiant Earth, Great Data Products, November 2025; Open Data Product Specification; Source Cooperative; Open Data Cube
- I. Begtin, Telegram (in Russian), dataset catalogs and data product catalogs, 12 August 2026: the estimate that no more than 5% of what is published is regularly updated
- What catalogs publish

PART III.

Types of Catalogs

How to classify data catalogs and how the main types differ: open data portals, geoportals, research repositories, indicator and microdata catalogs, ML catalogs, metadata registries and aggregators.

1. How to Classify Catalogs
2. Open Data Portals
3. Geoportals
4. Research Data Repositories
5. Indicator Catalogs
6. Microdata Catalogs
7. ML Catalogs
8. Metadata Catalogs and Registries
9. Search Engines and Aggregators
10. API Catalogs, Marketplaces and Dataset Lists
11. Comparing Types and Hybrids

How to Classify Catalogs

A data catalog is described by its answers to five questions: what it publishes, who runs it, where the data itself is, who can get it, and what it is built on. The answer to the first question is the catalog type. The type determines the description standard and the range of suitable platforms. The other four answers explain why two catalogs of the same type are still chosen and maintained differently.

Classification is not about tidying the shelf. It helps you find the catalogs to compare yourself with, the standard that aggregators expect, the platform class and the typical failures. A city's open data portal on CKAN and the same city's geoportal on ArcGIS Hub are both "about municipal data". But their standards differ, their platforms differ, and they send their descriptions to different federations. If you call both sites by one word, you will end up comparing the incomparable.

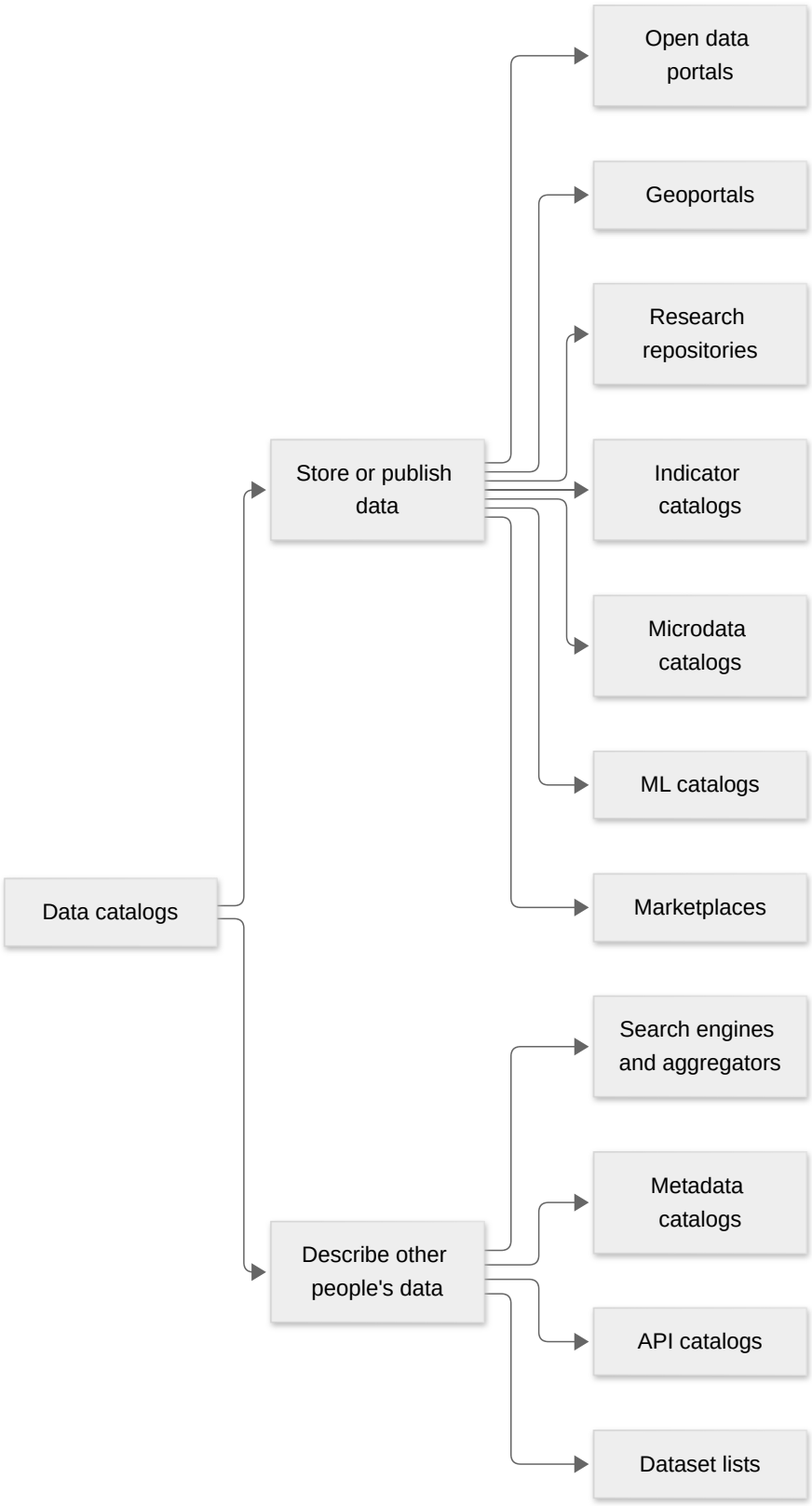
Five questions

Question	Possible answers	What follows from the answer
What is published	files and registers, map layers, research data, indicators, microdata, machine learning datasets, data for sale, descriptions of other people's data	catalog type, description standard, platform class
Who runs it and at what level	central government, region, city, university or research organisation, business, non-profit or international organisation, community	budget and team, legal obligations, whom the catalog passes its descriptions on to
Where the data is	in the catalog itself, linked at the source, the catalog only harvests other people's descriptions	whether a DOI can be issued, who is responsible for dead links, whether you need your own storage
Who can get the data	everyone without conditions, after registration, on request, for a fee; the description is open but the data is closed	roles and the release procedure, whether request tracking is needed, what to put in the licence
What it is built on	hosted service, ready-made platform on your own server, map server or protocol server, CMS website, custom build	who upgrades the software, whether a feature can be added, how much it costs to leave

The answers are independent of each other. A research repository can belong to a university or a government, be open or request-based, run on its own server or in the cloud. So the classification is easier to record as a line rather than a single word: "district geoportal, layers on its own map server, open access, hosted ArcGIS Hub". Each part of the line points to its own chapter of the book. How to write such a line for your own catalog is shown at the end of the chapter.

The figures in this chapter come from the dataportals-registry export (release 1.27.0 of 15 October 2026), which holds 43,614 catalogs. The registry is one source among others and sometimes gets the type wrong itself. Its limitations are discussed in the chapter The World of Catalogs in Numbers.

What is published: the catalog type



The two branches differ in the main thing. Catalogs in the left branch publish data themselves. Catalogs in the right branch lead to other people's data and are responsible only for the description. Within a branch, the type is determined by the main unit: what gets its own page and

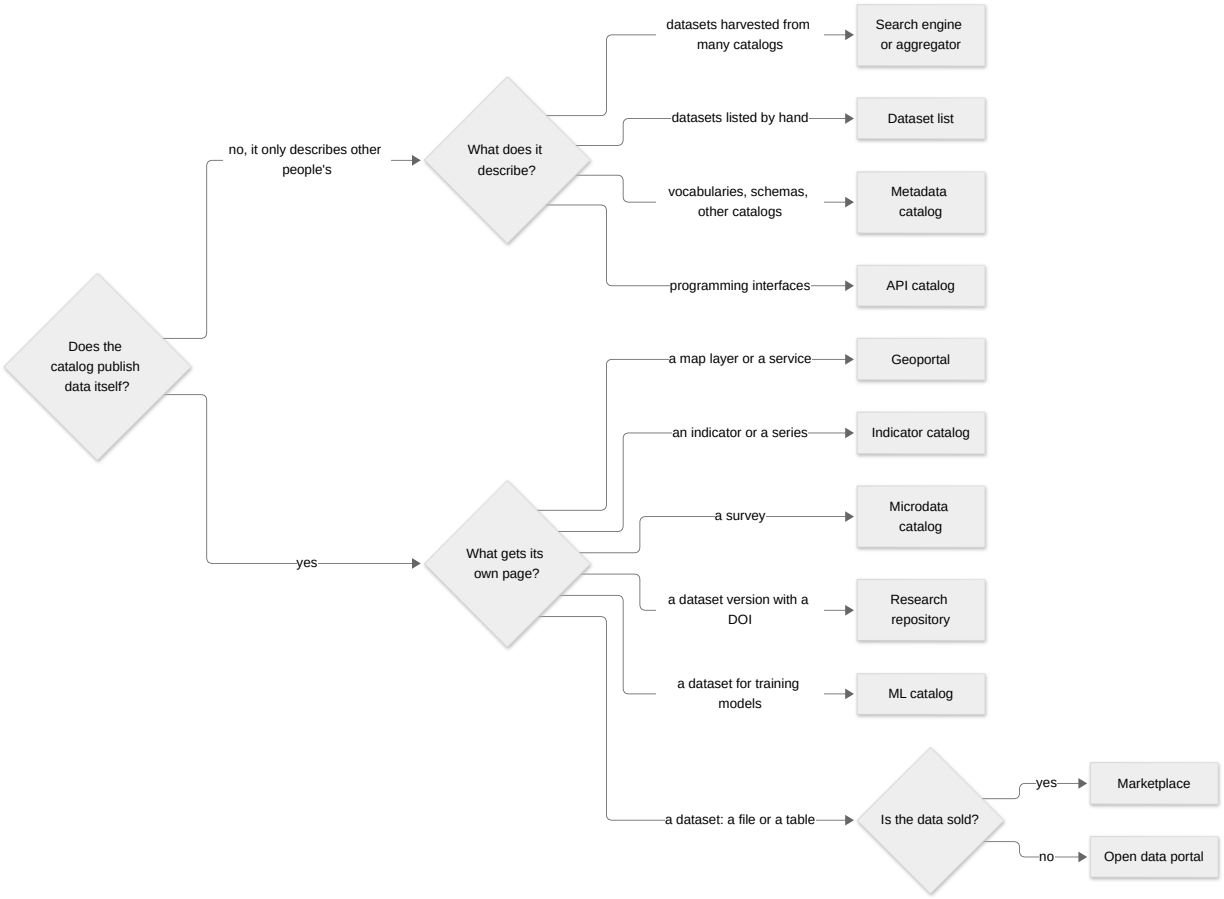
its own persistent address in the catalog.

Type	Main unit	Description standard	Example
Open data portal	a dataset: a file or a table	DCAT	data.gouv.fr
Geoportal	a layer and a map service	ISO 19115, CSW	Geoportal.de
Research repository	a dataset version with a DOI	DataCite	Zenodo, PANGAEA
Indicator catalog	an indicator and a time series	SDMX	ECB Data Portal
Microdata catalog	a survey and its files	DDI	World Bank Microdata Library
ML catalog	a dataset for training and evaluating models	Croissant	Hugging Face Datasets
Marketplace	a product: a dataset or a subscription under contract	no common one	AWS Data Exchange
Search engine or aggregator	a record about someone else's dataset	whatever it harvests: DCAT, schema.org, DataCite	data.europa.eu
Metadata catalog	a vocabulary, a schema, a description of another catalog	SKOS, SDMX, ISO/IEC 11179	re3data
API catalog	a programming interface	OpenAPI	api.gov.uk
Dataset list	a link to a file	usually none	awesome-public-datasets

Four large types (geoportals, research repositories, open data portals and indicator catalogs) account for 97.5% of the registry's records. The other seven types together amount to about a thousand catalogs. A small type is not unimportant because of that: there are only 326 microdata catalogs, but without them there is no access to the surveys on which social statistics rest.

How to determine the type

Look not at the site's name or the buttons on the home page, but at what gets its own address in the catalog. If the series "euro to dollar exchange rate" has its own page with a chart and an API, and there are no files by year, it is an indicator catalog, even if the site calls itself an open data portal.



If a site does two jobs at once, the type is chosen by what a person sees and does first. A map and a layer catalog make a geoportal, even if a few tables sit next to them. A dataset list with an embedded map is an open data portal. The platform should be chosen the same way: not "as universal as possible", but for the main product. Sites that genuinely do two jobs, and hubs made of several catalogs, are covered in the chapter on comparison and hybrids.

Who runs it and at what level

The owner determines the money, the team and the obligations. The owner also often decides which platform the catalog ends up on and where its descriptions go: a city passes them to the region, the region to the country, the country to the European or another shared portal.

Owners in the registry are distributed as follows: local government 40%, universities and research organisations 22%, central government 14%, regional government 12%. Businesses own 4% of catalogs, non-profits 3%, international organisations 3%. Each type has its own main owner.

Type	Most common owner
Geoportal	local government 58%, regional 14%
Open data portal	local government 48%, central 19%, regional 14%
Research repository	universities and research organisations 85%
Indicator catalog	central government 46%, regional 19%, international organisations 11%
Microdata catalog	central government 55%, universities and research organisations 32%

What changes from level to level:

- **City.** Few datasets, one or two people. The platform is often not chosen by the city: in Russia it is often set by the state website platform Gosweb, in Poland by the market, where more than half of the gminas run on e-mapa. Passing descriptions upwards matters more than having a showcase of your own. See the municipality example.
- **Region.** Harvests city catalogs and passes descriptions to the country. For a region, harvesting and export are two mandatory features.
- **Country.** Runs the national catalog of its type and is accountable under the law: the INSPIRE Directive for geodata, the EU Open Data Directive and the high-value datasets regulation, national lists. About 600 catalogs in the registry are flagged as national, although the flag is not applied consistently everywhere.
- **University or research organisation.** Requirements come from funders and journals: DOIs, versions, citation, harvesting of descriptions via OAI-PMH.
- **International organisation.** Publishes data about countries that have almost no catalogs of their own, and often keeps several catalogs of different types under one banner.
- **Business.** Opens data for the sake of a product, or sells it and closes access when it stops paying off. This is how data.world closed its open community in July 2026: more in the chapter on cloud storage and hubs.
- **Community and non-profits.** Sometimes build a catalog before the government does, like the Open Data Hub hubofdata.ru in 2010–2011, but rely on enthusiasts.

The owner does not set the type. 15% of the catalogs of universities and research organisations are geoportals, not research repositories. Nor does the owner set the coverage: an international organisation publishes data about countries, and a university about the territory where the research took place.

Where the data is

This question divides catalogs into three groups.

- **Stores it itself.** The file is uploaded to the catalog and lives as long as the catalog does. This is the normal mode of a research repository. This is where issuing a DOI is most honest: the identifier promises that the same file will be found by it in ten years, and only whoever stores the file can keep that promise.
- **Links to the source.** The catalog stores the description and the address, while the file sits on the agency's website or a map server. Many open data portals and most geoportals work this way. The risk is dead links: in a full export of data.gov.uk in 2023 about 10% of file links did not open, and by 2026 more than a quarter.
- **Harvests other people's descriptions.** An aggregator or search engine is not responsible for the data, but it is responsible for not showing dead entries and not multiplying one dataset that the city, the region and the country all have.

A protocol server is a special case: THREDDS, ERDDAP, a STAC server. Data is served on request, and the catalog and storage coincide here.

The answer often depends not on the type but on the platform and the owner's decision. CKAN can both store a file and link to someone else's. Zenodo expects the file to sit with it. ArcGIS Hub shows layers that live on ArcGIS servers. The options and when to choose which are covered in the chapter on data storage.

Who can get the data

re3data, the registry of research repositories, offers a convenient scheme. It asks not one question about access but three:

1. Is the catalog itself open: can you search and read the descriptions?
2. Is the data open? Options: open, closed until a certain date (embargo), released with restrictions, closed entirely.
3. Who can upload data: everyone, only members, or no outsiders.

Restrictions are listed separately: a fee, registration, institutional membership. The scheme is not only for science. A portal can be open for reading and closed for uploading, and these are two different decisions with different consequences for roles and for reviewing publications.

In the dataportals-registry, 97% of catalogs are marked only as open. In 1.4% part of the data is open and part is released with restrictions, in 1.2% access is restricted only. But the difference by type is huge:

Type	Share of catalogs with restricted access to at least part of the data
Marketplace	91%
Microdata catalog	49%
API catalog	14%
Metadata catalog	10%
Research repository	6%
Indicator catalog	4%
Open data portal, geoportal	under 1%

Restricted access can be a feature of a type rather than a flaw. Half of the microdata catalogs release some files on request because surveys contain information about people. The description remains open, though: a person must know the data exists in order to request it. The UK Data Service and the catalog of restricted public sector data in Ireland work this way; they are covered in the chapter on metadata catalogs. For an open data portal, on the contrary, closed access is a sign of trouble.

Access and licence are different questions. Access answers whether you can get the data. The licence answers what you can do with it. Data can be publicly available but not open: you can look at it, but you cannot download and reuse it. Examples are in the section Publicly available but not open of the chapter on geoportals, and the rules are in the chapter on licences and access.

What it is built on

The software class answers the questions of who upgrades it, whether a feature can be added and how much it costs to leave. The classes are covered in detail in the chapter on classes of software. Here is how they are distributed.

Class	Share of catalogs	Most common
Hosted service	30%	ArcGIS Hub (about 3,500), e-mapa (about 1,900), OpenDataSoft, Figshare
Ready-made platform on your own server or a contractor's	30%	CKAN, GeoNode, GeoNetwork, DSpace, Dataverse, PxWeb, NADA
Map server, protocol server or domain system	20%	ArcGIS Server (about 4,700), GeoServer (about 1,500), GBIF IPT, THREDDS, ERDDAP
Custom build	16%	—
CMS website	4%	Gosweb, WordPress, Drupal

With a specialised platform the type can be guessed almost for certain. CKAN accounts for 1,906 open data portals and only 39 research repositories. DSpace and Dataverse occur almost only among research repositories, GeoNode only among geoportals. A general-purpose program says nothing about the type: WordPress occurs in nine types out of eleven. The reverse does not work: the type narrows the choice of platform but does not choose it. How to get from the type and the required features to a specific platform is covered in the chapter From Features to a Platform.

Other attributes

Some attributes do not change the type but matter when you are looking for peers to compare yourself with.

Breadth of subject. re3data divides repositories into disciplinary, institutional, multidisciplinary, project-specific, governmental and other. One repository can belong to several groups at once. PANGAEA is disciplinary, Earth sciences. Harvard Dataverse is institutional and multidisciplinary at the same time. Zenodo is classed as other. The division works outside science too. A domain catalog lives by the standards of its field: biodiversity data is described in Darwin Core and published through GBIF IPT, oceanographic data is served by ERDDAP servers. A country's general portal does not compete with them but harvests their descriptions.

Territory. The world, a group of countries, a country, a region, a city. Coverage does not coincide with the owner: the World Bank publishes surveys from dozens of countries, GBIF attaches an observation to the country where it was made, not to the country of whoever sent it.

Official status. The national catalog of its type is the one where a country's data is collected by law or by agreement. It is compared with the national catalogs of other countries, not with the portal of the neighbouring ministry.

State. Working, abandoned, decommissioned, kept as an archive. In the registry 868 catalogs do not open as working sites and 98 are officially decommissioned. Comparing yourself with a dead catalog is pointless, but it is worth learning from one: why catalogs close and how they are rescued

is covered in the chapter on archiving.

What the descriptions are kept for. This attribute explains why catalogs of the same type have such different depth of description. In 2021 I. Begtin identified five models of working with metadata:

- research: do it right and without haste, by the standards of the discipline, so that it is convenient for researchers in that field;
- archival: preserve it so that the data can be understood and reused in 10–20 years;
- civic and governmental: open as much as possible quickly; availability matters more than quality, descriptions are minimal;
- technological: connect information systems and impose exchange rules, as reference data portals do;
- commercial: help analyst teams exchange data and not lose their work; this is the world of enterprise catalogs.

Research repositories and microdata archives follow the first two models, open data portals the third. Hence the difference in how detailed the descriptions are. What of this is mandatory for your catalog is covered in the chapter on metadata completeness.

Borderline cases: is it a catalog

A site can be considered a data catalog if three conditions are met:

1. A dataset has a description: at least a title, a source and a way to get the data.
2. Datasets can be found, by search, a list or a programming interface.
3. The data can be obtained, or at least you can find out how to request it.

Without the first condition it is a file archive or a website section with attachments. Without the third, it is a showcase where the data can be viewed but not taken; this case is covered in the anti-pattern Dashboard instead of data. Below are cases where the answer is not obvious.

A publications repository with almost no data. Universities run archives of papers, theses and books on DSpace, ePrints or Pure, and sometimes datasets end up there. In December 2024 the open access archive of St Petersburg State University held 47,619 records, of which 17 were datasets, that is, 0.03%. DSpace does not filter search by record type out of the box, so to find those 17 datasets the descriptions of all records had to be exported. Such an archive is a research repository but not a data repository. It is worth including in lists, but not expecting from it what you expect from a data catalog. Why such archives are hard to harvest is covered in the chapter on research repositories.

A map server without a catalog. ArcGIS Server and GeoServer serve layers but do not keep descriptions for search on their own. The registry counts them among geoportals: about 6,100 records. The book considers them services that a geocatalog relies on, not catalogs. More in the chapter on geoportals.

A reference data portal. Portals of regulatory reference data resemble open data portals: many datasets, export, an API. But they have a single owner, uniform terms of use instead of a licence per dataset, and they are built for exchange between government systems. The deciding question is what a person does on the site: downloads a reference list as data or checks codes against it. See the section Reference data portals.

An open budget website. If the budget tables can be downloaded with a description and a date, it is a single-topic open data portal. If the site has only PDF reports and charts, it is a site with publications, not a catalog.

A feed or an interface inside a catalog. ATOM feeds, WMS endpoints, a separate API are ways to get a dataset, not new catalogs and not new datasets. In 2024 data.europa.eu had more than 141,000 ATOM feeds: this is how GeoNetwork serves geodata. Whether such an object counts as data is decided not by its form but by whether people working with data need it. What counts as a unit of publication is covered in the chapter on units of publication and formats.

An enterprise catalog. Collibra, DataHub and similar systems describe tables inside a company and are not visible from outside. They are data catalogs, but not public ones; the book does not discuss them, and the boundary is drawn in the chapter What Is a Data Catalog.

A hub. Several catalogs of different types under one banner are not a new type. Each catalog in a hub is classified and assessed separately; see the chapter on hybrids.

Typical mistakes

- **Type by engine.** "We have CKAN, so it is an open data portal." For specialised platforms this is usually true, but research repositories also run on CKAN, and catalogs of nine types run on WordPress. The content determines the type, not the software.
- **Type by name.** The word "portal" or "geoportal" in the name guarantees nothing. A statistical office website becomes an indicator catalog when every table gets its own address and export, as at Statistics Canada, even though it is still called the office's website.
- **Type by owner.** A university runs not only a research repository but also a geoportal and an open data portal. A ministry runs not only an open data portal but also an indicator catalog.
- **Type by the most visible button.** A map on the home page does not make a site a geoportal if it shows tables that are already published.
- **Type by number of datasets.** Millions of "datasets" often turn out to be slices of one database or individual observation records. How this distorts the picture is shown in the chapter The World of Catalogs in Numbers.
- **One type for the whole hub.** A statistical office's indicators and microdata are two catalogs with two standards, even if the domain is the same.

How classifications have changed

Division into types is a working tool, and it has changed along with how many catalogs could be found.

- **2020.** Three big families: open data portals, research repositories and enterprise catalogs. This frame still helps you quickly understand what you are dealing with; it is covered in the chapter What Is a Data Catalog.
- **February 2023.** The catalog of catalogs datacatalogs.ru collected six categories: open data portals, research repositories, statistical indicator databases, geoportals, open budget websites, and portals of reference lists and classifications.
- **May 2023.** The Common Data Index had nine types: open data portals, geoportals, research repositories, microdata and indicator portals, machine learning data catalogs, dataset search engines, marketplaces and API catalogs. Reference data portals were not included: outside Russia there are almost no similar portals, and the type would have consisted only of Russian sites.
- **2026.** The dataportals-registry has eleven main types. Metadata catalogs and dataset lists were added, and generic research repositories and unsorted records are counted separately.

A type is introduced when there are enough catalogs that have to be harvested, described and assessed differently. So the book's classification is not final. If your catalog does not fit any type, describe it with the answers to the five questions: that is enough to find your neighbours and choose a platform.

How to record the classification of your own catalog

Answer the questions of this chapter and write the answers down on a card. It is convenient to keep it next to the decision records from the chapter Justifying a Decision.

```
Type:                what gets its own page
Owner and level:     who runs it, city / region / country / research / other
Where the data is:   stored / linked / descriptions only
Access:              to descriptions, to data, to uploading
Software class:      hosted / ready-made platform / server / CMS / custom build
Subject and area:    general or domain-specific, which territory
Compare with:        three to five catalogs with the same answers
```

Three examples based on the decision chains worked through in Part VI:

	City	University	Statistical office
Type	open data portal	research repository	two catalogs: indicators and microdata
Owner	local government	university	central government body
Where the data is	files in the region's catalog	files in the repository itself	indicators in the catalog database, survey files stored in-house
Access	everything open	descriptions open, some files closed	indicators open, microdata on request
Software class	the region decides: a ready-made platform	a ready-made platform on its own server, next to the publications archive	two ready-made platforms
Compare with	neighbouring cities in the same region	other universities' repositories in re3data	statistical offices of neighbouring countries

The detailed decisions are in the chapter Worked Examples. Next, the answers are used to choose the required capabilities and the platform: this is where the route map begins.

Sources

- dataportals-registry export, release 1.27.0 of 15 October 2026: types, owners, access modes, software classes and platforms by type. The queries are in the appendix.
- re3data Metadata Schema 4.0 and its XML schema: repository types, access to the catalog, data and upload, kinds of restrictions. The Zenodo, Harvard Dataverse and PANGAEA records in re3data, checked 12 October 2026.
- GDS, What's changing on data.gov.uk and why, 25 March 2026; I. Begtin, Good and bad data publishing practices (in Russian), November 2024: broken links on data.gov.uk.
- I. Begtin, Telegram (in Russian): three categories of data catalogs, 14 October 2020; five models of working with metadata, 10 May 2021; datacatalogs.ru categories, 8 February 2023 (repost from the Infoculture channel); reference data portals and Common Data Index types, 22 May 2023; what counts as a dataset: series, layers, ATOM feeds, 25 November 2024; the SPbU publications repository: 17 datasets out of 47,619 records, 25 December 2024.
- The World of Catalogs in Numbers, Comparing Types and Hybrids, Classes of Software, Capability Map

Open Data Portals

An open data portal publishes the datasets of a public body, city or organisation so that anyone can take and use them. A dataset page holds a table, a file, a link to an API. There may be a map next to it, but the main product is the dataset, not the layer.

Three conditions together make data open:

- **machine-readable form**: a file or API that a program reads without manual retyping; CSV, not a scanned table in a PDF;
- **an open licence**: copying, modifying and using the data, including commercially, is allowed with one condition, attribution of the source;
- **access without requests**: no contract, registration or explanation of purpose.

If at least one is missing, it is not yet open data, even if a section of the site is called that. More on the difference in the chapter on geoportals, in the section Publicly available but not open.

According to the dataportals-registry export (release 1.27.0 of 15 October 2026), there are about 6,000 open data portals, 14% of all catalogs. About 5,700 are active.

Who builds them

Almost half of the portals are run by local government: cities, districts, communes.

Owner	Portals	Share
Local government	2,862	48%
Central government	1,105	19%
Regions	818	14%
Universities and research organisations	465	8%
Companies	268	4%
Civil society organisations	264	4%
International organisations	127	2%

A national government usually runs one common portal, sometimes alongside portals of individual agencies. Regions and cities run their own or a shared regional one. International organisations and non-profit projects do the same with their data. The Humanitarian Data Exchange is the open data portal of the UN Office for the Coordination of Humanitarian Affairs, even though its owner is not the government of a single country.

There may be no national portal at all. Turkey had none in 2023, and open data was published by cities: Istanbul, Izmir, Konya. According to registry release 1.27.0 there is still no common portal, only agency ones, such as those of the Ministry of Health and the election commission. In China almost every province has a portal, and at the national level a registration platform for public data

resources has operated since 1 March 2025. It is a registry that connects data owners with those who want to use the data, including for a fee, not an open data portal: by August 2026 more than 300,000 resources, products and services were registered there.

The number of portals is not a measure of openness

By number of portals in the registry the leaders are Russia (572), the United States (565), Spain (485), France (394), Italy (337), Thailand (311), Indonesia (295), Japan (280) and China (227). The list says not where there is more open data, but how publishing is organised locally. In Russia 421 portals out of 572 are the "Open data" section on standard municipal websites. In Italy 153 portals are showcases of one hosted service for communes, in Japan 143 are showcases of a shared catalog of prefectures and municipalities, in Thailand more than 160 are catalogs of agencies and provinces on one government build. Comparing countries by the number of portals is like comparing them by the number of websites. How such portals are built is covered in the following sections.

What they publish

Budgets and procurement, registers of organisations and licences, transport and timetables, addresses, the state of the environment, election results, statistics. The lower the level of government, the shorter and simpler the list. In 2024 two standard open data pages of districts in the Kaluga region had two and seven datasets: registers of schools, kindergartens, libraries and sports organisations, the structure of the administration. Italian communes on the Municipium service publish dozens of datasets that the service itself exports from their record-keeping systems: families by street, births, marriages, foreigners by citizenship.

The content is set by mandatory lists: in the EU the regulation on high-value datasets, in the US the OPEN Government Data Act, in Russia Government Order No. 1187-r. Their content is covered in the chapter What Catalogs Publish. The lists show what the state considers the minimum, but not what users need.

EU national portals contain a lot of geodata. On 12 October 2026 the common European portal data.europa.eu showed 1.73 million datasets. The most common format among them is a WMS service (511,000 datasets), followed by WFS (363,000), with CSV only in third place (275,000). The largest source is the German spatial data infrastructure GDI-DE: 681,000 datasets. The boundary with a geoportal is blurred here: the national open data portal harvests layer catalogs and shows them next to tables. About 28,000 are marked as high-value datasets, 1.6% of the whole portal.

How they are built

On the outside portals look alike: search, a list of datasets, a dataset page. Inside they are built differently, and the structure determines which features are needed. Five models are the most common.

One platform for everyone

Agencies, cities, companies and civil society organisations publish on one site, each under its own organisation. The model is the French data.gouv.fr. On 12 October 2026 it had 74,400 datasets, 1,265 described APIs, 6,800 organisations and 81,700 registered users. The portal does more than accept publications: 507 configured harvesters collect descriptions from other catalogs. It rests on its community: 17,700 dataset discussions and about 6,000 reuses (apps, articles and studies that users themselves linked to datasets).

This model needs roles and organisations, moderation, discussions, accounts for external publishers and a section for reuses.

Harvester

The portal stores almost nothing itself; it harvests descriptions from agency and regional catalogs and offers a common search over them. The US data.gov collects the `data.json` files published by federal agencies, as well as the catalogs of states, cities and universities: on 12 October 2026 it held 590,000 datasets. The European data.europa.eu harvests national portals. In 2022 the Indonesian national portal Satu Data harvested regional portals on CKAN; in the registry, 196 out of 295 Indonesian portals run on CKAN.

A harvester needs harvesting via DCAT and other protocols, strong search and a link to the original source on every dataset page. It hardly needs its own files. How harvesting chains work and where they break is covered in the chapter Federation and Harvesting.

A standard portal for everyone

The state or a vendor hands out the same build to dozens or hundreds of public bodies. Each has its own showcase, but the software, the description fields and the export are the same. This is the answer to the main problem at the local level: a small town has no people to run its own portal.

- **Thailand.** The Government Data Catalog (GD Catalog) is a central catalog plus catalogs of agencies and provinces on `*.gdcatalog.go.th` subdomains, all on CKAN with a shared Thai extension. The registry lists 164 such catalogs. In October 2026 the central catalog showed about 25,000 datasets, 16,400 of them statistics.
- **Japan.** The BODIK ODCS service is run by the research institute ISIT in Fukuoka. Behind the showcases of prefectures and municipalities stands one shared CKAN, data.bodik.jp: in October 2026 it had 386 organisations and 18,500 datasets.
- **Italy.** Municipium is Maggioli's hosted platform for commune websites and apps, and the open data portal is part of it. Datasets are exported automatically from the commune's record-keeping systems, and the descriptions are passed in the Italian DCAT-AP_IT profile to the national portal dati.gov.it. In October 2026 the vendor's shared CKAN catalog contained 6,200 datasets from 193 organisations.
- **Argentina.** Andino is a CKAN build released by the country's government: it is packaged in Docker, easy to install and serves descriptions in the national metadata profile out of the box. The registry lists 49 portals on Andino.

- **Russia.** Gosweb is the state platform for government websites. The "Open data" section on it is a page of the site with a list of datasets, where each dataset is three CSV files: a passport, a structure and the data itself. This is what Russian methodological guidelines required. The section has no API and no DCAT export, and in the registry none of the 421 such portals has a programming interface marked.

The first four examples show that a standard portal works when it has shared storage or a shared metadata export, and datasets reach the higher level without manual work. The Russian variant gives identical pages but no shared catalog: you cannot use them to find the datasets of all municipalities at once.

Database inside, exports outside

The data lives in a database or an information system, and the portal serves snapshots of it in several formats and through an API. Datasets update themselves when the database changes. Such portals cost more than "file dumps" but are more mature. Moscow's data.mos.ru works this way. So do the portals of Chinese provinces, with an emphasis on real-time APIs: in July 2025 the portal of Guangdong province offered more than 98,000 datasets and more than 6,700 APIs. But there were no open licences, and data was released only after registration and identity verification. By the three conditions at the start of the chapter, this is publicly available rather than open data. On 12 October 2026 the portal did not open from abroad.

This model needs APIs, automatic updating of datasets from record-keeping systems, and field documentation. Manual file upload is secondary for it.

Curated collections

The portal stops being a warehouse of everything and shows selected, checked datasets. In March 2026 the British data.gov.uk rebuilt its home page: it now has thematic collections and guidance on working with data. The previous catalog remained as a separate section, and the service promised not to change it for now. In the summer of 2026 the home page was named the National Data Library. The service itself named the reason for the rebuild: more than a quarter of the links in the catalog led to an error, and many datasets had not been updated for years.

This model needs editorial work, link checking and quality indicators on datasets. It is the answer to the "dump" that a portal turns into when everyone publishes whatever they want and nobody cleans up.

What follows from this

The models combine: data.gouv.fr is both a platform and a harvester, and the Italian standard portal has the national harvester dati.gov.it above it. But the main model determines the main features:

Model	What you cannot do without	What you can skip
One platform	roles and organisations, moderation, file storage	harvesting other catalogs at the start
Harvester	harvesting via DCAT, search, a link to the original source	your own file storage

Model	What you cannot do without	What you can skip
Standard portal	a shared field schema, export upwards, centralised upgrades	a custom design for each showcase
Database and exports	APIs, automatic updates, a data dictionary	manual upload
Curated collections	editorial work, link checking, quality indicators	open publisher registration

How to get from these features to a platform is covered in the chapters Selecting Features and From Features to a Platform.

Standards

Descriptions are served in DCAT. In the EU this is the DCAT-AP profile (the current recommendation is 3.0.1 of 27 October 2025) and national refinements of it, such as the Italian DCAT-AP_IT. For high-value datasets there is a separate DCAT-AP HVD profile (release 3.0.0, October 2024). In the US, agencies publish a `data.json` file following the DCAT-US profile. The new version, DCAT-US 3.0, is built as a profile of W3C DCAT 3, so catalogs abroad understand it too. data.gov harvests the old version 1.1 during a transition period, and new catalogs are advised to use 3.0 from the start. In Russia, instead of DCAT, there is a passport defined by the methodological guidelines version 4.0; how it works is covered in the chapter on metadata standards. Search engines also need schema.org markup on the dataset page. More in the chapter on metadata export.

Programmatic access is specific to each platform: the Action API in CKAN, the udata API, the Explore API in OpenDataSoft, SODA in Socrata. There is no single standard for searching datasets, so a higher-level portal harvests descriptions via DCAT rather than through each platform's API. For high-value datasets in the EU an API is mandatory, and bulk download where appropriate. How to check them on a live site is covered in the chapter on APIs and bulk download.

Licences are CC BY 4.0, ODbL or national open licences such as the British Open Government Licence and the French Licence Ouverte. Russian portals publish "standard terms of use for publicly available information": in substance they allow free use, but international catalogs do not know this text, and software does not recognise it as an open licence. The licence must be on every dataset, not one line in the site footer. More in the chapter on licences and access.

Platforms

Platform	Portals in the registry	What it is	DCAT and API
CKAN	1,906	open-source platform	API in the core, DCAT via the ckanext-dcat extension
Custom build	1,144	—	API in only 19%, DCAT rarely
OpenDataSoft (Huwise since 2025)	454	hosted service	in the service
Gosweb	421	the Russian state platform for government websites	a CSV list on a page, no API

Platform	Portals in the registry	What it is	DCAT and API
Socrata (Tyler Data & Insights)	245	hosted service	in the service
DKAN	161	open-source platform on Drupal	in the core
Municipium	153	Maggioli's hosted platform for Italian communes	DCAT-AP_IT for the national portal
BODIK ODCS	143	a service for Japanese municipalities	the API of the shared CKAN
Liferay	117	content management system, almost all in Spain	as configured
EntryScape	99	commercial platform built on DCAT, mostly Sweden	in the core

Less common are Andino (49), JKAN (48), Datasette (44), Data Fair (32, mostly France), Piveau (19) and udata (15, data.gouv.fr runs on it). ArcGIS Hub is recorded as an open data portal in the registry only 19 times: it is much more often classed as a geoportal. It is a hybrid, covered in the chapter on comparison. Profiles of the main platforms are in the part Platforms and Extensions.

National and city portals choose differently. Among central government portals the leaders are CKAN (420) and custom builds (365), and there are few hosted services. Local government also puts CKAN first (707), but next to it are Gosweb (421), custom builds (307), OpenDataSoft (216), Municipium (153), Socrata (144) and BODIK (127). At the local level the platform is more often chosen not by the city itself but by the state, the region or a vendor that already serves the neighbours. Just as with geoportals, global platforms lead in the overall count, while within a country municipalities are split among one or two local vendors.

Open-source platforms and hosted services almost always provide an API: in the registry it is marked for 95% of CKAN portals and 98–100% on OpenDataSoft and Socrata. For custom builds only 19%. This is the main argument against a home-grown portal: the interface that a ready-made platform gives out of the box has to be commissioned separately for a custom build, and it is often forgotten.

Examples

National portals

- data.gouv.fr: France, its own open-source platform udata. The model of a platform with a community: agencies, cities, companies and civic projects publish, and datasets have discussions and reuses.
- data.gov: United States, a harvester. It ran on CKAN from 2013 (the launch of data.gov 2.0), and in April 2026 moved to a custom build: the harvester writes to a shared database, and the front end only reads it. Dataset page addresses were preserved. The old CKAN Action API at its previous address returns an error, and the new interface is described in OpenAPI. The old catalog was temporarily kept at catalog-old.data.gov, but on 12 October 2026 that address no longer opened. An example that even a CKAN flagship can leave the platform, and that the API other people's programs relied on changes along with the platform.

- data.gov.uk: United Kingdom. Since March 2026 the home page shows curated collections, since the summer of 2026 it has been called the National Data Library, and the previous catalog still works. An example of moving from a warehouse to curated collections.
- data.gov.au: Australia, CKAN. From 2019 search and the front end were provided by the Australian platform Magda on top of an outdated CKAN. In June 2025 the portal was moved entirely to CKAN (2.11.6 as of October 2026), and the pages to Drupal. Most datasets there come from the national geological survey.
- open.canada.ca: Canada, CKAN, 48,000 datasets in October 2026.

Standard and local portals

- GD Catalog: Thailand, a central catalog and catalogs of agencies and provinces on one CKAN build.
- BODIK ODCS: Japan, municipal showcases on top of the shared CKAN data.bodik.jp.
- Istanbul Open Data Portal: CKAN. A city publishing data in a country without a national portal: 286 datasets in 2023, 558 in October 2026.

International

- Humanitarian Data Exchange: CKAN 2.11.4 with the ckanext-dcat extension, 28,000 datasets in October 2026. UN agencies, governments and civil society organisations publish there.

Russia

- data.mos.ru: Moscow's own platform and an example of the "database inside, exports outside" model. One of the few live Russian portals, but it lives on inertia: in early 2022 there had been no new datasets since 2020, although some registers were updated daily. In March 2026 the portal was closed for about two weeks for "technical works" without explanation.
- data.gov.ru: the national portal. In 2023 it was closed "for migration to a new platform" and reopened in the summer of 2025 without the previous data. What was on it before and after is covered in the chapters on self-audit and on archiving.

Where it breaks

A showcase for a deadline. The portal is launched for a date or a report and then not updated. From the outside it looks alive: the home page opens, search works, but the datasets have been the same for years. The signs of such a portal are in the chapter on metrics.

Links instead of files. A dataset page leads to a file on the agency's website, and a year later the file is gone. On data.gov.uk by 2026 more than a quarter of the links led to an error, and this became the reason for the rebuild. The argument for "upload, don't link" is in the chapter on data storage.

A counter instead of data. If a portal is given the target "publish N datasets", one series is cut into dozens of files by year and district, and five-row tables count towards it. The number of datasets grows, the data does not. See the anti-pattern Splitting to inflate the counter.

A licence for the whole site. The terms of use sit in the footer and are not stated on the dataset. The harvester takes a dataset page without a licence, and on the higher-level portal the dataset already looks closed.

A home-grown list format. To collect datasets from thousands of websites, people invent their own schema instead of DCAT. This happened with the Russian dataset lists: the country harvested them, but not a single foreign aggregator read them. More in the anti-pattern A home-grown format instead of a standard.

The door is closed. Registration, identity verification, API limits or blocking requests from other countries. Each of these conditions knocks the portal out of search engines and aggregators. On 12 October 2026 data.gov.ru, the portals of Chinese provinces and standard municipal websites on Gosweb did not open from a foreign address. See the anti-pattern The door is closed to programs.

A custom build without an API. Only 19% of home-grown portals have a programming interface marked. A national or regional portal without a DCAT export cannot harvest the level below and will not itself get to the level above.

Migration and closure. About 4% of open data portals in the registry are no longer active, a higher share than among geoportals and research repositories. Portals move from platform to platform, and in the move addresses, APIs and archives are lost. Examples of migrations in both directions are in the chapter on the lifecycle.

How to check an open data portal

Five questions you can answer in a quarter of an hour without knowing what the portal is built on.

1. **Does the dataset have a licence?** Open three random dataset pages. The licence should be on each one, not only in the site footer.
2. **Are datasets updated?** Sort the list by modification date. If the freshest records are more than a year old, the portal lives on inertia.
3. **Can you download a file directly?** The link should lead to a file or an API, not to an agency page where you still have to find the file.
4. **Does the portal serve descriptions to programs?** Typical addresses:

```
# CKAN: platform version and list of extensions
curl -s "https://data.humdata.org/api/3/action/status_show"
# CKAN with ckanext-dcat: the catalog in DCAT
curl -sI "https://data.humdata.org/catalog.ttl"
# udata: portal summary
curl -s "https://www.data.gouv.fr/api/1/site/"
# OpenDataSoft: number of datasets
curl -s "https://opendata.paris.fr/api/explore/v2.1/catalog/datasets?limit=0"
# US: an agency's data.json file in DCAT-US
curl -sI "https://www.energy.gov/data.json"
```

5. **Is the portal visible from above?** Find its organisation on the national portal or on the common European data.europa.eu. If its datasets are not there, or there are noticeably fewer, harvesting does not work.

A detailed list is in the checklists, and commands for other platforms are in the chapters on metadata export and on APIs.

When you do not need your own portal

For a small town or an agency with a couple of dozen datasets, a portal of its own is often harmful: there is nobody to update it. It is better to publish on a regional or national portal, join a standard portal as Japanese and Italian municipalities do, or publish the datasets as a static catalog with a metadata file that the higher-level portal will harvest. Such a decision is worked through in the municipality example, and ways of publishing without a portal are in the part Publishing Without a Portal.

How it differs from its neighbours

Neighbour	Main difference
Geoportal	the main product is a layer and a service, not a dataset. If the site has nothing to show without WMS and WFS, it is a geoportal
Indicator catalog	files and tables are published, not series with breakdowns from which a selection is built
Research repository	regular updates matter more than an immutable version with a DOI that has been cited
Metadata catalog	the portal serves the data itself, not only descriptions or reference lists
Marketplace	data is given away free of charge and without a contract
An "Open data" section on a website	a portal has search, dataset pages and a metadata export; a section usually has only a list of files

A summary table of all types is in the chapter Comparing Types and Hybrids.

Sources

- dataportals-registry export, release 1.27.0 of 15 October 2026: owners, countries and platforms of open data portals, including separately for central and local government, the share with an API and the share inactive. The queries are in the appendix.
- CKAN: core and extensions, platform profiles
- Announcement of the new data.gov catalog, 15 April 2026; description of the catalog application and its API description. Number of datasets, behaviour of old dataset addresses and the Action API, unavailability of catalog-old.data.gov: checked 12 October 2026.
- DCAT-US 3.0 and migration from version 1.1; DCAT-AP releases, including DCAT-AP HVD 3.0.0
- GDS, What's changing on data.gov.uk and why, 25 March 2026; the data.gov.uk home page in the Web Archive on 1 April 2026, not yet named National Data Library, and on 2 July 2026, already with that name
- data.gouv.fr: portal summary via the API; data.europa.eu: dataset search via the API: formats, sources and the high-value dataset flag; open.canada.ca; HDX; Istanbul portal; checked 12 October 2026

- Standard portals: GD Catalog and the example of the Chaiyaphum province catalog; BODIK ODCS and the shared CKAN data.bodik.jp; Municipium, the description of the Bussero commune portal on Municipium (in Italian; automatic export, DCAT-AP_IT, transfer to dati.gov.it) and Maggioli's shared catalog; Andino; checked 12 October 2026
- The "Open data" section on Gosweb: Web Archive snapshots of the pages of Borovsky district, 14 September 2024, and Kozelsky district, 12 August 2024. From a foreign address on 12 October 2026 the sites did not open.
- China's national registration platform for public data resources: launch announcement (in Chinese), 1 March 2025; more than 300,000 registered resources (in Chinese), 25 August 2026
- I. Begtin, Telegram (in Russian): three functions of portals, 27 March 2023; data.mos.ru and portals with a database inside, 25 October 2021; data.mos.ru: "the inertia scenario", 17 January 2022; data.mos.ru unavailable in 2026, 5 March 2026; data.gouv.fr, 2 April 2025; open data in Turkey, 7 February 2023; Chinese provincial portals, 24 April 2023; the Guangdong province portal, 29 July 2025; open data in the EU, 18 December 2024; on renaming data.gov.uk, 5 June 2026; on data.gov.au returning to CKAN, 14 November 2025
- Link Digital, moving data.gov.au to CKAN; Magda, which ran on data.gov.au from 2019 to June 2025. The CKAN version is from the `/data/api/3/action/status_show` response, October 2026

Geoportals

A geoportal describes spatial data, that is, data tied to a place: layers, maps, imagery, services. Every object in it has coordinates or a boundary. There is almost always a map in the browser next to it. Often the portal is the visible part of a country's or region's spatial data infrastructure: a shared system of rules, catalogs and services to which public bodies contribute their layers.

According to the dataportals-registry export (release 1.27.0 of 15 October 2026), there are 24,800 records of this type, 57% of the registry. They include catalogs of descriptions, layer servers and municipal maps. ArcGIS Server (4,674) and GeoServer (1,460) serve maps; on their own they are not yet a catalog where a layer is found by its description. To avoid confusion, it is worth first looking at what a geoportal consists of.

What a geoportal consists of

The word "geoportal" hides three different things. Sometimes they are assembled on one site, sometimes they live separately.

- **The viewer** is a web map. It shows layers, lets you switch them on and off, search for an address, measure a distance. This is what a person sees first.
- **The catalog** is the layer records: what is depicted, who made it, when, in which coordinate reference system, with what accuracy, on what terms. A layer is found by its record without knowing its address.
- **The services** are a map server that serves a layer to programs: as an image for viewing, as features for download, as ready-made map tiles.

Part	What it does	Standards	Example software
Viewer	shows layers on a map	—	Oskari, TerriaJS, QGIS Web Client, Lizmap, Masterportal
Catalog	describes layers and searches the descriptions	ISO 19115, CSW, OGC API Records, DCAT	GeoNetwork, pycsw, ArcGIS Hub
Services	serves the data itself	WMS, WMTS, WFS, OGC API Features, ArcGIS REST, STAC	GeoServer, MapServer, QGIS Server, ArcGIS Server

There are also all-in-one platforms: GeoNode, ArcGIS Online with ArcGIS Hub, the Russian NextGIS Web.

Only a catalog combined with a service that allows download makes a layer open data. A viewer without them is a picture. A server without a catalog means only someone who knows the address will find the layer. A catalog without a service is a description without data. So both when choosing a platform and when checking someone else's geoportal, the first question is the same: which of the three parts are here.

Who builds them

National mapping and cadastral agencies, regions and especially cities. The geoportal is the most "local" type of catalog: a city may have no open data portal, but it almost always has a map of the master plan, addresses and land parcels, because the administration's own staff work with it every day.

Owner	Geoportals	Share
Local government	14,440	58%
Regions	3,534	14%
Central government	2,752	11%
Universities and research organisations	1,467	6%
Companies	1,264	5%
Civil society organisations	601	2%
International organisations	392	2%
Other	302	1%

Research and conservation organisations run catalogs of observations and imagery, companies run catalogs of their products and open layers commissioned by the government, and international organisations run catalogs for their programmes, for example UN agencies and the European Environment Agency.

The number of geoportals is not a measure of geodata

By number of geoportals in the registry the leaders are the United States (5,050), Poland (2,303), the Czech Republic (1,408), Italy (1,179), Canada (1,027), Germany (964), France (899) and Japan (652). As with open data portals, the list says something about how the market is organised, not about the volume of data. In the US the count comes from ArcGIS Server and ArcGIS Hub: almost 4,500 records. In Poland it comes from e-mapa, which more than half of the gminas use. In the Czech Republic from three hosted services for municipalities. Germany is only sixth, even though its common geodata catalog is one of the largest in the world: instead of a thousand websites it has one infrastructure that harvests the descriptions of all levels of government.

Where the geoportal is the main catalog

In developing countries the geoportal is often the main, and sometimes the only, catalog of government data. According to the dataportals-registry export (release 1.27.0 of 15 October 2026), in 47 countries and territories there is at least one active geoportal and not a single active open data portal. Egypt (16), Nicaragua (12), Guyana, Somalia and Greenland (10 each) have the most such geoportals. The explanation is simple: the value of geodata is the easiest to prove, as agencies, businesses and international programmes use it. So a geoportal is built before an open data portal, often with donor money and on the open-source GeoNode.

It also happens that an open data portal exists, but the data lives next to it. In Kazakhstan, according to a 2024 assessment, truly open data was published not by the data.egov.kz portal but by the national geoportal map.gov.kz on GeoNode. On 12 October 2026 its API returned

descriptions of 378 datasets without login.

What they publish

Base maps and topography, cadastre and parcel boundaries, addresses, spatial planning and zoning, orthophotos (aerial and satellite images rectified to map scale), utility networks, roads, protected areas, flood zones, geology, Earth imagery, scanned historical maps.

A geoportal has more than one unit of publication:

- a **layer**: one theme for one territory, for example the city's parcel boundaries;
- a **service**: several layers at one address, for example all the layers of a master plan;
- a **map or project**: a saved set of layers with styles and labels;
- an **imagery collection**: a series of images of one territory on different dates.

The same layer can simultaneously be a record in the catalog, a WMS and WFS layer on the server, and part of several maps in the viewer. That is why the number of "datasets" in two geoportals is almost never comparable: one counts records, another server layers, a third projects. How to count is covered in the chapter Units of Publication and Formats.

How they are built

On the outside geoportals look alike: a map, a list of layers, search. Inside they are built differently, and the structure determines which features are needed. Five models are the most common.

National infrastructure

Agencies, regions and cities run their own catalogs and services, the national catalog harvests their records via CSW and offers a common search, and the national geoportal shows the layers on one map. The model is Germany. Its common catalog Geodatenkatalog.de runs on GeoNetwork: on 12 October 2026 it held 814,000 records. From it the European portal data.europa.eu harvests 681,000 datasets, more than from any other source. The showcase for people is Geoportal.de.

This model needs harvesting of other catalogs, a record editor following the national profile, validation of records against the rules and monitoring of whether services respond.

An organisation's open geodata catalog

A city or agency publishes layers with records, download and a licence: in effect an open data portal where the main product is a layer. Most often this is ArcGIS Hub, a site on top of the ArcGIS Online cloud where each layer can be downloaded in several formats. A Hub site serves its own catalog in DCAT-US 1.1 and DCAT-AP 2.1.1 and offers search via OGC API Records, for example the Washington, DC open data portal opendata.dc.gov; DCAT-US 3.0 and DCAT-AP 3.0.0 feeds are enabled separately. In the registry such exports are marked for about 3,200 Hub sites. The open-source substitute is GeoNode: it serves records via CSW, but not in DCAT.

This model needs download in open formats, a licence on every layer and a catalog export in DCAT so that the open data portal can harvest it.

A vendor's municipal map

A hosted service from a local company that dozens or hundreds of municipalities subscribe to: parcels, the master plan, addresses, building permits. The main product is a map for residents and officials. The layer is visible, but often it cannot be downloaded. Apart from ArcGIS Hub and e-mapa, the registry has 132 such services, running about 5,300 geoportals, and a programming interface is marked for only 21% of them. More on vendors in the section on platforms below.

This model needs almost nothing except the map. But if a municipality wants open data, layer export (WFS or files) and a licence must be written into the contract with the vendor: the service will not add them by itself.

A layer server without a catalog

ArcGIS Server or GeoServer that serve layers to programs but keep no records for search. Often this is not a separate portal but the "back end" of the other models: the city map and its catalog rely on the server. If there is no catalog next to it, such a server is found only by geoservice search engines and by those who know the address.

It needs a catalog, its own or a higher-level one, where the layers are entered with descriptions.

An imagery and file catalog

Collections of satellite imagery, elevation, climate and other spatial files, described in STAC. This is not a city map but a store for programs: the files sit in cloud storage, and the catalog answers which images exist for this territory on these dates. An example is Earth Search by Element 84, a catalog of open datasets on AWS. The registry has about 140 STAC servers and about 80 STAC Browser viewers.

It needs search by territory and date, storage of large files and delivery via direct links. A map for people is secondary.

What follows from this

The models combine: a national infrastructure has hundreds of layer servers at the bottom, a Hub site has ArcGIS Server under the hood. But the main model determines the main features:

Model	What you cannot do without	What you can skip
National infrastructure	harvesting via CSW, a record profile, record validation, service monitoring	your own data storage
Open geodata catalog	download, a licence on the layer, DCAT export	harvesting other catalogs
Municipal map	a viewer, address and parcel search	an ISO catalog, if the layers are described at a higher level
Layer server	WMS, WFS, registration in a catalog	your own viewer
Imagery catalog	STAC, search by territory and date, cloud storage	a map for people

How to get from features to a platform is covered in the chapters Selecting Features and From Features to a Platform.

Standards

A layer description uses ISO 19115 and its XML encoding ISO 19139. The new edition of the standard is encoded in XML according to ISO 19115-3, but catalogs still more often exchange ISO 19139 records. The record contains what an ordinary dataset does not have: the territory as a coordinate bounding box, the coordinate reference system, the scale or resolution, the lineage of the data.

The catalog is served to programs via the CSW service. It is being replaced by OGC API Records: Part 1 was approved as an OGC standard in April 2025. It is the same catalog, but in the form of an ordinary web interface with JSON responses that a developer understands without knowing CSW. ArcGIS Hub sites and the lightweight pycsw catalog already serve it.

The bridge to open data portals is the GeoDCAT-AP profile. It describes how to translate an ISO record into DCAT, which open data portals run on. The current version is 3.1.0, a SEMIC recommendation of 16 February 2026. This is exactly how data.europa.eu reads national geocatalogs. GeoNetwork 4.4 can serve records via CSW directly in DCAT: in October 2026 the European Environment Agency's catalog offered GeoDCAT-AP, DCAT-AP and the high-value datasets profile.

The data itself is served with WMS and WMTS services for viewing, and WFS and OGC API Features for downloading features. A map image (WMS) is not yet data: you cannot get parcel boundaries from it. Only WFS, OGC API Features, files or ArcGIS export let you download features.

ArcGIS REST is a separate language for the same task. It is widespread not because it is an ISO standard but because a large share of portals are built on it. GIS software and many harvesters read it on a par with OGC standards.

Coordinate reference system. Each layer has its own, and it is specified with a code from the EPSG registry. World maps use WGS 84 (EPSG:4326) and Web Mercator (EPSG:3857), national data uses its own systems: S-JTSK in the Czech Republic and Slovakia, the local MSK coordinate systems for each region in the Russian cadastre. A layer without a CRS code has to be guessed.

STAC is a protocol for imagery and, following it, for other georeferenced file collections. It is a protocol, not a program. It is served by dedicated STAC servers, by static sets of JSON files, and also by general-purpose platforms: GeoServer has a STAC community module. That is why there are more STAC catalogs in the world than lists such as STAC Index show. The unit of publication in STAC is unusual for a portal: a collection is a series of images of one territory on different dates, with thousands or millions of files inside. Counting each file as a dataset is wrong, and showing a collection as an ordinary dataset page with a list of files is impossible. Aggregators that harvest STAC together with open data portals each solve this in their own way.

INSPIRE

Since 2007 the EU INSPIRE Directive has required public bodies in EU countries to describe spatial data according to common rules and to provide catalogs and view and download services. National infrastructures like the German one are built on it, and GeoNetwork has become the standard for national catalogs in the EU.

INSPIRE is now being simplified. In December 2025 the European Commission proposed removing from the directive the technical requirements that duplicate later legislation, namely the Open Data Directive and the high-value datasets regulation. The EU Council adopted its negotiating position in June 2026, the European Parliament on 6 October 2026. Parliament wants to keep open ISO and OGC standards mandatory, as well as the network of services. There is no final text yet.

One change has already happened: on 1 July 2026 the INSPIRE Geoportal was closed. Its address redirects to data.europa.eu, where geodata is now searched together with the rest of the EU's open data, and the INSPIRE registers and code lists have moved to the EU Publications Office's EU Vocabularies. The practical conclusion for a European geoportal: the main path upwards is now descriptions in DCAT and GeoDCAT-AP, not only a CSW catalog.

Platforms

Only 4% of geoportals are home-grown. This is the most "platform-based" type, but also the most varied: the registry lists about 350 different programs among geoportals.

Platform	Geoportals in the registry	What it is	Role
ArcGIS Server	4,674	Esri's commercial server	services
ArcGIS Hub	3,464	Esri's hosted service	open geodata catalog
e-mapa	1,927	Geo-System's hosted service, Poland	municipal map
GeoServer	1,460	open-source server	services
Custom build	1,043	—	any; API in only 11%
GeoNode	770	open-source platform	all in one
GeoNetwork	762	open-source catalog	catalog and harvester
GISPLAN	467	T-MAPY's hosted service, Czech Republic and Slovakia	municipal map, no API
Lizmap	346	open-source platform on QGIS Server, France and Italy	viewer and services

GeoServer and its kind serve layers. GeoNetwork and ArcGIS Hub describe them so that a layer can be found. One without the other is either a map without a catalog or a catalog without data.

Three open-source platforms

The three open-source platforms in the table have different roles, and they are often confused when choosing.

- **GeoNetwork** is a catalog of descriptions and an aggregator. It stores layer records in ISO 19115, serves them via CSW and itself harvests records from other sources: CSW and OAI-PMH catalogs, WFS and THREDDS services, STAC collections. It has a record editor that follows the INSPIRE rules, so many national catalogs in the EU run on GeoNetwork, for example the Estonian metadata.geoportaal.ee, the German Geodatenkatalog.de and the European Environment Agency catalog. It usually does not store the data itself but links to services. It is most accurately thought of as a search engine for the layers of a country or a sector.

- **GeoNode** is closest to an ordinary data catalog. A user uploads a file, and it becomes a layer with a record, a map for viewing, a download link and an API. Inside, GeoServer serves the layer and pycsw serves the records via CSW. GeoNode is chosen when data is published by hand and one all-in-one site is needed.
- **GeoServer** serves layers via WMS, WFS and other OGC protocols. Its layer description is short: a title, an abstract, keywords and a link to an external record. The INSPIRE extension adds the fields required by the directive to the service response, but GeoServer does not keep a full ISO record: that is held in GeoNetwork or another catalog. A server without a catalog next to it is a map without descriptions: only someone who knows the address will find the layer.

Alongside them are showcases built on top of other catalogs. The Finnish Oskari, the Australian TerriaJS and QGIS Web Client (QWC2, 159 installations in the registry, mostly in Finland and Germany) show layers from many servers on one map but hold almost no data of their own. GeoBlacklight is a search across university library collections of maps and geodata, pycsw a lightweight catalog of descriptions. A short note on each is in the platform profiles.

Local vendors

e-mapa shows the path most municipal geoportals take. It is a commercial product of the Polish company Geo-System: each gmina buys it itself, but one vendor covers most of the country, so the portals look the same and can pass data to the national registers of addresses and plans. This is how the market works in many countries: global platforms lead in the overall count, while within a country municipalities are served by one or two local vendors.

Country	Main platforms of municipal geoportals	Geoportals in the registry
Poland	e-mapa (Geo-System)	1,927
Czech Republic	GISPLAN (T-MAPY), GisOnline, Mapotip	424, 287, 261
Slovakia	mOBEC (T-MAPY), Geodeticca	177, 159
Slovenia	iObčina (Kaliopa)	213
Italy	GisMaster (Technical Design)	274
Austria and Romania	Map2Web (Schubert & Franzke)	114 and 159
Japan	Waga Machi Guide (PASCO), SonicWeb	259, 124
Finland	Louhi (Sitowise)	117
Australia	IntraMaps (TechnologyOne), Pozi	111, 80
Switzerland	GeoMapFish (open source)	223
France	Lizmap (open source), mviewer (open source)	156, 94
Germany	Masterportal (open source)	85
China	Tianditu (state-run)	100

In Slovenia, which has 212 municipalities, iObčina has more than two hundred geoportals: municipalities lease the system under contract, and the vendor's team maintains the data. The difference in approach is visible too. In Central Europe and Japan municipalities lease a commercial hosted service, and most such services have no programming interface in the registry at all: of 132 local vendor services, 104 do not have one marked on a single site, including

GISPLAN, GisMaster, iObčina and Waga Machi Guide. In France, Switzerland and Germany open-source software developed by a public body or a community is more common: Masterportal was created by Hamburg's geospatial agency, mviewer by the Brittany region. Such software serves layers using OGC standards, and a programming interface is marked for 84% of Masterportal installations and 99–100% of Lizmap, GeoMapFish and mviewer installations.

In Russia, of the registry's 286 geoportals, 60 are custom builds, 37 run on NextGIS Web, 27 on ArcGIS Server, 25 on ActiveMap GIS; GeoServer runs on only 22. Geoservice search engines such as GeoSeer and Spatineo see only standard OGC and ArcGIS interfaces and therefore cover no more than half of such portals.

The hidden ArcGIS catalog

ArcGIS has a large catalog that few people know about. Any organisation in the ArcGIS Online cloud can make its layers, maps, apps and files public, and all of this ends up in the common ArcGIS search. ArcGIS Hub search is narrower: it searches open data sites built on Hub. In 2024 the size of the common search was estimated at hundreds of thousands of items; in 2025 Hub had more than 25,000 sites and about 300,000 layers, maps and documents. A more precise count is impossible: ArcGIS search stops counting results at 10,000.

It is a useful but poor catalog. An item often has only a title and tags, without a description, licence or update date. On the other hand, it contains layers for countries that have almost no geodata of their own online: international projects, universities and individual specialists publish them. Many sites at `*.maps.arcgis.com` addresses do not say that layers can be downloaded, even though the server serves them to any program. How to check this is in the section on implicit interfaces.

Examples

National and international

- data.europa.eu: since 1 July 2026 the single point for searching EU geodata, replacing the closed INSPIRE Geoportal. It harvests national geocatalogs, the largest of which is the German one.
- Geoportal.de and Geodatenkatalog.de: the showcase and the common catalog of Germany's spatial data infrastructure.
- mapy.geoportal.gov.pl: Poland's national geoportal.
- European Environment Agency catalog: GeoNetwork 4.4, about 3,400 records, served in ISO and in DCAT.

Cities

- opendata.dc.gov: Washington, DC, ArcGIS Hub: layer download, a DCAT-US catalog, search via OGC API Records.
- Helsinki map service: the city's open layers on GeoServer; in October 2026 about three hundred layers could be downloaded via WFS.

Imagery and files

- Earth Search: a STAC catalog of open satellite data on AWS.
- Google Earth Engine Data Catalog: imagery and other layers as a STAC catalog, not a city map.

Russia and Kazakhstan

- Russian Geographical Society Geoportal: a catalog of scanned historical maps, plans and atlases from the society's map collection, which has more than 48,000 items. The collections are growing: the latest were added on 9 October 2026.
- Geoportal of the Republic of Mordovia: a regional geoportal of the Russian Geographical Society with thematic layers of settlements, landslides and protected areas.
- map.gov.kz: Kazakhstan's national geoportal on GeoNode, in effect the country's main source of open government data.

Where it breaks

A map without download. The layer is visible on the map, but there is no export: the service serves only a WMS image. For a resident that is enough; for someone who wants to do calculations or overlay the layer on their own data, it is not.

Records for the sake of it. ISO fields are filled with dashes, the record creation date is given instead of the data date, and instead of a licence there is "no restrictions" as free text in the terms of use field. A harvester does not recognise such a licence, and on the higher-level portal the layer looks closed.

The catalog and the service diverge. The service returns an error or has moved, while the record still points to the old address and does not show it. The Arkhangelsk region geoportal lost its layers when its ArcGIS licence expired, while the catalog continued to show their records.

A coordinate reference system without a code. The layer is served in a local CRS without its code. It can be combined with other data only by hand, if at all.

One vendor, hundreds of sites. Nobody updates the long tail of identical municipal portals one by one: when the platform breaks, hundreds of sites break. When the vendor leaves or the contract ends, all the maps disappear at once.

An open-source platform without maintenance. Someone has to update your own server with GeoNode, GeoNetwork or GeoServer. In the registry 11% of geoportals on GeoNode and about 7.5% on GeoServer and GeoNetwork are inactive, against 1.5–2.3% on ArcGIS Hub and ArcGIS Server. A common story: the geoportal was built under a donor programme, the programme ended, the server was switched off.

Publicly available but not open

A layer that everyone can see is not yet open data. Machine-readable export, a clear licence and access without requests make it open. Many government geoportals only allow viewing. Russian ones are a clear example.

- **Rosreestr's National Spatial Data System** (nspd.gov.ru). Since 23 December 2024 the public cadastral map has run on it: viewing is free and without login. There is no layer export in the interface. In January 2024 a GeoServer with more than 384 layers was found inside the portal: WMS viewing was open, and the key to WFS export sat right in the page code. The conclusion of that review: there is no open data, there is publicly available data.
- **The Federal State Information System for Spatial Planning (FGIS TP)** of the Ministry of Economic Development. Public bodies are required to upload maps of master plans and spatial planning schemes in vector form (GML) and additionally in PDF. But, as the system itself replies, only the owner of the information can get the vector model; everyone else gets raster copies. In 2024 the system held about 200,000 documents, and at least a quarter of them should contain vector geodata.
- **The St Petersburg Spatial Data Fund** in 2021 called "open" the data that is released free of charge but on request with passport details and the purpose of use. All state spatial data funds in Russia work this way: data is released on application and under a contract. Free is not yet open.

The second barrier is access from abroad. On 12 October 2026 neither the NSPD, nor FGIS TP, nor the St Petersburg fund's portal, nor the Rosreestr website opened from a German address. In 2025 most Russian installations of GeoNetwork, GeoNode and GeoServer in the registry did not respond to requests from outside the country. Such a geoportal ends up in neither international search engines nor aggregators.

How to check a geoportal

Five questions you can answer in half an hour.

1. **Which of the three parts are there?** Look at the page and link addresses: `/geonetwork/` is a GeoNetwork catalog, `/geoserver/` a GeoServer server, `/arcgis/rest/services` ArcGIS Server, `/api/v2/` next to a map is often GeoNode. If there is only a map, it is a viewer.
2. **Can you download the layer?** Look for a download button or a WFS service. A WMS image is not a download.
3. **Does the layer have a record with a licence and a data date?** Open three random records. The licence should be a link to a clear text, not "no restrictions".
4. **Does the portal serve descriptions and data to programs?** Typical addresses:

```
# GeoNetwork: catalog name and software version
curl -s -H "Accept: application/json" "https://sdi.eea.europa.eu/catalogue/srv/api/site"
# CSW catalog: how many records it has
curl -s "https://gdk.gdi-de.org/gdi-de/srv/eng/csw?
service=CSW&version=2.0.2&request=GetRecords&typeNames=csw:Record&resultType=hits"
# GeoServer: which layers can be downloaded via WFS
curl -s "https://kartta.hel.fi/ws/geoserver/avoindata/wfs?
service=WFS&version=2.0.0&request=GetCapabilities"
# ArcGIS Server: folders and services
curl -s "https://maps2.dcgis.dc.gov/dcgis/rest/services?f=json"
# ArcGIS Hub: the catalog in DCAT-US and search via OGC API Records
curl -s "https://opendata.dc.gov/api/feed/dcat-us/1.1.json"
curl -s "https://opendata.dc.gov/api/search/v1/collections/dataset/items?limit=1"
# GeoNode: number of datasets
curl -s "https://map.gov.kz/api/v2/datasets?page_size=1"
```

```
# STAC: list of collections
curl -s "https://earth-search.aws.element84.com/v1/collections"
```

5. **Is the portal visible from above?** For Europe, find its catalog among the sources of data.europa.eu or in the national geocatalog. For others, search for its services in GeoSeer. If the portal is found in neither place, its data is known only to those who know the address.

A detailed list is in the checklists, and commands for other platforms are in the chapter on APIs and bulk download.

When you do not need your own geoportal

For a small municipality with a dozen layers, a server of its own with GeoNode or GeoServer is more often harmful: there is nobody to update it, and it closes together with the project that paid for it. It is better to pass the layers to the regional or national infrastructure, subscribe to the service the neighbours already use and write export and a licence into the contract, or publish the layers as GeoPackage and GeoJSON files on an open data portal with a description on the dataset page.

A research project whose spatial data is a research result needs a research repository, not a geoportal: a file with a DOI will outlive the grant, a server will not. For imagery and other large files, a static STAC catalog in cloud storage without a server is enough. The decision for a city is worked through in the municipality example, and ways of publishing without a portal in the chapters on static sites and on cloud storage.

How it differs from its neighbours

Neighbour	Main difference
Open data portal	the main product is a dataset as a file. If the main thing on the site is a map and a layer catalog, it is a geoportal, even when tables sit next to it. Hybrids are ArcGIS Hub sites and data.europa.eu, where layers and tables sit side by side
Research repository	stores an immutable version of research data with a DOI. A geoportal shows the current state of a territory and updates it
Metadata catalog	GeoNetwork without data of its own is close to one, but its records lead to live services rather than only describing them
Search engine and aggregator	GeoSeer and similar services collect the addresses of other people's geoservices and keep no records of their own
Map viewer	shows layers, but without a catalog and download it is not a catalog

A summary table of all types is in the chapter Comparing Types and Hybrids.

Sources

- dataportals-registry export, release 1.27.0 of 15 October 2026: owners, countries, geoportal platforms with the share having an API and the share inactive, software classes, interfaces, platforms by country, countries with a geoportal and no open data portal, platforms of Russian geoportals. The queries are in the appendix.

- STAC module for GeoServer: a community module, not the core; STAC Index
- I. Begtin, The spread of STAC services for geodata (in Russian), April 2026: STAC as a protocol and difficulties with the unit of publication
- Platform profiles
- Geo-System, portals on e-mapa technology (in Polish): about 2,000 gmina and powiat portals; summary of deployments in 2025
- Local vendors: iObčina (municipalities lease the system under contract), GISPLAN, GisMaster, Map2Web, Waga Machi Guide, Louhi, IntraMaps; open source: Lizmap (MPL-2.0), GeoMapFish, Masterportal, mviewer
- I. Begtin, geoportal software in the registry, Telegram (in Russian), 17 September 2026, and distribution of geoportals by country, 27 September 2026
- I. Begtin, why geodata "absorbs" open data, Telegram (in Russian), 18 April 2023; the Malawi geoportal and Africa without open data portals, 17 May 2023; open data in Kazakhstan, 11 September 2024. The number of datasets on map.gov.kz is from the `/api/v2/datasets` response, 12 October 2026.
- Documentation: harvesting records in GeoNetwork, editing INSPIRE records in GeoNetwork, GeoNode, the INSPIRE extension for GeoServer, pycsw features
- I. Begtin, GeoNetwork, GeoNode and GeoServer: roles and number of installations, Telegram (in Russian), 9 June 2025: an assessment of the platforms' roles and the unavailability of Russian installations from abroad
- Standards: OGC, OGC API — Records — Part 1: Core approved as a standard (approved on 7 April 2025, text of the standard); SEMIC, GeoDCAT-AP 3.1.0, recommendation of 16 February 2026
- INSPIRE: European Commission, proposal amending Directive 2007/2/EC, COM(2025) 985, 10 December 2025; procedure file 2025/0393(COD) at the European Parliament, Parliament's position of 6 October 2026; the EU Council's negotiating position, June 2026; data.europa.eu, INSPIRE datasets move to the European Data Portal, 22 June 2026; EU Publications Office, INSPIRE in EU Vocabularies. The redirect from inspire-geoportal.ec.europa.eu to data.europa.eu was checked on 12 October 2026.
- Checks on 12 October 2026: the number of records in Geodatenkatalog.de and in the EEA catalog via CSW, CSW output formats in the EEA catalog (GeoNetwork 4.4.9); GDI-DE datasets on data.europa.eu: search by the gdi-de catalog; DCAT-US, DCAT-AP and OGC API Records exports on opendata.dc.gov; WFS layers on kartta.hel.fi; Earth Search
- ArcGIS search and ArcGIS Hub; I. Begtin, hidden data on ArcGIS sites, Telegram (in Russian), 22 July 2024, ArcGIS search as a catalog, 31 July 2024, and where to look for geodata, 7 June 2025: size estimates
- I. Begtin, the Arkhangelsk region geoportal after its ArcGIS licence expired, Telegram (in Russian), 21 August 2024
- Rosreestr, the public cadastral map on the NSPD platform, CNews (in Russian), 27 December 2024; I. Begtin, Rosreestr's spatial data portal, Telegram (in Russian), 7 January 2024
- Order of the Russian Ministry of Economic Development No. 349 of 17 June 2021 (in Russian) on FGIS TP formats (maps in GML and PDF); questions and answers on the FGIS TP website (in Russian) on releasing the vector model, Web Archive snapshot of 26 May 2020; I. Begtin, FGIS

TP: only rasters can be downloaded, Telegram (in Russian), 8 April 2024

- Russian Government Resolution No. 262 of 4 March 2017 (in Russian) on providing data from spatial data funds; I. Begtin, when open data is not open, Telegram (in Russian), 25 October 2021
- Availability of NSPD, FGIS TP, the St Petersburg Spatial Data Fund and the Rosreestr website: checked from a German IP address on 12 October 2026.

Research Data Repositories

A research repository accepts research data for preservation, fixes its version, issues a persistent identifier and provides a ready-made citation for a paper. There is one goal: that years later the link in the paper opens the same version of the same files.

What distinguishes it from an open data portal is not the subject but the commitments. Four things together make a catalog a repository:

- **deposit**: the author hands the files over to the repository rather than leaving a link to their own website;
- **an immutable version**: what has been published is not silently edited; a correction comes out as a new version;
- **a persistent identifier**: usually a DOI, which will survive a change of address and platform;
- **a description for citation**: authors, year, title, publisher and version, from which a citation line is assembled.

If something is missing, it is file storage or a showcase, even if a section of the site is called a repository.

Repositories are expected to make data comply with the FAIR principles (2016): findable, accessible, interoperable and reusable. FAIR does not mean "open": a dataset with closed files can also be FAIR if its description is open, it has a DOI and it is clear how to request access.

According to the dataportals-registry export (release 1.27.0 of 15 October 2026), there are 8,404 research repositories, 19% of all catalogs, the second most numerous type after geoportals. 8,285 are active. 119 are inactive or decommissioned, about 1.4%: research repositories close less often than open data portals, which stand at about 4%. Restricted access to at least part of the data is marked for 527 repositories, 6%.

On 12 October 2026 the international registry re3data described 3,535 repositories. The difference from the catalog registry is not an error: re3data describes in detail the repositories that have passed its editorial review, while the catalog registry also counts university publication repositories with little data, and data servers such as THREDDS and ERDDAP.

Why researchers publish data

A researcher most often puts data into a repository because a funder or a journal requires it.

- **Funders.** In the EU's Horizon Europe programme a data management plan is mandatory, and data is opened according to the rule "as open as possible, as closed as necessary". Since 25 January 2023 the US National Institutes of Health (NIH) have required a data management and sharing plan in every application that generates scientific data. A 2022 memorandum of the US Office of Science and Technology Policy required federal agencies to introduce policies by the end of 2025 under which publications and data from federally funded research are made available without delay. Most such policies are already in force, but since February 2026 the administration has been reviewing the memorandum itself.

- **Journals.** Many journals require a data availability statement in the paper: where the data is and on what terms. PLOS introduced this requirement in 2014.
- **Recognition.** A dataset with a DOI can be cited, and its citations are counted. A data paper turns work on data into a scholarly publication; see the section Data papers.

A statement that "data is available from the authors on request" does not replace a repository. In a 2022 study, the authors of 1,792 papers that promised data on request were asked to send it. 93% did not reply or refused; 123 authors, 6.8%, sent the data.

For a catalog this means that DOIs, long-term preservation, a licence and a description for citation are not optional features here. Without them a funder will not accept the repository in a data management plan. NIH, for example, advises first looking for a domain repository for your discipline, and if there is none, a generalist or institutional one. NIH's list of requirements for a repository begins with the same things: a persistent identifier and a dataset page that opens even if the dataset itself has been withdrawn. A repository is checked against re3data and the CoreTrustSeal certification, covered in the section Standards.

Who builds them

Research repositories are almost entirely run by universities and research organisations.

Owner	Repositories	Share
Universities and research organisations	7,164	85%
Central government	592	7%
International organisations	217	3%
Civil society organisations	137	2%
Companies	117	1%
Regions	79	1%
Other: communities, local and federal government	98	1%

Central government here means agency research services: geological, meteorological, oceanographic. International ones are consortia such as GBIF and intergovernmental organisations.

By who can deposit data, repositories fall into five kinds.

Kind	Who runs it and for whom	What it gives the author	Examples
Institutional	a university or institute for its own staff	library moderation, its own DOI prefix, often a shared repository with papers and theses	DaRUS (Stuttgart), the SPbU archive
Domain	a research community for one discipline	strict acceptance according to the discipline's standards, readers from the field	PANGAEA, GBIF, ICPSR
Generalist	a service for any discipline and any author	quick publication with a DOI without an installation of your own	Zenodo, Figshare, Harvard Dataverse, Dryad, Mendeley Data, OSF

Kind	Who runs it and for whom	What it gives the author	Examples
National or shared	the state or a consortium for all organisations in a country	a ready-made repository without an administrator of your own	Fairdata (Finland), JAIRO Cloud (Japan), DABAR (Croatia)
CRIS	the university as a research information system	a dataset as one more output next to papers and grants	portals on Pure, Research.fi

National models are covered in the section National models, CRIS in the section CRIS: hidden data catalogs.

The number of repositories is not a measure of research data

By number of research repositories in the registry the leaders are the United States (2,018), Germany (764), the United Kingdom (503), France (289), Japan (275), Canada (245), China (244) and Italy (211). Another 681 repositories are international. As with open data portals, the count shows not the volume of data but how publishing is organised in a country. In Japan 160 repositories out of 275 are showcases of the shared hosted service JAIRO Cloud on the WEKO3 software. In Croatia 76 out of 84 are organisations' repositories in the national DABAR system. In Germany 121 are university library publication servers on OPUS, in the US 146 are university repositories on the hosted Digital Commons service. Most of them hold papers and theses, and data is a small share.

What they publish

They publish tables and files behind papers, observations, code, collection descriptions. A good dataset package is more than files: it includes a description of the methodology, a data dictionary, processing code and a licence. If the repository is a shared one, the publications themselves sit next to the data. When harvesting descriptions, papers have to be separated from data, otherwise a dataset search engine will show theses.

How data is published depends heavily on the discipline. The life sciences, especially bioinformatics, have long lived in their own connected infrastructure: their own standards, identifiers and databases, where the unit of record is a study or a genome rather than a dataset. The Earth sciences (climate, meteorology, the ocean) also keep apart, with their own formats and servers. Machine learning data mostly sits on Kaggle and Hugging Face; see the chapter on ML catalogs. Economics, the social sciences and the humanities more often put data where they put papers: in the university repository or a generalist one.

The registry shows this through themes: after the general theme "science and technology", the most common are environment (about 2,300 repositories), living organisms (about 1,600), health, oceans, Earth sciences and climate. The natural sciences are described and harvested better than the social sciences and humanities: they have their own software and networks, covered in the section Disciplinary infrastructures.

Standards

Description. The basis is the DataCite schema: the latest version, 4.7, came out in March 2026. There are six mandatory fields: identifier, creators, title, publisher, publication year and resource type. Via the OAI-PMH protocol repositories also serve a short Dublin Core. Disciplines have their own schemas: DDI for social surveys, Darwin Core and EML for biodiversity, ISO 19115 for the Earth sciences, the CF conventions for NetCDF files. Platforms lag behind the schema: according to its documentation, InvenioRDM relies on DataCite version 4.3. If you need new fields, such as a link to a project via RAiD, check which version of the schema the platform supports.

Identifiers. A dataset gets a DOI, an author an ORCID, an organisation a ROR ID. Through these links search engines and aggregators connect a dataset with people, organisations and papers. In the description of a dataset on Harvard Dataverse, the schema.org markup gives the author a link to ORCID and the organisation a link to ROR. DOIs, versions and a citation line are mandatory here. How they work is covered in the chapters on identifiers and on citation.

Passing on descriptions. The main interface of research repositories is OAI-PMH: in the registry it is marked for 2,068 repositories, about a quarter. About as many serve a sitemap, and about 200 a DCAT export. Google Dataset Search needs schema.org markup on the dataset page. Newer is FAIR Signposting: the dataset page lists links to the description, the files, the licence and the authors' identifiers in a response header or a separate file. This way a program learns the essentials about a dataset without parsing HTML. Signposting is in the core of Dataverse since version 5.14, InvenioRDM since version 12 and DSpace since version 7.6. More on export formats in the chapter on metadata export.

Trust. CoreTrustSeal is a certification for long-term preservation against 16 requirements: mission, rights, continuity of access, preservation, security. About 160 repositories are certified, and the requirements have been updated for 2026–2028. In 2026 the CoreTrustSeal board approved a separate methodology for how a repository should prepare for its own closure. The certification and a profile in re3data are the first things funders and journals look at when checking a repository.

Platforms

About a third of research repositories are home-grown, and they have the worst programmatic access. Among the ready-made platforms many were not built for data.

Platform	In the registry	What it is	API marked
Custom build	2,836	—	5%
DSpace	829	open-source platform for library repositories: publications and data	99%
Pure	369	Elsevier's commercial CRIS	100%
GBIF IPT	360	open-source tool for publishing biodiversity data	91%
Figshare	288	hosted service: the shared figshare.com and organisational repositories	99%
Dataverse	257	open-source data repository	99%
THREDDS	249	server for NetCDF and HDF arrays	98%
ERDDAP	175	server for observation series and grids	99%

Platform	In the registry	What it is	API marked
Digital Commons	174	Elsevier's hosted service for university repositories, mostly in the US	99%
WEKO3	161	software of the Japanese JAIRO Cloud service	100%
EPrints	129	open-source platform for publication repositories	99%
OPUS	124	open-source publication server of German libraries	98%

Less common are InstDB (83, China), DABAR (76, Croatia), InvenioRDM (74, including Zenodo), DSpace-CRIS (44) and CKAN (39).

The table shows three things. First, the platforms fall into four families: publication repositories (DSpace, EPrints, OPUS, Digital Commons, WEKO3), CRIS (Pure), data repositories (Dataverse, InvenioRDM, Figshare) and data servers (THREDDS, ERDDAP, IPT). Only the third family was designed for datasets, so repositories on platforms of the first family hold little data. Second, ready-made platforms almost always have an API, while for custom builds it is marked in only 5%. The argument against a home-grown repository is even stronger here than for open data portals. Third, the share of inactive installations is highest for IPT, 12%; for the other large platforms it is no more than 2%.

What is in the core and what is not

DOIs are in the core of Dataverse, InvenioRDM, Figshare and DSpace, and a ready-made citation with BibTeX and RIS export in the first three. DSpace up to and including version 10.1 has only meta tags for Google Scholar; the "Cite" button is being ported from DSpace-CRIS. Installing an open data portal for research and bolting on DOIs with an extension is a workaround.

Capability	Dataverse	InvenioRDM	DSpace	CKAN
DOI	core: DataCite, EZID, also Handle	core: DataCite	core, enabled by configuration; Handle by default	the ckanext-doi extension, declared for CKAN 2.9 and 2.10
Dataset versions	core	core: each version has its own DOI plus a common one for all	core	no
Embargo and restricted files	core, with access requests	core, with access requests	core: embargo until a date, request a copy of a file	only an entire private dataset
OAI-PMH	core	core	core	not in the core
schema.org markup	core	core	not found in the core	the ckanext-dcat extension
DCAT	no	core (DCAT-AP)	no	the ckanext-dcat extension
FAIR Signposting	core since 5.14	core since v12	core since 7.6	no
Citation line, BibTeX, RIS	core	core (CSL styles, BibTeX, CSL-JSON; no RIS)	no: only meta tags for Google Scholar	the ckanext-doi extension (only with a DOI) or ckanext-citeproc
Domain-specific description fields	core: ready-made blocks and your own	core configuration	core configuration	the ckanext-scheming extension

Extensions work differently in research platforms than in CKAN: the essentials are in the core, and outside are file previewers and additional export formats. In Dataverse these are external tools and external exporters, maintained by the GDCC community, for example the RO-Crate and DCAT 3 exporters; the Croissant exporter moved into the core in version 6.10. The Dataverse documentation itself calls the exporter mechanism experimental and warns that it may need rework when the platform is upgraded. DSpace has entity models and configuration instead of plugins: DSpace 10 (2026) introduced a separate "dataset" item type with its own page and a detailed file list. CKAN remains a workaround for research: the documentation of the DOI extension does not confirm support for the current core line; details are in the chapter on CKAN.

Versions and maintenance. As of 12 October 2026 the latest releases are Dataverse 6.12.1 (8 October), InvenioRDM v14 (7 August; v13 is supported until 7 February 2027, v15 is in beta), DSpace 10.1 (14 September; fixes 9.4 and 8.5 came out the same day). Large installations lag one or two versions behind: Harvard Dataverse runs 6.10.1, DSpace@MIT 8.2. This is fine if upgrades are planned. How much maintenance costs is covered in the chapter on deployment, and platform profiles are in the part Platforms and Extensions.

How much you can put in a generalist repository

Generalist repositories are free up to a limit, and big data costs money.

Service	Free	Beyond the limit
Zenodo	up to 50 GB and 100 files per record, unlimited number of records	a one-off increase to 200 GB per record on request. The rules prohibit splitting one dataset into records to get around the limit
Harvard Dataverse	files up to 2.5 GB via the browser, up to 1 TB per researcher	larger by arrangement, primarily for Harvard researchers
Figshare	20 GB of personal space, files up to 20 GB	Figshare+: a one-off fee of 1,225 dollars per 250 GB, files up to 5 TB
Dryad	authors from partner organisations up to 10 GB, paid by the partner	others pay 150 dollars per dataset up to 5 GB and more on a scale, up to 12,162 dollars for 2 TB

Limits and prices are as of 12 October 2026; Dryad's fees have applied since 6 May 2025. If you have terabytes of data, a generalist repository does not fit: you need the discipline's domain archive or your own storage with a description in a catalog. The options are in the chapters on data storage and on cloud storage and hubs.

Disciplinary infrastructures

Many sciences have their own catalogs on their own software. These are full catalogs, with descriptions, licences and programmatic access. But few people outside the discipline know about them, and general dataset search engines hardly harvest them.

- **ERDDAP** is a data server from the US National Oceanic and Atmospheric Administration (NOAA) for ocean, weather and climate observations. It serves series and grids through an API in CSV, JSON, NetCDF and other formats. According to the project website, about 100 organisations in at least 17 countries have installed ERDDAP; the registry lists about 175 servers.

- **THREDDS Data Server** from Unidata and **OPeNDAP Hyrax** servers distribute large arrays in NetCDF and HDF. The registry lists about 250 THREDDS servers and about 100 OPeNDAP servers. Their descriptions are sparse: from the outside such a server looks more like scientific file storage than a catalog.
- **GBIF IPT** is a tool for publishing biodiversity data. Universities and museums install it, and the global GBIF network harvests data from all installations. In October 2026 GBIF had 392 registered IPT installations, and the network itself about 125,000 datasets and almost 4 billion species occurrence records. Each installation serves descriptions in DCAT. In Russia, for example, the IPT of the Zoological Institute of the Russian Academy of Sciences (ipt.zin.ru) works this way.
- **WIS 2.0** is the World Meteorological Organization's system for exchanging weather station data. The open wis2box package serves data according to the OGC API standard. The registry lists about 95 installations, including Russian (wis2box.mecom.ru) and Kazakh ones. Its predecessor is OpenWIS, built on GeoNetwork. Its code has not been developed since 2022, and 11 such portals remain in the registry.

The conclusion for a data owner in such a discipline: install its standard tool, not a general portal. That way the data reaches the network where colleagues look for it. For it to be found outside the discipline too, you also need a metadata export in DCAT or schema.org; see the chapter on discoverability.

National models

Countries bring research data together into one system in different ways.

- **Finland.** The Ministry of Education and Culture offers the Fairdata set of services, built by the CSC centre: IDA storage, Etsin dataset search, the Qvain description editor, the Metax metadata interface, long-term preservation and the AVAA publishing platform. A researcher goes from storage to publication within one system.
- **Japan.** Since 2012 the National Institute of Informatics (NII) has provided universities with the hosted repository JAIRO Cloud on its WEKO3 software. Since 2016 the service has been run together with the Japan Consortium for Open Access Repositories (JPCOAR); since 2017 it has been paid, with 100 GB included free. At the end of fiscal year 2025, 822 organisations used it. An organisation cannot modify the software, but NII takes care of upgrades, servers and metadata export. The model grew out of a shortage of people: according to NII, in almost 90% of consortium members less than one full-time staff member works on the repository. In 2026 access requests for restricted files are being added to the service.
- **Croatia.** The national DABAR system has existed since 2015 and is run by SRCE, the computing centre of the University of Zagreb. It has 185 repositories of research organisations and universities. In November 2025 the system was moved to a new architecture based on Fedora Commons and connected to the national CRIS, CroRIS. According to a 2025 study, though, only just over half of the objects were open.
- **Germany.** There is no single platform, but there is a set of open-source products. Dataverse runs DaRUS in Stuttgart and heiDATA in Heidelberg, with 25 German installations in the registry. Invenio runs the RODARE archive. DSpace runs OpARA in Dresden. The local MyCoRe

has 22 German portals in the registry, for example opendata.uni-kiel.de. Alongside are large domain archives such as PANGAEA and the agency da|ra, which issues DOIs for social and economic data and provides search over them.

- **China.** The Science Data Bank (ScienceDB, scidb.cn) is a shared repository run by the Computer Network Information Center of the Chinese Academy of Sciences. Institutes publish data on the standard InstDB software from the FairStack suite, with 82 Chinese installations in the registry. InstDB supports three access modes: open, after login and on request. On top of it runs the FinData search engine (findata.cn). From outside the system is almost invisible: in December 2025, of 72 known InstDB installations, between 6 and 12 opened from foreign addresses.
- **United States.** Data is collected by agency aggregators. The US Geological Survey (USGS) Science Data Catalog combines several repositories, the main one being ScienceBase. According to a 2022 observation, a DOI, a geographic location and the observation period are mandatory there for every dataset. The Department of Energy runs OSTI.GOV and the DOE Data Explorer, and Science.gov searches the research outputs of many federal agencies, including data.
- **France.** The national repository Recherche Data Gouv runs on Dataverse.
- **Russia.** There is no national research data infrastructure. According to a July 2025 count, there were 6,806 datasets with a DataCite DOI and a Russian affiliation, most of them from the Russian Academy of Sciences, and almost all were uploaded to foreign platforms such as Zenodo, Dryad and Mendeley Data. The registry lists 75 Russian research repositories, more than half of them home-grown. Where science is embedded in global networks, data is published on their tools, as in GBIF IPT nodes.

The Japanese and Croatian models are the equivalent of the standard portal in open data: one build for hundreds of organisations that have no people of their own. This works when the shared system issues DOIs itself and passes descriptions upwards, leaving the organisation only to fill it. The weak spot is the same as with the standard portal: such systems are filled mainly with papers and theses, and data remains a small share.

CRIS: hidden data catalogs

A CRIS (current research information system) records an organisation's research activity: publications, projects, grants, staff, equipment. Many universities publish data not in a separate repository but in their CRIS, as one more research output. The most widespread system is Elsevier Pure, with about 370 portals in the registry. Next to it are DSpace-CRIS (about 45) and Converis. In Pure a dataset is a separate content type. It is disabled by default and enabled by an administrator, and DOIs can be minted directly from the system via DataCite. A national example is the Finnish Research.fi: the country's publications, projects, funding, researchers and datasets in one search.

What makes these catalogs hidden is that the portal looks like a university website about research rather than a data catalog, and datasets get lost among papers and profiles. Commercial platforms also shut themselves off from programs: on 12 October 2026 four university portals on Pure responded to an automated request with a refusal or a bot check. For data in a CRIS to be found, a dataset must have a DataCite DOI: then its description lives outside the university website as well.

Why they are hard to harvest

There is a lot of research data, but bringing it into one search is harder than with open data portals or statistics. The reasons are technical and commercial.

1. **The OAI-PMH protocol is not about files.** It is the main interface of research repositories. Its mandatory format is simple Dublin Core, which has no place for a list of files: a record usually contains only a link to the landing page. Selection by sections (sets) is defined by the repository itself, and the protocol does not allow asking it for "only data".
2. **Data is scarce among papers.** In DSpace a filter by item type is described in the configuration but not enabled in search by default. To find datasets, a harvester downloads the descriptions of all records and selects the data itself. In December 2024 the SPbU open access archive had 17 records marked as datasets out of 47,619. In October 2026 the archive runs on DSpace 10.1, and there is still no item type filter among the search filters. According to an estimate from the same year, datasets account for less than 5% of records in DSpace repositories.
3. **Commercial platforms.** Figshare and portals on Pure made up about a third of the research catalogs in the Dateno registry in 2024. They restrict automated harvesting.
4. **The files are somewhere else.** Increasingly a researcher puts the files in Zenodo or Mendeley Data, and the university repository keeps a record with a link to someone else's record.
5. **Aggregators count everything as data.** In October 2026 DataCite had more than 76 million records of type "dataset", including slices of large databases and objects far removed from tables. More in the chapter The World of Catalogs in Numbers.

What follows for a repository owner: enable a filter by item type or a separate collection for data; if the platform can, serve not only Dublin Core but also the DataCite schema via OAI-PMH; store files yourself or provide direct links to them; enable FAIR Signposting; create a profile in re3data.

Data papers

A data paper is a peer-reviewed article that describes a dataset: how it was collected, what it contains, how to use it. It has no hypotheses or conclusions. The data itself sits in a repository; the paper cites it and gives the authors scholarly credit for working with the data. Such papers are published by Scientific Data, Earth System Science Data, Biodiversity Data Journal, Data in Brief and other journals. GBIF keeps a list of them, and IPT can generate a draft paper from a dataset description. There are still few of them: in October 2026 DataCite had about 18,500 records of type "data paper".

For a catalog the conclusion is simple: if a dataset has a paper, link them by giving the paper's DOI in the dataset description, and vice versa. For a dataset search engine the question is open. A data paper is not the data itself, but sometimes it is the only place where the dataset is mentioned at all.

Examples

Generalist

- Zenodo: a generalist repository at CERN on InvenioRDM. It accepts data, code and publications in any discipline, with a DOI for each version.
- Harvard Dataverse: the Dataverse installation at Harvard University, open free of charge to researchers worldwide.
- Figshare: a hosted service, a free generalist repository and paid university repositories.
- Dryad: a non-profit repository where a curator checks each dataset; partner organisations or authors pay.

Domain

- PANGAEA: Earth and environmental sciences.
- GBIF: the global biodiversity data network, which harvests datasets from hundreds of IPT installations.
- ICPSR: the social science data archive at the University of Michigan.

National and shared

- Etsin: search for Finnish research datasets in the Fairdata system.
- JAIRO Cloud: the shared hosted repository of Japanese universities.
- DABAR: Croatia's national repository system.
- Science Data Bank: the shared repository of the Chinese Academy of Sciences.
- Recherche Data Gouv: France's national repository on Dataverse.

Institutional

- CaltechDATA: the California Institute of Technology repository on InvenioRDM, about 77,000 records in October 2026.
- DaRUS: the University of Stuttgart data repository on Dataverse.

Where it breaks

A publications repository posing as a data repository. The catalog is called a data repository, but there is almost no data in it: the SPbU archive has 17 datasets per 47,000 records. A harvester cannot ask for "only data" and takes thousands of papers. Why this happens is covered in the section Why they are hard to harvest.

A DOI, but no file. The DOI leads to a landing page, but the file has been deleted, sits on someone else's site or is available "on request". The NIH requirement is precise here: the dataset page must open even if the dataset has been withdrawn, and explain what happened to it. An empty error page breaks the link in the paper.

Published material gets edited. A file was replaced without a new version, and whoever cited the dataset now cites different data. Research platforms prevent this: Zenodo allows changing the files of a published record only for 30 days; after that, a new version or a justified request to support. On an open data portal a file is usually replaced silently, and for research that is a breakage.

The repository closed. According to a 2023 study, of 3,069 repositories in re3data, 191, or 6.2%, had closed. The median age at closure was 12 years. Only 44% of the closed ones transferred their data to another repository, and in almost half of the cases the data may have been lost. Hence CoreTrustSeal's requirements for a closure plan and the conclusions in the chapter on archiving.

The robot cannot get through. On 12 October 2026 a dataset page on Harvard Dataverse responded to an automated request not with the page itself but with a bot check from the AWS cloud firewall, although the API and OAI-PMH responded normally. Portals on Pure responded with a refusal the same day. For a search engine or harvester that reads the dataset page, the schema.org markup and the Signposting header, such a dataset does not exist. Bot protection is necessary, but the machine entrances (the dataset page, the API, OAI-PMH) must stay open.

A home-grown repository without an API. Only 5% of home-grown research repositories have a programming interface marked. Neither research aggregators nor dataset search engines will harvest such a repository.

Reliance on one person. The repository lives as long as a particular staff member or PhD student runs it. Among IPT installations the share of inactive ones is 12%, several times higher than for other large platforms. Shared services such as JAIRO Cloud and DABAR appeared precisely as the answer to this.

How to check a research repository

Five questions you can answer in half an hour.

1. **Does the dataset have a DOI, and where does it lead?** A query to DataCite shows the landing page address, the resource type and the version.
2. **Does the repository serve descriptions to harvesters?** The OAI-PMH protocol should respond, and among the formats, besides simple Dublin Core, there should be the DataCite schema (`oai_datacite`).
3. **Can you select only data?** Look at the OAI-PMH sets and the search filters: is there a separate collection or a "dataset" item type?
4. **Can programs see the dataset?** Check the schema.org markup and the `Link` header with a link to a `linkset`; that is FAIR Signposting. If a bot check comes back instead of the page, search engines will not read the dataset.
5. **Is the repository listed in re3data, and is it certified?** This is what a funder will check.

```
# DataCite: where the DOI leads, resource type and version
curl -s "https://api.datacite.org/dois/10.7910/DVN/TJCLKP"
# OAI-PMH: is the interface alive, which schemas it serves and which sets it has
curl -s "https://dataverse.harvard.edu/oai?verb=Identify"
curl -s "https://dataverse.harvard.edu/oai?verb=ListMetadataFormats"
curl -s "https://dataverse.harvard.edu/oai?verb=ListSets"
# Dataverse: platform version and the dataset description in schema.org
curl -s "https://dataverse.harvard.edu/api/info/version"
curl -s "https://dataverse.harvard.edu/api/datasets/export?
exporter=schema.org&persistentId=doi:10.7910/DVN/TJCLKP"
# InvenioRDM: number of records and the FAIR Signposting header on a record page
curl -s "https://data.caltech.edu/api/records?size=1"
curl -sI "https://data.caltech.edu/records/0jxb3-ny611" | grep -i '^link'
# DSpace 7 and later: platform version
```

```
curl -s "https://dspace.mit.edu/server/api"
# re3data: is the repository in the registry
curl -s "https://www.re3data.org/api/beta/repositories?query=PANGAEA"
```

All commands were checked on 12 October 2026. Besides Dublin Core, Harvard Dataverse serves the DataCite and DDI schemas via OAI-PMH. A CaltechDATA record page returns a header with a link to a `linkset`. A detailed list of checks is in the checklists, and commands for identifiers are in the chapter on identifiers.

When you do not need your own repository

A repository of your own is ongoing work: upgrades, moderation, a contract with DataCite, answers to authors. For a small group Zenodo or a domain repository of its discipline is often enough, and funders accept this. For a university without library staff it is better to join the country's or a consortium's shared service, as Japanese and Croatian universities do. Your own Dataverse or InvenioRDM is justified when the library moderates dozens of groups, you need your own DOI prefix and there is someone to maintain the system. Such a decision is worked through in the university example. Requirements for a research organisation are in the chapter on scenarios, and for a consortium in the neighbouring section.

How it differs from its neighbours

Neighbour	Main difference
Open data portal	regular updates matter more than an immutable version with a DOI; a public body publishes, not the author of a study
Microdata catalog	survey data about people, access on request and descriptions in DDI. Social science archives such as ICPSR sit at the junction of the two types
ML catalog	the dataset is needed to load into training code, not to cite in a paper; next to it are a dataset card and splits
Geoportal	the main product is a layer and a service, not a fixed file. THREDDS and ERDDAP servers sit at the junction
Publications repository	the unit is a paper or a thesis, and data is a small share of records
CRIS	records all of an organisation's research activity, and a dataset is just one of the record types

A summary table of all types is in the chapter Comparing Types and Hybrids.

Sources

- dataportals-registry export, release 1.27.0 of 15 October 2026: owners, countries, platforms with the share having an API and the share inactive, main platforms by country, interfaces, themes and access of research repositories. The queries are in the appendix.
- re3data and its API: number of repositories, 12 October 2026
- M. D. Wilkinson et al., The FAIR Guiding Principles for scientific data management and stewardship, Scientific Data, 2016; the FAIR principles at GO FAIR

- Funder and journal policies: open science in Horizon Europe; the NIH Data Management and Sharing Policy, in force since 25 January 2023, including the requirements for repositories; selecting a repository according to NIH; US federal agency policies under the 2022 OSTP memorandum, SPARC, updated 4 May 2026; on the review of the memorandum, STM, 19 February 2026; PLOS data availability requirements
- M. Gabelica, R. Bojčić, L. Puljak, Many researchers were not compliant with their published data sharing statement, Journal of Clinical Epidemiology, 2022
- DataCite Metadata Schema, version 4.7 of 3 March 2026; InvenioRDM metadata reference: the supported schema version
- FAIR Signposting and who supports it; Signposting in Dataverse, InvenioRDM and DSpace
- CoreTrustSeal: the 2026–2028 requirements, the number of certified repositories and the methodology for closure, 12 October 2026
- Platforms: Dataverse export formats and external exporters; the GDCC exporter catalog; InvenioRDM export formats in the source code; InvenioRDM maintenance policy; DOIs in DSpace; the "dataset" item type in DSpace 10; Dataverse features; releases of Dataverse and DSpace. Versions of Harvard Dataverse and DSpace@MIT, and OAI-PMH, Signposting and bot check responses: checked 12 October 2026.
- Limits and prices: Zenodo and editing files after publication; Harvard Dataverse; Figshare and Figshare+; Dryad; checked 12 October 2026
- OAI-PMH 2.0 specification: the mandatory Dublin Core format and sets; DSpace default search configuration: the item type filter is described but not enabled; the dataset content type in Pure; checked 12 October 2026
- ERDDAP: who uses it; THREDDS Data Server; OPeNDAP; GBIF IPT and the GBIF installation registry API; wis2box; OpenWIS; checked 12 October 2026
- National models: Fairdata, Research.fi; JAIRO Cloud at JPCOAR: history, terms and 822 organisations at the end of fiscal year 2025; JAIRO Cloud at NII; NII's presentation for COAR, 2026, on staff workload; access requests in JAIRO Cloud, 23 March 2026; DABAR and on the new version of DABAR, 15 May 2026; National Repository Infrastructure and Open Access Challenges: The Croatian Perspective, LIBER Quarterly, 2025; DaRUS, heiDATA, RODARE, OpARA, MyCoRe, da|ra, ScienceDB, FinData, USGS Science Data Catalog, DOE Data Explorer, Science.gov, Recherche Data Gouv; checked 12 October 2026
- D. Strecker, H. Pampel, R. Schabinger, N. L. Weisweiler, Disappearing repositories: Taking an infrastructure perspective on the long-term availability of research data, Quantitative Science Studies, 2023
- Data papers at GBIF; the DataCite API: number of records by type, 12 October 2026
- I. Begtin, Telegram (in Russian): Germany, 4 April 2023; Fairdata, 30 July 2023; data papers, 7 April 2024; InstDB in China, 7 April 2024; data depends on the discipline, 12 September 2024; why research data is hard to index, 16 November 2024; ERDDAP, 9 December 2024; OpenWIS, 23 December 2024; the SPbU DSpace, 25 December 2024; IPT in Russia, 26 December 2024; research catalogs in the UK, 16 June 2025; DataCite and datasets by country, 20 July 2025; availability of InstDB from abroad, 25 December 2025; the USGS data catalog, 17 February 2022

Indicator Catalogs

An indicator catalog serves statistical series and tables with breakdowns: time, territory, sex, industry. A person does not download "the 2019 file" but chooses a breakdown. A program must be able to do the same with a request.

This is how official statistics, macroeconomics, central bank data, health statistics and Sustainable Development Goal indicators are published. According to the dataportals-registry export (release 1.27.0 of 15 October 2026), there are about 3,400 such catalogs, 8% of all catalogs. Almost all are active: fewer than 1% are inactive.

The main conclusion of the chapter: an indicator catalog is useful when the data in it is primary and the website with charts is only one of the showcases. The whole database can be downloaded, each series has a persistent address, and reference lists are shared across all tables. A table builder in the browser without an API and export keeps the data on the website, even if the website looks modern.

How an indicator is structured

An indicator catalog has a different unit than an open data portal. There the main thing is a file. Here it is a number with coordinates.

- **Indicator**: what is measured: population, an exchange rate, the unemployment rate.
- **Breakdown** (a dimension in the standards): what the data is divided by: year, region, sex, age, industry.
- **Code list**: the list of permitted values of a breakdown with codes: region codes, currency codes, codes of economic activities.
- **Value** (observation): one number at the intersection of breakdowns: the population of France on 1 January of a given year.
- **Time series**: values of one indicator for the same breakdowns, arranged in time.
- **Attributes**: notes on a value: unit of measure, "provisional data", "break in series due to a change of methodology".
- **Methodology** (reference metadata, an indicator passport): how the indicator is calculated, where the source data comes from, when the methodology changed.

All combinations of breakdowns together form a table that the standards call a cube. An indicator catalog is a search over cubes, a way to cut out the needed part and serve it to a person or a program.

What this looks like in practice can be seen with the US dollar to euro exchange rate on the European Central Bank's data portal. The series has the key `EXR.D.USD.EUR.SP00.A`, and each part of it is the value of one breakdown:

Key part	Breakdown	Value
`EXR`	dataset	exchange rates

Key part	Breakdown	Value
`D`	frequency	daily
`USD`	currency	US dollar
`EUR`	against which currency	euro
`SP00`	type of rate	ECB reference rate
`A`	series variation	standard value

The same key also works as an address. On 12 October 2026 a request to the ECB API returns the latest value (1.1206 for 9 October) together with attributes: observation status, number of decimals, unit, full series title. Replace `USD` with `JPY` and you get the yen rate, without searching for a new file. All good indicator catalogs work this way: the address of a series is assembled from code list codes.

What gets its own page

Three levels get their own page or address in an indicator catalog:

1. **A theme or database**: "Population", "Prices", "Balance of payments".
2. **A table** (a dataflow in SDMX): a cube with a fixed set of breakdowns. This is what is usually called a dataset.
3. **A series**: one line on a chart that can be cited.

That is why the number of "datasets" in indicator catalogs cannot be compared with open data portals. On 12 October 2026 Statistics Sweden had 2,814 tables, Statistics Norway 3,753, Statistics Canada 8,270 tables available via its API, while the ECB had only 108 datasets, but with millions of series inside them. One table replaces hundreds of files "by year and region" on an ordinary portal.

Who builds them

Almost half of indicator catalogs are run by central government: statistical offices, central banks, ministries.

Owner	Catalogs	Share
Central government	1,556	46%
Regions	662	19%
International organisations	367	11%
Local government	273	8%
Civil society organisations	250	7%
Companies	140	4%
Universities and research organisations	134	4%

The picture is the reverse of open data portals, where almost half belong to municipalities. Statistics are calculated centrally, and a city has few series of its own.

- **National statistical offices and central banks** are the foundation of the type. Next to them are ministries with their own sector statistics, above all health ministries.
- **International organisations**: Eurostat, the OECD, the World Bank, the IMF, the Bank for International Settlements, UN agencies. They not only publish their own databases but also collect country series, so they need standards-based exchange.
- **Regions**: statistical offices of the German Länder, the Spanish autonomous communities, US states. About 43% of regional showcases in the registry are custom builds.
- **Local government**: mostly Dutch and Flemish municipalities: 131 of 273 local showcases run on one platform, Swing.
- **Civil society and research projects** bring other people's series into one search. This is how Our World in Data and DBnomics from the French research centre CEPREMAP work: in October 2026 DBnomics collected series from 94 sources, from statistical offices to central banks.

By country the leaders are the United States (333), the Netherlands (186), Germany (152), France (103) and Spain (82); another 373 catalogs cover the whole world. As with open data portals, the number speaks of organisation, not of the quality of statistics. In the Netherlands 167 catalogs are Swing showcases of cities and provinces. In the US 69 are county and hospital network health indicator sites on one cloud service. In France 38 are geostatistical observatories on the Géoclip platform.

What they publish

Macroeconomics, demography, prices, the labour market, social statistics, finance, foreign trade. Separate layers are health statistics from health ministries, Sustainable Development Goal indicators and local indicators: neighbourhood population, employment, housing.

Statistics are published at one of three levels. At the first, they are documents: yearbooks and bulletins in PDF, from which a ten-year series is assembled by hand. At the second, tables: Excel on the website or a table builder that builds a table on request and lets you download it. At the third, the data is primary and the website with charts is one of the showcases: the whole database can be downloaded, each indicator has an address in the API, and reference lists are shared. An indicator catalog in the sense of this chapter begins at the second level, and it is really useful only at the third.

What an indicator catalog usually lacks: records of individual people and businesses (that is a microdata catalog), original questionnaires, territory geometry. Regional boundaries for a map have to be taken from a geoportal, and the codes in the indicator code list must match the codes of the layer, otherwise the series does not fit onto the map.

Standards

SDMX is the main standard for exchanging statistics. It is maintained by seven international organisations: the Bank for International Settlements, the ECB, Eurostat, the IMF, the OECD, the UN and the World Bank. SDMX is not one format but three parts:

- **the model**: how to describe a cube: which breakdowns a table has, which code lists are attached to them, which attributes are permitted. These descriptions are called structures;
- **formats**: SDMX-ML (XML), SDMX-JSON and SDMX-CSV;
- **the API**: rules for addresses by which a program gets data and structures: the list of tables, code lists, a slice of a series.

Live services most often run on version 2.1: the ECB, Eurostat and IMF APIs use it. Version 3.0 came out in 2021, 3.1 in May 2025; 3.1 makes it possible to add breakdowns to a table without breaking data already collected. For common subjects there are global structures: national accounts, the balance of payments, the Sustainable Development Goals. If an office transmits data according to such a structure, an international organisation takes it without manual recoding. In the registry, SDMX addresses are marked for 177 indicator catalogs, about 5%.

JSON-stat is a lightweight cube format in JSON: values, breakdowns and labels in one response. Version 2.0 is the default format of the Eurostat API and the new PxWeb API. It is simpler than SDMX for a programmer, but it has no exchange of code lists and structures between organisations.

PC-Axis (PX) is a file format for tables with a built-in description, native to PxWeb. Many open data portals serve it too: in 2023 the Spanish datos.gob.es held more than 24,000 datasets in PX, and the Irish data.gov.ie more than 7,000.

Methodology. The description of an indicator (how it is calculated, the unit of measure, breaks in series) is stored separately from the numbers but next to them. Without it a series cannot be compared across years. In the European Statistical System methodology is described according to a common structure, SIMS; the IMF requires descriptions for data that countries publish under its dissemination standards.

IMF standards. Countries that have subscribed to the IMF data dissemination standards (e-GDDS, SDDS, SDDS Plus) publish a National Summary Data Page: a standard page with the main macroeconomic series, a release calendar and often SDMX files. The registry lists 118 such pages. This is not a catalog in the full sense but a mandatory minimum that shows what a country publishes and when.

DCAT for statistics. For a statistical office's tables to appear on an open data portal, their descriptions must be provided in DCAT. There is a profile for this, StatDCAT-AP: it adds breakdowns, attributes and a link to the SDMX structure to DCAT-AP. The current version is 1.0.1 (2019). Version 3.0.0 has been aligned with DCAT-AP 3.0.1 and since 10 June 2026 has remained a working draft. In practice, the registry found a DCAT export for fewer than ten indicator catalogs. That is why statistics are poorly visible on open data portals and in dataset search engines unless the office publishes them there separately.

Linked data. The W3C has the RDF Data Cube vocabulary (2014) for statistics as linked data. It is rarely used, mostly in research projects.

Platforms

Custom-built showcases make up 44% (1,505), more than for any large type. Only 37% of indicator catalogs have programmatic access, worse than open data portals (69%), geoportals (70%) and research repositories (63%). Among custom builds an API is marked for only 8%.

Platform	Catalogs in the registry	What it is	Programmatic access
Custom build	1,505	—	for 8%
Swing	214	ABF Research's commercial indicator platform for municipalities and provinces in the Netherlands and Flanders	marked for 91%
WordPress, Drupal	134 and 131	websites with indicator pages	the CMS API serves pages and posts, not series
Knoema	125	cloud service; mostly the African Open Data for Africa showcases	service shut down, see "Where it breaks"
IMF National Summary Data Page	118	standard page under the IMF standards	most often SDMX files
PxWeb	110	Statistics Sweden's open-source platform	own API, JSON-stat, CSV, PX
DHIS2	74	the University of Oslo's open-source health statistics system	Web API in the core
OpenSDG	72	open-source static website for the Sustainable Development Goals	export of all indicators as files
Conduent HCI	68	cloud health indicator service for US counties and hospitals	no
Géoclip	53	commercial platform for geostatistical observatories, mainly in France	no
.Stat Suite	40	the SIS-CC community's open-source SDMX platform	SDMX API in the core
SDMX-RI	36	Eurostat's open-source SDMX components	SDMX API
Power BI, R Shiny, Tableau	42, 39 and 29	dashboards	almost none: 0–5%

The type divides into three worlds. The first is specialised statistical software, where an API and export are available out of the box: PxWeb, .Stat Suite, SDMX-RI, DHIS2, OpenSDG. The second is cloud services for local and sector showcases: Swing, Géoclip, HCI, Knoema; convenient, but the data lives with the vendor. The third is CMS websites, dashboards and custom portals, where the indicator is on the screen but not available to a program.

Reusable statistical software

There are few programs that many statistical offices install themselves.

- **PxWeb** was developed by Statistics Sweden; the offices of the Nordic countries and dozens of others, from Estonia to the Philippines, run it. The last release of the old line is 2025.v1 (February 2025). The next generation is two separate applications. Sweden built the PxWebApi 2 API together with Statistics Norway and launched it in October 2025 (release 2.7.0, September 2026). The new PxWeb 2 interface takes data from this API (release 2.12.0, 7 October 2026). The new API accepts queries right in the address and returns JSON-stat 2, CSV,

PX, Excel and HTML, and in beta, Parquet; Statistics Sweden has not yet enabled Parquet. An important caveat: PxWeb has no full SDMX service. The old API can return a table as an SDMX-ML 2.0 file; in the new one SDMX is not among the formats at all. For exchange with international organisations PxWeb is usually supplemented with another tool.

- **.Stat Suite** is the open-source platform of the SIS-CC community, run by the OECD. It is built around SDMX and assembled from modules: the Data Explorer (release 27.7.0, July 2026), a tool for loading and maintaining data, and an SDMX service. It powers the OECD Data Explorer, the FAO, the ILO, and the statistical offices of Australia, New Zealand, Italy and Switzerland.
- **SDMX-RI** is a set of Eurostat components under the EUPL licence. The main one is a web service that takes data from a dissemination database and serves it via SDMX. The components can be embedded into your own system separately. .Stat Suite uses exactly this service: an office that installs .Stat Suite is in fact also installing SDMX-RI.
- **Fusion Metadata Registry** is a registry of SDMX structures: code lists, table descriptions and concepts. It is maintained by the Bank for International Settlements. The application is free under the BIS public licence, the accompanying tools are open under Apache 2.0, and release 12.4.2 came out on 19 September 2026. Eurostat's SDMX registry, for example, runs on it.

Next to them are sector-specific and niche products:

- **DHIS2** is the University of Oslo's system for collecting and analysing health statistics, release 2.43, May 2026. Health ministries in dozens of countries in Africa and Asia use it to collect hospital reports and calculate indicators. The registry lists 74 installations in 54 countries, mostly in Africa and Asia. It is above all a collection system; a public showcase is an add-on, and the owner of the installation decides which data is visible without login.
- **OpenSDG** is a static website of Sustainable Development Goal indicators, release 2.4.0, August 2025. A clear showcase and a full export as files, but not an exchange channel. More in the chapter on static sites.
- **PxStat** is the platform of Ireland's statistical office: each table is also a dataset with export in JSON-stat, PX, CSV and XLSX. It has hardly been reused elsewhere.

Everything else is the offices' own development, hence the 44% custom-built showcases.

Core and add-ons

Indicator platforms have no extension ecosystem like CKAN's. Capabilities are either in the product or added with a neighbouring product.

Capability	PxWeb	.Stat Suite	OpenSDG	DHIS2
API	core	core (SDMX)	no, files instead	core
SDMX	only an SDMX-ML 2.0 file in the old API	core	optional	not checked
JSON-stat	core	not checked	not checked	not checked
Bulk download	a whole table via the API within the limit	via the API	core	via the API
Structure registry	no	core; Fusion Metadata Registry can be connected	no	no

Capability	PxWeb	.Stat Suite	OpenSDG	DHIS2
DCAT	no	not checked	not checked	not checked
Separate interface	the new PxWeb 2 interface is separate from the API	the Data Explorer is separate from the SDMX service	the site template is edited in Git	applications on top of the Web API

Two practical conclusions follow from the table and the registry figures. The registry found a DCAT export for fewer than ten indicator catalogs, whatever platform they run on. For statistics to reach an open data portal and search engines, you will have to make the descriptions yourself or provide the tables to an open data portal through a separate channel, as the Central Bank of Brazil does. And the combination "PxWeb for people, .Stat Suite or SDMX-RI for exchange" is a normal setup, not redundancy.

How to choose

The choice begins not with a product but with who takes the data. If only people, a showcase is enough. If programs and international organisations, you need an API, SDMX and bulk download.

Situation	What to take	Why, and what it costs
The office publishes tables, SDMX exchange is not yet required of it	PxWeb, new installations straight on PxWebApi 2	quick, API and JSON-stat in the core; SDMX will have to be added later
Exchange with Eurostat, the IMF, the OECD is needed	.Stat Suite or SDMX-RI, structures in Fusion Metadata Registry	SDMX in the core; implementation is heavy and rarely done without consultants
A clear Sustainable Development Goals showcase is needed	OpenSDG	static website and full export; weak as the only exchange channel
A ministry collects health statistics	DHIS2	collection and publishing in one system; the public part must be configured separately
A city or region wants a showcase of its indicators	a local vendor such as Swing, or a section on a regional platform	quick and inexpensive; the data stays with the vendor, so agree on a full export in advance
Dozens of indicators, updated once a year	no portal: CSV or Parquet with a field dictionary on the website or in Git, plus transmission to the national office	cheap and durable; no slice builder
Charts for management are needed	a dashboard on top of the export, not instead of it	dashboards in the registry have almost no API

A custom build is justified when the office already has a team and wants a showcase that ready-made products do not offer. Models are the portals of the ECB, Statistics Canada and Malaysia's OpenDOSM. But all three have their own website on top of an open API and bulk download, not instead of them. If the team is not ready to build and maintain an API, its own showcase will share the fate of the 1,505 custom catalogs, of which programs can read only 8%. The choice for a statistical office is worked through in the worked example, and the general logic is in the chapter "Build, Open Source or SaaS".

When you do not need your own catalog

An agency or a city with a few dozen indicators often does not need a separate catalog. It can transmit the series to the national statistical office, publish them on an open data portal as tables with a field dictionary, or publish CSV and Parquet in a Git repository with a description of the indicators. For such series to be found, each file needs a methodology, a unit of measure, territory codes from a shared code list and an update date. Ways to publish without a portal are in the part "Publishing Without a Portal".

What makes an indicator catalog convenient

Six signs that statistics have been published as data rather than as a website with charts.

1. **Shared code lists.** Territories, industries and goods are coded with the same code lists for all indicators. Then any series can be read by one program. Eurostat has shared SDMX code lists for all datasets.
2. **Bulk download.** The whole database can be taken without clicking through each indicator. FAOSTAT provides an archive for each thematic database and a machine-readable list of archives. Eurostat publishes an inventory of all datasets with links to each file in TSV, CSV and SDMX. The ILO has a separate ILOSTAT bulk download page.
3. **Unified search.** One search finds both an indicator in the database and a table in a yearbook, and an archived publication.
4. **An ordinary open data portal as a channel.** Indicators are published as datasets on a standard platform with its API. The Central Bank of Brazil does this: its open data portal runs on CKAN.
5. **Several formats and code.** Not only Excel: CSV, SDMX, JSON-stat, Parquet, formats of statistical packages, an example request in Python or R right on the indicator page.
6. **Technical documentation.** An API description in OpenAPI, example requests, limits and licence stated explicitly. Documentation as web pages with examples, not a contractor's PDF report. Statistics Sweden's PxWebApi 2 serves its settings (a limit of 150,000 values per request, 30 requests per 10 seconds, the CC0 licence and the list of formats) at a separate address.

A catalog with only a table builder fails on points two, five and six, even if it looks modern.

Examples

Good indicator portals are organised similarly: one database and several entrances for different people. An analyst needs a chart, a programmer an API, someone building their own database a full download, and an author of a paper a persistent link to a series.

National offices and central banks

- The European Central Bank data portal serves all these audiences at once. Each series has a chart and a persistent address such as `data.ecb.europa.eu/data/datasets/EXR/EXR.D.USD.EUR.SP00.A` that can be cited. For

programs there is an API following the SDMX 2.1 standard, and each dataset can be downloaded whole. There were 108 datasets in 2024 as well; at that time together they took 93 GB in CSV and 664 MB after conversion to Parquet.

- The BIS data portal is a model of working with SDMX. There is an API, a persistent link to a series and a bulk download section where each theme is downloaded as one archive in CSV or SDMX.
- Statistics Canada has turned the office website into a data catalog. In October 2026, 8,270 tables were available through its API, about 5,100 of them active and the rest archived. Each table can be downloaded whole in CSV or SDMX. Tables from the same database are attached to analytical texts.
- Mexico's INEGI brings together everything about one survey on the tabs of a single page: documentation with the methodology and a description in the DDI standard, tables, microdata, open data and tools. There is an API for the indicator bank and a bulk download section by territory, theme and year.
- Croatian Bureau of Statistics copes with fragmentation through a single catalog. A theme page lists in one place first releases, yearbooks, PC-Axis tables, short-term statistics databases and geostatistics layers whose metadata is kept in GeoNetwork. The user does not need to know which system holds the table they need.
- OpenDOSM is the open-source portal of Malaysia's statistical office. Each dataset is in CSV and Parquet, the page has an example of reading it in Python, and there is an API for programs. The national portal data.gov.my runs on the same platform.
- Statistics Sweden and Statistics Norway have moved their databases to PxWebApi 2: a query to any table is assembled in the address, and the response is JSON-stat 2 by default.

Platforms and international databases

- OECD Data Explorer: .Stat Suite.
- Statbank Denmark: PxWeb.
- Eurostat data browser: series with breakdowns, not files by year; API in SDMX and JSON-stat.

Aggregators

- DBnomics: CEPREMAP's open-source series aggregator. It collects data from statistical offices, central banks and international organisations and serves it through an API and clients for Python, R, Julia and Stata. It is convenient when you need series from many countries in one form, but it is a secondary source: check against the original.
- Our World in Data: a showcase with its own platform, not national statistics.

The indicators and microdata of one office are two catalogs. Why they are not glued into one custom website is worked through in the statistical office example.

Where it breaks

A table builder instead of data. A table is available only through a builder: there is an export but no API. The indicator methodology lives in a PDF, not next to the series. The number of "datasets" is inflated by publishing one indicator as a separate file for each year.

The export repeats the screen. In the Russian EMISS (fedstat.ru) in 2022 an indicator opened in a "default view" (the last few years and several regions), and only this slice could be downloaded. To get the full series you had to tick all the breakdowns by hand.

Everyone has their own code lists. In the same EMISS there were almost 700 code lists for seven thousand indicators, and many of them duplicated each other. Eurostat at the same time had 283 code lists for 10,500 datasets. Without shared code lists, series from different agencies cannot be joined.

Series stop. In June 2025 a full download from EMISS managed to retrieve 6,905 indicators out of 8,773 declared, and 41.8% of them had not been updated since 2022. Some agencies' series stopped much earlier: the Rosselkhoznadzor's last update was in 2014. An independent count by the project "Esli byt tochnym" ("To Be Precise", August 2025) gave a similar picture: of 7,280 available indicators, about 3,700 were current and updated.

The system is closed down. In May 2025 it became known that by an order of the Ministry of Digital Development and Rosstat, EMISS was being decommissioned by 31 December 2025. At the time the text of the order was not on the agencies' websites or on the official publication portal. The replacement is the state system "Digital Analytical Platform", whose launch has been postponed several times since 2022; according to "To Be Precise", the transition has been postponed to January 2027. In November 2025 EMISS was still working: according to a web archive snapshot, 8,875 indicators from 61 agencies. In 2026 the website does not open from abroad, and its state cannot be checked from outside. The conclusion for a user: download a series you need for a long time in full and keep it yourself. How a public archive preserved EMISS is described in the chapter on archiving.

Publications are scattered. Statistics of different levels may not come together in one place. The regional offices of Rosstat each have their own website with their own set of yearbooks, some of which are sold by subscription, and there is no single catalog of these publications (2024).

The vendor leaves. Knoema is a cloud indicator service on which the registry lists 125 showcases, mostly African: statistical offices, central banks and the Open Data for Africa platform. In 2020 the company was bought by the Eldridge holding; in 2023 it changed its name to Seek and turned towards analytics for retail. Since January 2026 knoema.com does not open: the last successful web archive snapshot is from 1 December 2025. In October 2026 the Open Data for Africa portals respond to programs with a Cloudflare check instead of data. For an office that keeps its statistics in a cloud service, this is the main risk: write a full export of the database and code lists in an open format into the contract in advance.

The platform is not upgraded. Reusable software helps only if it is upgraded. The Armenian Statbank (statbank.armstat.am) ran for a long time on an old version of PxWeb on ASP.NET: the standard API did not respond, and getting the data required writing a separate parser (November 2025). In October 2026 it runs the 2025 release of PxWeb, and the standard API responds. The upgrade returned the data to common tools without new development.

The API generation changes. The move to PxWebApi 2 changes request addresses: they have become more stable, but users' old scripts stop working. Statistics Sweden released a converter from old links to new ones. An office moving to the new generation should keep the old API until

users have migrated and announce a date for switching it off.

SDMX is hard to implement. SDMX is needed for exchange with international organisations, but it is hard to implement. According to a 2026 estimate, there are only three server products under active adoption, and they are rarely installed without consultants. That is why outside national offices and international organisations SDMX is almost never seen, nor is the lighter JSON-stat. A small office would do well to start with bulk download in CSV or Parquet and a documented API, and add an SDMX endpoint with a ready-made product rather than writing one itself.

A new portal loses the old. In 2024 the IMF data portal provided datasets for bulk download. The new data.imf.org, opened in spring 2025, gained a dataset catalog, cross-cutting search, dashboards and a new SDMX API. But in April 2025 it was impossible to download everything at once, there was no machine-readable list of datasets, no schema.org or DCAT markup, you could not link to an individual series, and the API required registration. In October 2026 the IMF SDMX API responds without a key; the rest is worth checking again.

A dashboard instead of data. Power BI, Tableau and R Shiny account for about a hundred indicator showcases in the registry, and only a handful have programmatic access. The chart is visible, but the series behind it cannot be retrieved. More in the anti-pattern "Dashboard instead of data".

How to check an indicator catalog

Four steps for a quarter of an hour.

1. **Take one series and try to cite it.** Does the series have an address that will open with the same slice for another person?
2. **Download a whole table.** Does the file match the full table or only the slice shown on the screen?
3. **Find the API and its documentation.** Is registration needed, what are the limits, what is the licence of the data?
4. **Check the code lists.** Can the list of region codes be downloaded, and are the codes the same in two different tables?

Typical addresses for different platforms:

```
# SDMX REST, ECB: latest value of a series in CSV with attributes
curl -s "https://data-api.ecb.europa.eu/service/data/EXR/D.USD.EUR.SP00.A?
lastNObservations=1&format=csvdata"
# SDMX REST: list of tables (dataflows) at the OECD, IMF and Eurostat
curl -s "https://sdmx.oecd.org/public/rest/dataflow/all"
curl -s "https://api.imf.org/external/sdmx/2.1/dataflow"
curl -s "https://ec.europa.eu/eurostat/api/dissemination/sdmx/2.1/dataflow/ESTAT/all/latest"
# Eurostat: a table slice in JSON-stat 2.0
curl -s "https://ec.europa.eu/eurostat/api/dissemination/statistics/1.0/data/demo_pjan?
geo=FR&sex=T&age=TOTAL&lastTimePeriod=1"
# PxWebApi 2: limits, formats and licence of the installation
curl -s "https://statistikdatabasen.scb.se/api/v2/config"
# PxWebApi 2: how many tables are in the database
curl -s "https://data.ssb.no/api/pxwebapi/v2/tables?lang=en&pageSize=1"
```

A response at an SDMX or PxWebApi address is a good sign. If there is no API but only a table builder, check whether there is an inventory for bulk download, as at Eurostat and FAOSTAT. The summary list is in the checklists, and general commands are in the chapter on APIs and bulk download.

How it differs from its neighbours

Neighbour	Main difference
Open data portal	the unit of publication is an indicator with breakdowns, not a file; one table replaces hundreds of files by year and region
Microdata catalog	aggregates rather than respondent records; access is usually open without requests
Geoportal	an indicator is tied to a territory code, not to geometry; the map is built from a code list shared with the geoportal
Metadata catalog	an SDMX structure registry stores code lists and table descriptions, while an indicator catalog serves the values themselves
Dashboard	a catalog has search, a series address and export; a dashboard has only a picture

A summary table of all types is in the chapter Comparing Types and Hybrids.

Sources

- dataportals-registry export, release 1.27.0 of 15 October 2026: number of catalogs, owners, countries, platforms with the share having an API, SDMX and DCAT addresses. The queries are in the appendix.
- SDMX, the SDMX 3.1 release, May 2025; JSON-stat; StatDCAT-AP and draft 3.0.0; RDF Data Cube
- FAOSTAT bulk download, Eurostat dataset inventory, ILOSTAT bulk download, Central Bank of Brazil open data portal; checked 12 October 2026
- PxWeb: the old line and its releases, PxWebApi 2 at Statistics Sweden, PxWebApi 2 user guide, releases of PxWebApi and PxWeb 2; Statistics Sweden API settings and Statistics Norway tables; the old Statistics Sweden API response in SDMX-ML 2.0; checked 12 October 2026
- .Stat Suite and Data Explorer releases; SDMX-RI and its components under the EUPL; the SDMX-RI service image for .Stat Suite; Fusion Metadata Registry and its licence, Eurostat SDMX registry; checked 12 October 2026
- DHIS2 and its releases; OpenSDG; PxStat; DBnomics and its API; checked 12 October 2026
- Knoema: acquisition by Eldridge, 21 December 2020; merger with Seek, 15 February 2023; web archive snapshots of knoema.com: the last successful one, 1 December 2025, followed by 530 errors; responses of knoema.com and opendataforafrica.org portals checked 12 October 2026
- ECB data portal: list of datasets with bulk download, SDMX API description; BIS bulk download, BIS API; INEGI indicator bank API, INEGI bulk download, the ENIGH 2022 survey page; Statistics Canada web services; Croatian statistics portal; OpenDOSM and its code; Armenian Statbank, IMF SDMX API; checked 12 October 2026

- I. Begtin, Statistics as a data product (in Russian), August 2024: levels of statistics publishing and signs of a modern portal
- I. Begtin, Why it is impossible to praise Rosstat (in Russian), January 2022, and Russian statistics: non-machine-readable institutional fragmentation (in Russian), December 2024
- "Esli byt tochnym" ("To Be Precise"), In 2025, 140 datasets disappeared in Russia and 425 indicators stopped being updated (in Russian), 6 August 2025: the state of EMISS and the postponed move to the Digital Analytical Platform; web archive snapshot of fedstat.ru, 1 November 2025
- I. Begtin, Telegram (in Russian): EMISS: code lists and the default view, 30 April 2021; the PX format on open data portals, 8 January 2023; the ECB data portal, 23 July 2024; ECB data volume in CSV and Parquet, 20 August 2024; INEGI, 12 April 2024; Statistics Canada, 20 November 2024; Croatia, 3 December 2024; OpenDOSM, 24 March 2023; DBnomics, 19 November 2024; reusable statistical software, 1 February 2025; IMF data, 31 July 2024; the new IMF portal, 19 April 2025; EMISS to be decommissioned, 12 May 2025; the state of EMISS, 20 June 2025; Armenian Statbank, 27 November 2025; the difficulty of SDMX, 3 September 2026
- Units of Publication and Formats

Microdata Catalogs

A microdata catalog describes censuses and surveys: who was asked, how the sample was drawn, what questions were asked, what each variable means. The description comes with files of records about individual people, households or businesses: anonymised, but still sensitive. That is why the description here is almost always open, while the file is often provided after registration, on request or only in a secure environment.

According to the dataportals-registry export (release 1.27.0 of 15 October 2026), there are 326 such catalogs, less than 1% of all catalogs. Half of them (49%) have some files closed, and this is normal for the type. But the share of dead websites here is the highest of all types: 59 catalogs, about 18%, do not open.

The main conclusion of the chapter: a microdata catalog is valuable for its description, not for its file. A study that cannot be found by its variables and for which there is no explanation of how to get access does not exist for a researcher, even if the files are on the server. A good catalog openly shows everything except the records themselves, serves descriptions to programs in the DDI standard and says honestly who will provide the data, on what terms and how quickly.

How a survey description is organised

On an open data portal the main thing is the file; in an indicator catalog, a number with breakdowns. Here the main thing is the study, which has three levels of description.

- **Study**: one census or one survey wave: who conducted it, when, in what territory, who was interviewed (the unit of observation and the population), how the sample was drawn, how responses were collected.
- **Data file**: one table of records: households, household members, land plots. A single study can have more than a hundred files.
- **Variable**: one column of a file: name, label, question text, permitted values with labels, number of responses for each value, missing values, a weight flag.

Next to them are **materials**: questionnaires, interviewer instructions, a sampling report, code for calculating weights. And **access conditions**: the access type, the text of the agreement, how to cite the data.

What this looks like in practice can be seen with the Nigeria General Household Survey Panel 2023–2024 in the World Bank Microdata Library. On 12 October 2026 the catalog API returns for it:

Level	What the description contains
Study	identifier `NGA_2023_GHSP-W5_v02_M`, National Bureau of Statistics of Nigeria, units of observation: households, individuals, plots and communities; sample description; DOI `10.48529/90ec-pv69`
Files	108 files, each with a description: which section of the questionnaire and at what level the data was collected
Variables	4,531 variables. The variable `zone` has six values, from "1. North Central" to "6. South West", and the number of records for each: 851, 824, 870, 835, 839, 848, 5,067 in total

Level	What the description contains
Access	public use files: download after registration and agreeing to the terms: do not copy for others, do not try to identify respondents, cite the source as shown

The whole description can be downloaded in DDI Codebook 2.5, and the DOI leads to the study page and returns a ready-made BibTeX citation. From the value frequencies a researcher sees, before applying, whether there are enough observations for the region they need. This is the main job of a microdata catalog: to let people decide whether they need the data without handing it over.

What gets its own page

In a microdata catalog the study almost always gets its own page, and in good catalogs each variable does too. That is why the number of "datasets" here cannot be compared with open data portals: one study is a hundred files and thousands of variables. On 12 October 2026 the World Bank Microdata Library had 7,145 studies, the IHSN central catalog 13,085, and the Census Digital Library of India 40,268 records. The Maelstrom Research catalog on OBiBa Mica describes 473 epidemiological studies and about 2.05 million variables: search works by variable, not by study title.

Access levels

Access levels are what distinguish a microdata catalog from other types. This is not a flaw but a way to give data to those who need it without exposing people. Removing names is not enough: a person is often given away by a combination of age, occupation and a small settlement (more in the entry "Anonymisation" in the glossary). The more detailed the file, the stricter the access.

NADA, the most common platform of the type, has seven access types:

Access type	Login needed	What the user does	Approval	Where the data comes from
Open data	no	accepts the terms	no	from the catalog, in open formats such as CSV
Direct access	no	accepts the terms	no	from the catalog, often in SPSS or Stata formats
Public use files	yes	fills in a request form	no	from the catalog
Licensed data	yes	submits an application with the research purpose	yes	from the catalog after approval
Data enclave	no	downloads an application form	outside the site	only in a secure room or environment
External repository	no	follows a link	—	from the data owner's website
Data not available	—	—	—	description only

The last two types matter no less than the others. A catalog can be useful even if it does not store a single file itself: in the World Bank Microdata Library on 12 October 2026, for 67% of studies (4,790 of 7,145) the data sits in an external repository: with the statistical office, in the DHS

programme or in IPUMS. Public use files account for 18%, direct access for 12%, open data and licensed access about 2% each.

The most detailed data (tax, health, linked registers) is not handed out at all. It is used in a **trusted research environment**: an approved researcher logs into a secure system, runs analyses there and takes away only checked results. This is how SecureLab at the UK Data Service and the French CASD work. The Norwegian microdata.no, run since 2018 by Sikt and Statistics Norway, has gone further: staff and students of approved organisations analyse register data online without an application, but see only the results of calculations, not the records. According to a 2025 OECD report on 16 statistical offices, this role of a "trusted intermediary" is becoming normal for offices. Another route is **synthetic data**: artificial records with the same statistical properties, which can be given to everyone.

Three requirements follow for a catalog. The description is open at any access level. The terms are written down before an application: who can get the data, what has to be attached, how long to wait for an answer. And the page shows the level of each file, not only of the study as a whole. How this works in different platforms is covered in the chapter on access and roles.

Who builds them

More than half of microdata catalogs belong to central government, mostly national statistical offices.

Owner	Catalogs	Share
Central government	179	55%
Universities and research organisations	104	32%
International organisations	20	6%
Civil society organisations	14	4%
Regions	4	1%
Companies and others	5	2%

- **Statistical offices** publish censuses and surveys of households, the labour force, budgets, health. In developing countries a catalog often appeared with the support of World Bank programmes and the IHSN network, which is why 59% of office catalogs run on NADA.
- **University social data archives** hold opinion polls, panel and cohort studies. Platforms here are more varied: only 21% use NADA, alongside OBiBa Mica, Colectica, Dataverse and their own systems.
- **International organisations**: the World Bank, WHO, FAO, UNHCR, the Pacific Community. 18 of their 20 catalogs run on NADA.
- **Survey consortia** such as Afrobarometer, Arab Barometer, Latinobarómetro and Asian Barometer publish their waves on ordinary websites: on WordPress or their own builds.

Microdata catalogs exist in 139 countries, and another 27 cover the whole world. No country has more than two dozen: the US has 18, Canada 12, Germany 11. The type is organised so that a country usually has one or two statistical office catalogs and several university archives.

Why 326 is a lower bound

The number of microdata catalogs in the registry greatly understates the world of surveys. The largest social data archives (the UK Data Service, ICPSR, GESIS, the Finnish FSD, the Swedish SND, the Australian Data Archive, the Norwegian Sikt) are recorded in the registry as research repositories or open data portals: they have DOIs, long-term preservation and data not only from surveys. The European CESSDA catalog, which collects descriptions from national archives, served about 33,000 study descriptions on 12 October 2026. So to search for surveys, look also at the chapter on research repositories: the boundary between the two types runs through the social science archives.

What they publish

- **Population censuses**: full or sample record files. In Latin America they are distributed through REDATAM, in the rest of the world through NADA and IPUMS.
- **Household surveys**: budgets, living standards, the labour force, health. Many of them are international programmes with a common questionnaire: the World Bank's LSMS, DHS, UNICEF's MICS, WHO's STEPS.
- **Opinion polls and panels**: the European Social Survey, the German SOEP, the Russian RLMS-HSE, the Chinese CFPS.
- **Cohorts and biobanks**: epidemiological studies where people are followed for years: CanPath and CLSA in Canada, the European projects RECAP Preterm and EUCAN.
- **Business and bank microdata**: the research data centres of the Bundesbank, the Bank of Italy and the Bank of Portugal.
- **Administrative registers**: taxes, employment, benefits, linked by a personal identifier. They are almost never handed out as files, only in a secure environment.
- **Code and data for reproducing research**: the World Bank Reproducible Research Repository on the same NADA: 585 records in October 2026 compared with 163 in 2024.

What a microdata catalog usually lacks: names, addresses and exact coordinates of respondents (they are removed before publication), ready-made indicator series (that is an indicator catalog), and for closed studies, the files themselves.

Standards

DDI (Data Documentation Initiative) is the main standard of the type. It is maintained by the DDI Alliance, and on 6 March 2026 the common part of the DDI standards was published as ISO/PAS 25955:2026. It is not one format but a family:

- **DDI Codebook**: a description of one study and its files, an "electronic codebook". This is what NADA, Dataverse and the European CESSDA catalog serve. Version 2.5 came out in 2012 and remains the main one. Version 2.6 came out on 15 April 2026, and support for it in platforms is only beginning.
- **DDI Lifecycle**: a description of the whole cycle: the questionnaire, reuse of questions and variables across waves, processing, versions. Version 3.3 (2020). It is needed by longitudinal studies and offices that maintain a question bank. Colectica runs on it.

- **DDI-CDI**: a new model for data in any domain; version 1.0 came out in 2025. It describes not only surveys but also links between data across domains and the provenance of each value. It is still almost never seen in catalogs.

Controlled vocabularies work alongside DDI: shared lists of values for the mode of collection, the unit of observation, the sampling procedure. They are maintained by the DDI Alliance and CESSDA. Without them the "mode of collection" field is filled in differently in two archives, and a common search by it does not work.

Exchanging descriptions. Harvesters take DDI via the OAI-PMH protocol. This is how the CESSDA catalog works: it accepts descriptions only via OAI-PMH in DDI Codebook 2.5 and according to its own metadata profile, checked by a separate validator. Dataverse serves descriptions via OAI-PMH in DDI in its core. NADA has no OAI-PMH: descriptions are taken through its own API or downloaded in DDI one study at a time. That is why a catalog on NADA cannot be connected to the European federation of archives directly, without an intermediate export.

For search engines what is needed is not DDI but schema.org/Dataset markup and a DOI. NADA puts JSON-LD markup on the study page in its core. DOIs are issued by DataCite, as for research data; by October 2026 the World Bank had registered about 3,000 DOIs for pages of its microdata library. More in the chapters on metadata export and on identifiers.

Description tools. DDI is rarely written by hand. For surveys there is the World Bank Metadata Editor, open source (MIT) since 2025, an editor for describing microdata, documents, tables and geodata; NADA has used its templates since release 5.7. For data left in Nesstar Publisher, IHSN has published on GitHub a tool for exporting DDI and data from its files (updated in February 2026). The commercial alternative is Colectica Designer.

Platforms

Platform	Catalogs in the registry	What it is	Do not open
NADA	158	open-source (MIT) platform of the World Bank and IHSN	47
Custom build	43	office and archive websites; programmatic access marked for 12%	0
REDATAM	38	free software from UN ECLAC for censuses, mainly Latin America	6
OBiBa Mica	27	open-source (GPL-3.0) catalog of epidemiological studies	3
Nesstar	13	discontinued	2
IPUMS	13	websites of the University of Minnesota harmonisation project	0
Colectica	11	commercial platform on DDI Lifecycle, mostly longitudinal studies	0
WordPress, Drupal	11 and 7	archive and survey websites	0

The type rests on one open-source platform. NADA is almost half of the catalogs, and almost all of the type's dead websites are NADA too. This is not a flaw of the software but a trace of how it was installed; see the section "Where it breaks". Custom catalogs account for only 13%, less than for any large type: the task is narrow, and a ready-made solution exists. But the custom ones have almost no programmatic access.

Main platforms

- **NADA** is an open-source microdata catalog developed by the World Bank together with the IHSN network. Releases come often: 5.6 in July 2026, 5.7 in September, 5.7.1 on 4 October 2026. Access levels, applications for licensed data, DDI import and export, and search by variable are in the core. Since release 5.6 NADA describes not only surveys but also indicators with time series, documents, tables and geodata; permissions can be granted by user and collection, and there are built-in view and download statistics. For administrators there are API clients for R and Python (NADAR and PyNADA).
- **REDATAM** is software from ECLAC's Population Division (CELADE), developed since 1985. It stores censuses in its own format and builds tables from them for any territory, down to a city block. ECLAC distributes it free of charge and calls it free software; according to its website, it has helped 34 countries. The current generation is RedatamX (since 2024): it runs on Windows, Linux and macOS and as the `redatamx` package for R. The previous Redatam7 runs only on 32-bit Windows; its last release was in March 2021. Online processing lets users calculate tables right on the office's website without downloading records.
- **OBiBa Mica** is an open-source catalog of epidemiological studies and consortia: detailed variable dictionaries, study networks, access requests. Releases come regularly (6.3.7, September 2026). It works together with the OBiBa Opal data store. On such portals data is almost always provided only on request.
- **Colectica** is a commercial software suite from a US company built on DDI Lifecycle: an editor, a metadata repository and a portal. Statistics New Zealand, Ireland's statistical office, the British CLOSER and American longitudinal studies run on it. In August 2026 the company announced Colectica Repository 8 with an MCP server through which an AI agent reads DDI descriptions.
- **IPUMS** is not a reusable platform but a project: the University of Minnesota collects censuses and surveys from many countries and harmonises the variables so that countries and years can be compared. In October 2026 IPUMS International covered 104 countries, 656 censuses and surveys and more than a billion person records. Data is provided free of charge after registration, as extracts on request.
- **Dataverse** is listed among research repositories in the registry, but it is often chosen for microdata: DDI descriptions, restricted-access files, DOIs and OAI-PMH are in its core. When SPSS or Stata tables are uploaded, it extracts variable descriptions itself. The Canadian ODESI (into the Borealis repository) and the Australian Data Archive moved to Dataverse.

Core and add-ons

Capability	NADA	Dataverse	OBiBa Mica	REDATAM
File access levels	core, seven types	core: open and restricted files, access requests	core: descriptions open, data on request	data on the office's server, only tables go out
Applications and their log	core	core	core	no
Variable descriptions	core, from DDI	core, extracted from SPSS and Stata files	core, the main object of the catalog	core, the database dictionary
DDI export	core, DDI Codebook	core, DDI Codebook	not checked	no, own format
OAI-PMH	no	core	not checked	no
schema.org	core (JSON-LD)	core	not checked	no
DOI	a field for a DOI minted separately at DataCite	core	not checked	no
Citation export	core, via DataCite if there is a DOI	core	not checked	no
API	core	core	core	REST service and an R package
Online calculations without releasing records	no	not in the core	no	core

Two conclusions follow from the table. NADA covers the essentials (access levels and DDI descriptions), but its descriptions reach the European federation of archives and research data harvesters only through an intermediate export, and DOIs have to be registered separately. Dataverse provides DOIs and OAI-PMH, but has no seven access types and no online calculations. That is why NADA usually suits a statistical office better, and Dataverse a university archive.

Nesstar: what to do with the legacy

Nesstar is a software suite from the Norwegian NSD and the UK Data Archive on which many archives and statistical offices ran in the 2000s and 2010s. With version 4.0 in 2015 it was declared obsolete, and by 2022 it was finally discontinued; Nesstar Publisher is distributed without support. Statistics Canada switched off its Nesstar in January 2022 for cybersecurity reasons. The Canadian ODESI chose Borealis on Dataverse for its migration, and in October 2026 the former address of its Nesstar redirects to the new website, while ODESI collections are visible in Borealis as separate OAI-PMH sets. At GESIS the address of the ZACAT Nesstar catalog leads to the general GESIS search. The UK Data Service still keeps Nesstar, but in its accessibility statement of 21 August 2026 it says that the application will eventually be retired, so accessibility issues in it will not be fixed.

Nesstar is not chosen for a new catalog. If a catalog still runs on it, a migration plan is needed before the website stops opening: export descriptions to DDI and data with the IHSN tool, choose a platform for the task (NADA for a catalog with access levels, Dataverse for a repository with DOIs) and keep redirects from the old addresses.

How to choose

The choice begins with two questions: who will receive the data, and how detailed the records are.

Situation	What to take	Why, and what it costs
A statistical office publishes surveys with different access levels	NADA	access levels and DDI in the core; OAI-PMH and DOI registration have to be added separately
A university archive needs DOIs, long-term preservation and exchange with CESSDA	Dataverse, or joining the national social data archive	DOIs and OAI-PMH in the core; no seven access types
An epidemiological cohort or consortium, data only on request	OBiBa Mica with Opal	detailed variable search and applications; almost no data goes out, search engines skip such catalogs
A census with tables for small areas	REDATAM for online tables plus a DDI description and a file in an open format	quickly provides tables; without an open file and DDI the data remains in one program's format
A longitudinal study with a question bank and many waves	Colectica or other DDI Lifecycle tools	versions of questions and variables across waves; commercial licence
Registers and other data that cannot be handed out as a file	a trusted research environment, with a catalog of descriptions only	data does not leave the office; the environment costs more than any catalog
A few surveys a year, no team of your own	hand descriptions and files over to the national archive, IHSN or the World Bank library	cheap; the showcase is not your own

A custom build is almost never justified in this type: NADA is free, it had three releases in 2026, and access levels are already built in. Of 43 custom catalogs in the registry only 12% have programmatic access marked. If an office keeps its own system, as Statistics Korea or the Australian Bureau of Statistics do, it should at least serve study descriptions in DDI. How a statistical office should separate indicators and microdata is worked through in the statistical office example, and the general logic of choice is in the chapter "Build, Open Source or SaaS".

When you do not need your own catalog

A research group or agency with one or two surveys does not need its own catalog. The survey can be deposited in a national social data archive or a generalist repository with restricted file access, and the description in the IHSN catalog. If the data can be opened, a CSV or Parquet file with a variable dictionary and a sample description in a Git repository or on Zenodo is enough. For such a survey to be found, the description needs the unit of observation, territory and period, sampling procedure, list of variables and access conditions. Ways to publish without a portal are in the part "Publishing Without a Portal".

What makes a microdata catalog convenient

1. **The description is always open.** A study is visible without login, even if the files are closed. Someone who does not know about data cannot request it.
2. **Search by variable.** Each variable has its own page with the question text, values and frequencies. A researcher finds the question they need rather than going through survey titles.

3. **Terms written down in advance.** The access type is shown for each file, and the terms and response time are stated before an application.
4. **Descriptions for programs.** DDI export, a catalog API, and for search engines schema.org markup and a persistent identifier.
5. **How to cite.** A study has a DOI or a persistent address and a ready-made citation; repeated releases have versions.
6. **Open formats.** Open files are in CSV or Parquet, not only in SPSS and Stata formats that require a specific program.

Examples

International catalogs

- World Bank Microdata Library: NADA, 7,145 studies from many countries; two thirds lead to data owners' websites, and some have DOIs.
- IHSN central catalog: NADA, 13,085 survey descriptions, mostly from developing countries.
- WHO NCD microdata repository: NADA, 937 STEPS, tobacco and school health surveys. For WHO this is one of several catalogs of different types, as for the Pacific Community with its Pacific Data Hub.
- IPUMS International: harmonised censuses of 104 countries.

National catalogs

- Census Digital Library of India: NADA, 40,268 records, censuses since 1901 with detail down to districts. In 2023 there were more than 29,000 records.
- Armstat microdata catalog: NADA, 26 surveys.
- The national statistical offices of Bolivia (121 studies) and Tanzania (62): NADA with files at different access levels.
- INEGI: microdata on the tab of each survey next to tables and a DDI description; covered in the chapter on indicator catalogs.

Social data archives

- UK Data Service: open, safeguarded and controlled data; the most detailed only in the SecureLab environment.
- CESSDA Data Catalogue: a common search across European archives: about 33,000 descriptions harvested via OAI-PMH in DDI.
- ODESI: a Canadian search of social surveys whose data moved from Nesstar to Borealis on Dataverse.

Health and secure access

- Maelstrom Research: OBiBa Mica, 473 epidemiological studies and about 2.05 million described variables.
- microdata.no: Norwegian register data since 1964: online analysis without an application, only results go out.

Russia

- RLMS-HSE: the Russia Longitudinal Monitoring Survey of HSE University, a panel survey since 1994; data after registration.
- The FOM database and the sociological data bank of the Institute of Sociology of the Russian Academy of Sciences: opinion polls on their own websites.
- Microdata from Rosstat's sample surveys is published on separate pages of the agency's website and in the obdx.gks.ru system. In 2023 the data there was last updated in 2021, and the software had not changed for about nine years. In October 2026 neither address opens from abroad, and their state cannot be checked from outside.

Where it breaks

The installation is abandoned after the project. Of 59 dead microdata catalogs, 47 are on NADA. Almost all are old statistical office catalogs in Africa, Asia and Latin America: on IP addresses instead of domains, on contractors' subdomains, without HTTPS. Among catalogs without HTTPS, 38 of 58 do not work; among those with HTTPS, 21 of 268. The catalog was installed as part of a support programme, and when the programme ended, there was no one to maintain it. For 18 of the 38 countries with an abandoned NADA, the registry has no other active microdata catalog. It also happens that the website opens but the catalog is empty: the NADA of Madagascar's statistical office responds on 12 October 2026, but holds zero studies. The lesson: install a microdata catalog on the office's own domain and budget for maintenance, not just installation.

The product is withdrawn. Nesstar is an example of the platform itself being closed down. In the registry this shows less than one might think: of 13 records on Nesstar two are marked as dead, but another two addresses, at GESIS and ODESI, in fact lead to new systems. The main risk is not that the website disappears, but that descriptions and files remain in a format that nobody supports any more.

One program's format. Latin American censuses have been distributed through REDATAM since the 1990s, but the database is stored in its own format without a published description: tables are built in the program itself or in its online processing. To work with the records in ordinary tools, researchers built the open converter Open Redatam (Apache 2.0): it reads the database dictionary and exports tables to CSV, with packages for R and Python. There is also the `redatamx` package for R, a wrapper around REDATAM itself, with CELADE holding the rights to the core. Both routes work, but without them the records are inaccessible to ordinary software.

The catalog is closed to programs. On 12 October 2026 the FAO and Pacific Community microdata catalogs respond to programs with a Cloudflare check, and the UNHCR catalog with a 403 error, with the API closed too. People in a browser see the data, but harvesters, dataset search engines and web archives do not. For a catalog whose value lies in its descriptions, this is especially unfortunate: the study cannot be found where people look for it.

Descriptions, but no data. Portals on OBiBa Mica describe variables in detail but provide data only on request. That is why dataset search engines often skip them: there are no files, only descriptions. This is normal for a microdata catalog, but the description must be complete,

otherwise the study will not be found or requested.

The description is hidden or empty. A study is described only by a title and an abstract, without variables or sample, so it is impossible to decide whether the data is needed. It is worse when the description is visible only after login: then the study cannot be found at all.

Access exists on paper. Applications are accepted, but no response time is stated, there are no approval criteria, and there is no answer for months. Access exists formally but not in practice. The response time and the appeal procedure should be written on the study page.

The link points outside. When a catalog mainly points to external repositories, as the World Bank's does, its quality depends on other people's websites. If the data owner has moved, the catalog record remains but the link leads nowhere. Such links need regular checking, as in any catalog with links instead of files; see the chapter on internal storage.

How to check a microdata catalog

Five steps for half an hour.

1. **Find a study without logging in.** Are the sample, unit of observation and period visible? Is there a list of variables?
2. **Open one variable.** Is there the question text, values with labels and the number of responses?
3. **Read the access conditions.** Is the access type shown for the files, what is needed for an application and how long is the wait?
4. **Export a description with a program.** Does the catalog serve DDI, is there an API or OAI-PMH, does it open without an "I am not a robot" check?
5. **Check the citation.** Is there a DOI or a persistent address, and does the DOI lead to the study page?

Typical addresses for different platforms:

```
# NADA: how many studies, and how many of them with licensed access
curl -s "https://microdata.worldbank.org/index.php/api/catalog/search?ps=1"
curl -s "https://microdata.worldbank.org/index.php/api/catalog/search?ps=1&dtype=licensed"
# NADA: study description, its files and one variable
curl -s "https://microdata.worldbank.org/index.php/api/catalog/NGA_2023_GHSP-W5_v02_M"
curl -s "https://microdata.worldbank.org/index.php/api/catalog/NGA_2023_GHSP-
W5_v02_M/data_files"
curl -s "https://microdata.worldbank.org/index.php/api/catalog/NGA_2023_GHSP-
W5_v02_M/variables/V2"
# NADA: the whole description in DDI Codebook (the address uses the study's numeric id)
curl -s "https://microdata.worldbank.org/index.php/metadata/export/6410/ddi"
# DOI: where it leads, and a ready-made BibTeX citation
curl -sL -H "Accept: application/x-bibtex" "https://doi.org/10.48529/90ec-pv69"
# OBiBa Mica: number of studies, datasets and variables
curl -s "https://www.maelstrom-research.org/ws/studies/_rql?query=study(limit(0,1))"
# Dataverse and CESSDA: description formats via OAI-PMH, DDI among them
curl -s "https://borealisdata.ca/oai?verb=ListMetadataFormats"
curl -s "https://datacatalogue.cessda.eu/oai-pmh/v0/oai?verb=ListMetadataFormats"
```

All commands were checked on 12 October 2026. A JSON response at `api/catalog/search` is a sign of NADA, a `_rql` response a sign of OBiBa Mica. A "Just a moment..." page instead of a response is a sign that the catalog is closed to programs. The summary list is in the checklists, and general commands are in the chapter on APIs and bulk download.

How it differs from its neighbours

Neighbour	Main difference
Indicator catalog	records about people and households, not aggregates; the description is a study and variables, not a series with breakdowns
Research repository	access levels and anonymisation are mandatory, a DOI is desirable; the description is in DDI, not only DataCite. Social science archives sit at the junction of the two types
Open data portal	closed access to files is the norm, not a sign of trouble; the description is much more detailed
Metadata catalog	also shows descriptions of closed data, but does not handle applications or hand out files

A summary table of all types is in the chapter Comparing Types and Hybrids.

Sources

- dataportals-registry export, release 1.27.0 of 15 October 2026: number of catalogs, owners, countries, platforms, dead catalogs, catalogs without HTTPS. The queries are in the appendix.
- DDI: DDI Codebook, DDI Lifecycle, DDI-CDI, ISO/PAS 25955:2026, controlled vocabularies; checked 12 October 2026
- CESSDA: catalog OAI-PMH, requirements for metadata providers, metadata profiles; number of descriptions via the catalog's OAI-PMH and API, 12 October 2026
- NADA: code and releases, access types, documentation; schema.org markup, the DOI field and citation export via DataCite according to the NADA 5.7.1 source code; Metadata Editor, Nesstar Publisher exporter
- World Bank Microdata Library: number of studies and access types, the description of the Nigeria panel survey 2023–2024 and its DOI, via the catalog API and DataCite, 12 October 2026; IHSN catalog, Census Digital Library of India, Armstat, WHO, World Bank Reproducible Research Repository, via the API, 12 October 2026
- REDATAM: website, FAQ, the redatamx package; Open Redatam and the redatam package; checked 12 October 2026; M. Vargas Sepúlveda, L. Barkai, The REDATAM format and its challenges for data access and information creation in public policy, Data & Policy, 2025
- OBiBa Mica and its releases; Maelstrom Research, via the API, 12 October 2026; Colectica Portal, Colectica Repository 8 and MCP, 12 August 2026; IPUMS International
- Nesstar: product history, the Nesstar Publisher entry in the CESSDA directory, Statistics Canada switching off Nesstar, January 2022; ODESI migration to Borealis; UK Data Service accessibility statement for Nesstar, 21 August 2026; responses of Nesstar addresses at ODESI and GESIS checked 12 October 2026

- Access: microdata.no, CASD; OECD, National statistical offices as emerging trusted intermediaries in data governance, OECD Digital Economy Papers, No. 378, 10 September 2025
- Responses of the FAO, Pacific Community and UNHCR catalogs to programs, a sample check of dead NADA catalogs, the Madagascar statistical office catalog: 12 October 2026
- I. Begtin, Telegram (in Russian): microdata catalogs and Rosstat, 22 March 2023; Census Digital Library of India, 8 April 2023; REDATAM, 22 August 2023; World Bank Reproducible Research Repository, 28 October 2024; portals on OBiBa Mica, 19 December 2024; Redatam and Open Redatam, 4 June 2025; WHO data catalogs, 15 September 2026
- Access, Roles and Workflows

ML Catalogs

An ML catalog publishes datasets for training and evaluating machine learning models, and often the models themselves. A dataset here is not meant to be read by a person or cited in a paper, but to be loaded into code by another researcher who compares their model with others on the same data.

The main practical conclusion of the chapter: almost no one who publishes such a dataset needs a catalog of their own. Datasets are searched for on a few large platforms, and conferences, libraries and search engines look there too. A catalog of your own is justified when the data cannot be handed to someone else's platform, and a copy of the release for long-term preservation goes into a research repository.

The scale confirms this. On 12 October 2026 Hugging Face had 1,085,895 datasets and 3,143,746 models, OpenML 6,434 active datasets, and the UCI archive 689. In the dataportals-registry (release 1.27.0 of 15 October 2026) there are only 100 ML catalogs. The registry counts websites, not datasets: Hugging Face and Kaggle are one record each for huge platforms, next to websites of a single benchmark with a few datasets. The number of registry rows says nothing about the size of this world.

What makes a catalog an ML catalog

Five things distinguish an ML catalog from an open data portal and a research repository.

- **The dataset is split.** The training part (train) is for training a model, the validation part for tuning settings, and the test part for honestly measuring the result. The split is fixed in advance, otherwise the results of different authors cannot be compared.
- **The dataset is loaded from code.** Instead of a link to an archive, the dataset page has a line of code that loads it with a library: `datasets` for Hugging Face, `kagglehub` for Kaggle, `openml` for OpenML.
- **The dataset has a card.** This is a description of the composition, the collection method, known biases and limitations: on what data a model will perform worse and what the dataset is not suitable for.
- **The licence answers the question about training.** Can a commercial model be trained on the dataset, can derived datasets be published, must the source be credited.
- **Models and results sit next to it.** Which models were trained on the dataset and what results they showed on the test part. A dataset with a shared results table is called a benchmark.

If none of this is present, you are looking at an ordinary catalog that, among other things, holds datasets suitable for machine learning. This happens with open data portals: they have many tables on which models are trained, but no splits and no cards describing biases.

Kinds of ML catalogs

The hundred registry records are varied: they are eight different kinds of websites. The classification below was made for this book from the titles and addresses of the records; the registry itself has no such field.

Kind	What it is	Examples	Records
Organisation collection	a university, lab or company publishes its own datasets	UCI archive, OpenML, Monash Forecasting, Yandex Research, Moscow State University CMC datasets	23
Hub	a shared platform where anyone uploads datasets and models	Hugging Face, Kaggle, ModelScope, OpenDataLab, Roboflow Universe	15
Competition and benchmark platform	a task, data, a hidden test part and a results table	Codabench, EvalAI, Grand Challenge, Zindi, DrivenData, Tianchi	14
Single-benchmark website	a project page with several datasets and a results table	Open Graph Benchmark, Therapeutics Data Commons, DataComp	14
Government AI data platform	datasets the state collects or funds for its developers	AI Hub (Korea), AIKosh (India), NAIRR (US), AI-on-Demand (EU), the CIT registry (Russia)	13
Library catalog	a list of datasets and models a particular library loads	TensorFlow Datasets, Keras, PyTorch Hub, NVIDIA NGC	9
List and guide	a selection of links to datasets, mostly on other hubs	Hugging Face dataset collections by language, Open Corpus Registry, Post-Training Data Atlas	9
Data vendor	a company sells ready-made and custom training datasets	Datatang, GTS AI	3

The kind determines what to check. For a hub, the card and licence of a specific dataset, not of the platform as a whole. For a competition platform, the terms of using the data after the competition ends. For a library catalog, where the library actually downloads the files from. For a government platform, who can get the data at all. Data vendors are closer to marketplaces; they are covered in the chapter on other types.

Who builds them

ML catalogs are run mainly by universities and companies. Governments are fewer here than in other types, but their share is growing.

Owner	Records
Universities and research organisations	37
Companies	31
Communities	16
Central government	8
Civil society organisations	5
Regions	2
International organisations	1

By country a third of the records are international (34). Then come the US (19), China (12), Russia (6), Japan and Korea (5 each), Germany and India (4 each). China has its own ecosystem of hubs and competition platforms: Alibaba's ModelScope, OpenDataLab and OpenXLab of the Shanghai AI Laboratory, the BAAI platform, Tianchi, Baidu AI Studio, HeyWhale and WiseModel. The Russian records are datasets from Yandex and from the computer graphics lab of Moscow State University's CMC faculty, the Moscow medical datasets MosMedData, the datasets section of the Moscow AI portal, Medical Data Hub and the CIT registry of the Ministry of Industry and Trade.

There are 78 custom-built websites among ML catalogs, and another 9 are made with the Jekyll static site generator on GitHub Pages; that is more than half of the single-benchmark websites. A programming interface is marked for only 23 records. Ready-made platforms are found mainly for competitions: CodaLab, Codabench, EvalAI and Grand Challenge.

What they publish

- **Training and test datasets**: texts, images, speech, video, tables, graphs, molecules. Large text datasets for language models take terabytes, a tabular dataset from the UCI archive kilobytes.
- **Models**: files with the weights of a trained network, with a model card: what it was trained on, what it is good for, where it makes mistakes. Hugging Face has three times as many models as datasets.
- **Tasks and results**: competition terms, the metric used to compare models, and the results table.
- **Derived datasets**: subsets, translations, re-annotations of other datasets and synthetic data generated by another model. This is now probably the fastest-growing part, and it is the one with the worst provenance.

A library catalog is a special case. TensorFlow Datasets or `torchvision` store not data but loader code: the address of the source file, a checksum and the split rules. The file still sits with the dataset's author or on a mirror, and when the original website disappears, loading breaks for all users of the library. An example is in the section Where it breaks.

Standards

Dataset card. This is text for a person: what is in the dataset, how and why it was collected, who annotated it, what biases are known, what the dataset should not be used for. The idea was proposed in the paper "Datasheets for Datasets" (Gebru et al., 2018); similar templates are Google's Data Cards and model cards (2019). On Hugging Face the card is a `README.md` file in the dataset repository. At the top of the file is a metadata section: licence, languages, tasks, size, and which files belong to which split. Search filters work on these fields.

Croissant. A machine-readable description of an ML dataset maintained by the MLCommons consortium. It is built on top of schema.org/Dataset and adds what a program needs to load the data: which files make up the dataset, what fields and types they have, how the dataset is split, where the labels are. A Croissant file is also ordinary schema.org markup, so search engines read it too: Google Dataset Search indexes datasets from it and can show only datasets with Croissant.

- Version 1.0 came out in 2024, version 1.1 on 29 January 2026. Version 1.1 adds data provenance using the W3C PROV-O vocabulary (from what and by what process a dataset or an individual field was produced), usage terms using the DUO and ODRL vocabularies (for example, "research only, not commercial"), rules for dataset versions and file checksums, and a flag for datasets that are continuously appended.
- The RAI extension (Responsible AI, version 1.0 of March 2024) describes a dataset's limitations, biases, sensitive information, intended use, whether the data is synthetic and which datasets it was made from.
- GeoCroissant 1.0 (January 2026) is an extension for geodata: coordinate reference system, coverage, imagery.

Implementations lag behind the specification. On 12 October 2026 Hugging Face, Kaggle, OpenML, Harvard Dataverse and the ckanext-dcat extension served descriptions marked as version 1.0. If you need the provenance and usage terms fields from 1.1, fill them in yourself and check that the platform does not lose them.

Conference requirements. NeurIPS, the main scientific conference on machine learning, requires a Croissant file in its data track. In 2026 the track is called Evaluations & Datasets, and the file is needed for every submission that introduces a new dataset. In the same year minimal RAI fields were added to the mandatory Croissant fields. The preferred platforms under the track's rules are Kaggle, OpenML, Hugging Face and Dataverse: they generate the main Croissant fields themselves, and the author adds the RAI fields. If the dataset is on the author's own website, the author makes the whole Croissant file. For a catalog this means that Croissant support is no longer a nicety but a condition for a paper with a new dataset to be accepted in this track.

Formats. Tables on hubs are increasingly stored in Parquet: it has column types, and programs read it in parts without downloading it whole. Hugging Face converts datasets to Parquet itself, but no more than 5 GB: if the dataset is larger, the converted copy will be incomplete. Hugging Face generates a Croissant description only for datasets it managed to convert, so an archive of images in ZIP will remain without one. OpenML has historically stored tables in ARFF, the format of the Weka program, and serves Parquet alongside. Formats in general are covered in the chapter on formats and units of publication.

Licences. Besides the usual Creative Commons and free software licences, there are licences with use restrictions here, for example OpenRAIL (bans on certain uses of a model), "research only" notes and competition rules, which often allow the data to be used only within the competition. A dataset's licence does not override the rights to the source material: an image dataset under CC BY does not make the images themselves free if they were collected from the internet.

Disclosure of training data. Since 2 August 2025 developers of general-purpose models in the EU have been obliged by the AI Act (Article 53) to publish a summary of the data on which a model was trained. The European Commission published the template on 24 July 2025, and its use is mandatory; for models released before 2 August 2025 the deadline is 2 August 2027. The summary lists the main datasets and other sources. For such a summary to be compiled and checked, a dataset needs a persistent name, an address, a version and a licence, which is exactly what a good card in a catalog provides.

DOI. In ML catalogs a DOI is an exception, not a rule. Hugging Face issues a DOI via DataCite at the click of a button in the dataset settings; after that the dataset cannot be deleted, renamed or hidden without contacting support. As of 12 October 2026 Hugging Face had 5,433 dataset DOIs in DataCite, about half a percent of its datasets. Kaggle has issued 66,208 dataset DOIs, the UCI archive 693. How DOIs work is covered in the chapter on identifiers.

Platforms

There are two different choices here: publish a dataset on someone else's platform, or set up an ML catalog of your own.

Other people's platforms

Platform	What it is	Loading from code	Croissant	DOI	Limitations
Hugging Face Hub	cloud hub for datasets and models	the `datasets` library, API, `hf://` addresses in DuckDB, pandas and Polars	generated automatically for datasets converted to Parquet	at the click of a button, via DataCite	gated access is enabled by the author; free space for public datasets is "best effort", 100 GB for private ones
Kaggle	cloud platform for datasets, models and competitions	`kagglehub`, Kaggle API	on the dataset page and at `/croissant/download`	issues them, prefix 10.34740	up to 200 GB per dataset; files are downloaded after login
OpenML	open-source platform for datasets, tasks and experiments (BSD-3)	the `openml` library, API	at `/croissant/dataset/` plus the dataset number	not found in the documentation	mostly tables; a task fixes the split and the metric
Codabench	open-source competition and benchmark platform (Apache-2.0)	—	no, it is not a dataset catalog	no	the test part can be hidden from participants
Zenodo	generalist research repository	API	no	for each version	up to 50 GB per record free

Codabench is the successor of CodaLab Competitions. CodaLab stopped accepting new competitions in September 2025 and closed submissions on 31 December 2025. Tasks are moved to Codabench unchanged, but results tables are not: submissions have to be made again.

Platforms provide what your own catalog will not have for a long time: an audience, loading from code, table previews, Croissant and download counts. The price is dependence on a single company. What this risks and how to hedge is covered in the chapter on cloud storage and hubs.

Your own catalog: what is in the core and what is not

An installation of your own is usually built on a research repository or open data portal platform with Croissant added. There is no dedicated open-source ML catalog platform that can be installed as easily as CKAN or Dataverse. OpenML can be deployed on your own servers, but it is a complex

system for experiments, not just a catalog.

Capability	Hugging Face	Kaggle	OpenML	Dataverse	CKAN
Croissant	core	core	core	core since 6.10, an external exporter before that	the ckanext-dcat extension, `croissant` plugin
Splits	core: splits are described in the card	no: only the dataset's files	core: a task defines the split	no	no
One-line loading from code	core	core	core	no: API and the pyDataverse client	no: API and direct file links
Gated access	core: the author enables access requests	no: only an entire private dataset	not found	core: restricted files with access requests	only an entire private dataset
DOI	core	core	not found	core	the ckanext-doi extension

Details for your own installations:

- **CKAN.** The `croissant` plugin in the ckanext-dcat extension appeared in version 2.3.0 (February 2025); the latest release of the extension is 2.4.4 of 11 August 2026. The description is embedded into the dataset page and served at `/dataset/<id>/croissant.jsonld`. Table fields (`RecordSet` in Croissant terms) get into the description only if the file is loaded into the CKAN DataStore table store. Otherwise a program will learn about the file but not about its columns. On the extension itself and its compatibility with CKAN versions, see the chapter on CKAN.
- **Dataverse.** Since version 6.10 Croissant is built into the core: the dataset page embeds a slim version without the file list, and the full description comes from the export. Before that it was an external exporter from the GDCC community (latest release 0.1.6), which worked with Dataverse 6.0 and later and embedded the description into the page from version 6.2; its repository is now marked as deprecated. Installations older than 6.10 still add Croissant with that exporter. On 12 October 2026 Harvard Dataverse served Croissant via an API request.
- **GeoNetwork** embeds schema.org markup into the record page. Croissant fields for the ISO 19115-3 schema have been added to the branch of the upcoming 4.4.13 release; release 4.4.12 does not have them. This is useful if the training datasets are imagery and layers.
- **InvenioRDM and Zenodo** do not serve Croissant: the platform code has schema.org, Data Package and RO-Crate markup, but not Croissant.
- **A static site.** Nine benchmark websites in the registry are made with Jekyll and hosted on GitHub Pages. For a collection of several datasets this is enough, provided each dataset has a Croissant file checked with the validator next to it, and the files themselves are on a hub or in a repository with a DOI. How to make such a website is covered in the chapter on static sites.

Government AI data platforms

Governments increasingly build their own ML catalogs: they collect datasets in national languages, on health, transport and industry, and hand them to their own developers. They are organised differently from hubs and are closer to microdata catalogs: what matters most is who has the right to get the data.

- **AI Hub, Korea.** Run by the National Information Society Agency (NIA) on behalf of the Ministry of Science and ICT. The dataset pages state plainly that only Korean citizens can apply for data. A foreign organisation needs a separate contract with the contractor and NIA, and exporting data abroad also requires a contract. Medical data is opened in a "safe zone": on NIA's server without downloading, and a trained model is released after review.
- **AIKosh, India.** In 2026, more than 12,000 datasets and models. A dataset has a quality score and terms of use, downloading requires registration, and the licence usually prohibits commercial use. Alongside is Bhashini Vatika, a platform for language datasets.
- **US and EU.** The pilot of the National AI Research Resource (NAIRR) provides a list of partners' datasets, models and computing resources. The European AI-on-Demand catalog collects datasets, models and tools from EU projects.
- **Russia.** The registry of industry machine learning models and datasets is run by the federal autonomous institution CIT under the Ministry of Industry and Trade. In December 2025 it had 42 datasets and 7 models; on 12 October 2026, 141 datasets. Among them are copies of other people's datasets from Hugging Face and Kaggle with a link to the source in the description, and the licence on the card is written as the words "Free licence" without a name. Moscow publishes a datasets section on the city's AI portal and the MosMedData medical imaging datasets.

The conclusion for such a platform: if the data is closed, at least the card should be open, with the licence, provenance and access procedure, in Croissant and schema.org. Then the dataset can be found from outside, and a researcher understands whether it is worth applying. How access on request and a secure environment work is covered in the chapter on microdata catalogs and in the chapter on access and workflows. The trend as a whole is in the chapter on AI and data spaces.

Examples

Hubs

- Hugging Face Datasets: datasets and models in one place, loading from code, Croissant and DOIs.
- Kaggle Datasets: datasets, notebooks and competitions.
- ModelScope: Alibaba's Chinese hub of datasets and models.

Collections and archives

- UC Irvine Machine Learning Repository: one of the oldest archives of ML datasets, with an API and DOIs for datasets.
- OpenML: datasets together with tasks, splits and experiment results.

Competitions and benchmarks

- Codabench: competitions and benchmarks with a hidden test part.
- Open Graph Benchmark: a single-benchmark website on GitHub Pages: datasets, a loader and a results table.

Library catalogs

- TensorFlow Datasets: a list of datasets that the TensorFlow library loads.

Government platforms

- AI Hub: Korea.
- AIKosh: India.
- CIT registry of industry models and datasets: Russia.

Where it breaks

The licence is unclear or wrong. The Data Provenance Initiative study (Longpre et al., 2023; journal version in Nature Machine Intelligence, 2024) traced the provenance of more than 1,800 text datasets for fine-tuning language models. For popular datasets on GitHub and Hugging Face, more than 70% of licences were not specified at all. Where there was a licence, for 66% of datasets on Hugging Face it belonged to a different category than the licence of the original data author, and was usually more permissive. A dataset is moved from platform to platform, and restrictions are lost along the way.

The card is empty. According to a study of Hugging Face cards (ICLR 2024, data as of March 2023), of 24,065 dataset repositories only 30.9% had a non-empty card. But these accounted for 95% of downloads. All recommended sections were filled in for 86% of the hundred most downloaded datasets and only for 7.9% of datasets without downloads. The section on the risks of using the data took up only 2.1% of the text in the most popular datasets. The hub shows such datasets in search on an equal footing with well-described ones.

A dataset is withdrawn, but copies live on. On 19 December 2023 the Stanford Internet Observatory found links to child sexual abuse images in the LAION-5B dataset. LAION took the dataset and its derivatives down the same day. On 30 August 2024 a corrected version, Re-LAION-5B, was released with 2,236 links removed. Models trained on the old version and copies of the old dataset on other platforms remained. A good catalog in such a case does not delete the dataset page but keeps it with a "withdrawn" note, the reason and a link to the replacement. Then whoever arrives from a link in a paper will learn what happened.

The library points to a dead website. The classic MNIST handwritten digits dataset was downloaded by the `torchvision` library from the personal page of one of the dataset's authors, Yann LeCun. The page started responding with a 403 refusal, and loading broke for users. The library developers got permission to keep a copy of the dataset themselves and later put their own mirror first in the list. What saved the day was that each file has a checksum recorded in the code: it shows that the file from the mirror is the same. Croissant 1.1 strongly recommends recording such checksums for all files of a versioned dataset.

The test part leaked into training. Language models are trained on texts from the internet, and published test sets get into training together with the answers. Then a model's benchmark result is inflated, and the comparison loses its meaning. There are two defences. The first is not to publish the test answers: in the GLUE dataset on Hugging Face, all 1,063 rows of the test part of the CoLA task have the label `-1`, and the score is calculated by the benchmark server. Hidden results tables on Kaggle and Codabench work the same way. The second is a "canary" string: a special string is written into every task file of the BIG-bench benchmark so that compilers of training datasets can find and exclude such files.

Files only after login. Kaggle itself marks the file as requiring login in the dataset page markup (`requiresSubscription: true`). On Hugging Face the author can enable access requests, at AI Hub only Korean citizens get the data, and at AIKosh only after registration. For a person this is an inconvenience; for a dataset search engine or harvester it is a dataset without files. If a restriction is necessary, the card should still be open, with the reason and access procedure written in it.

The platform closed. For years Papers with Code linked papers, code, datasets and results tables. On 24 July 2025 Meta shut the service down, and the address of the datasets section now leads to the Hugging Face trending papers page, which has neither datasets nor results tables. The last data snapshot was saved as an archive, but links from papers to dataset pages stopped working. The catalog registry does not learn about this immediately: in release 1.27.0 the Papers with Code record is still marked as active and with an API. Likewise, when CodaLab closed, results tables were not moved to Codabench. The conclusion for a dataset author: a card on a hub is a way to find the dataset, not a place for its long-term preservation. A copy of the release with a DOI is kept in a research repository.

The page is empty without a script. On 12 October 2026 the datasets section of the Moscow AI portal served a robot an empty page shell of about a kilobyte, while the list of datasets was drawn by a script. A search engine, a web archive or an AI assistant will not see such a list. How to fix this is covered in the chapter on discoverability.

A custom website without an API. Of the 78 custom ML catalogs in the registry, 13 have a programming interface marked. For an ML catalog this is especially strange: its users work from code and hardly use a browser.

How to check an ML catalog and a dataset

Six questions for a specific dataset:

1. **Is there a machine-readable description?** Request Croissant or look for schema.org markup on the dataset page.
2. **How is the dataset split, and where are the test answers?** If the answers are published, results on it may be inflated.
3. **What is the licence, and where does the data come from?** The licence must be named, and for a derived dataset the source dataset and its licence must be stated.
4. **Can it be downloaded without login and without a browser?** Try loading the dataset from code or by a direct link.

5. **Is there a version and a persistent identifier?** A DOI, or at least a version number and file checksums.
6. **Is there a copy outside the platform?** If the dataset is only on one hub, the hub closing is equivalent to deletion.

```
# Hugging Face: Croissant description and the list of Parquet files by split
curl -s "https://huggingface.co/api/datasets/nyu-mll/glue/croissant"
curl -s "https://datasets-server.huggingface.co/parquet?dataset=nyu-mll/glue"
# Read part of a dataset straight from the hub, without downloading an archive (DuckDB
0.10.3 and later):
# in the CoLA test part all labels are -1, the answers are hidden
duckdb -c "SELECT label, count(*) FROM 'hf://datasets/nyu-mll/glue/cola/test-*.parquet'
GROUP BY 1"
# Kaggle: a dataset's Croissant (the response redirects to a file in cloud storage)
curl -sL "https://www.kaggle.com/datasets/unsdsn/world-happiness/croissant/download"
# OpenML: Croissant and the dataset description with ARFF and Parquet file addresses
curl -s "https://www.openml.org/croissant/dataset/61"
curl -s "https://www.openml.org/api/v1/json/data/61"
# UCI: dataset description, the response includes a DOI
curl -s "https://archive.ics.uci.edu/api/dataset?id=53"
# Dataverse: the full Croissant description
curl -s "https://dataverse.harvard.edu/api/datasets/export?
exporter=croissant&persistentId=doi:10.7910/DVN/TJCLKP"
# DataCite: how many dataset DOIs Kaggle has issued (prefix 10.34740); Hugging Face's prefix
is 10.57967
curl -s "https://api.datacite.org/dois?prefix=10.34740&resource-type-
id=dataset&page%5Bsize%5D=0"
# Validate your own Croissant file before publishing
pip install mlcroissant
mlcroissant validate --jsonld metadata.json
```

All commands were checked on 12 October 2026. In the responses from Hugging Face, Kaggle, OpenML and Dataverse, the `conformsTo` field specifies Croissant 1.0. The general list of checks is in the checklists.

How to choose

Situation	What to choose	Why
A dataset for a paper, or a research group's benchmark	Hugging Face or Kaggle, a copy of the release with a DOI in Zenodo or a research repository	that is where datasets are searched for and loaded from code; conferences accept these platforms, and a copy with a DOI will outlive the platform
A competition with a hidden test part	Codabench (the shared codabench.org or your own installation) or Kaggle competitions	hidden answers and a results table are in the core; your own website will not provide this
The data cannot be given to someone else's platform: health, personal data	your own repository with access on request, for example Dataverse (Croissant in the core since 6.10), or a secure environment	restricted files with access requests are in the Dataverse core; an open card in Croissant makes the dataset findable
An open data portal wants its datasets to be used for training	do not build a separate ML catalog; enable the `croissant` plugin in CKAN, serve tables in Parquet with a field dictionary	developers will find the dataset through a dataset search engine and load it without manual work

Situation	What to choose	Why
A lab's collection of a dozen datasets	a static site with a Croissant file for each dataset, files on a hub or in a repository with a DOI	cheap and durable; most benchmark websites are built this way
A government AI data programme	a catalog with open cards and access on request, Croissant, description export and an API	closed data stays under control, while descriptions are visible from outside and reach search engines

When you do not need your own ML catalog

Almost always, if the goal is for the dataset to be found and used. The audience, loading from code and Croissant already exist on the hubs, while your own website will have to be maintained for years for the sake of a few datasets. A careful card on a hub, a named licence, splits and a copy of the release with a DOI are enough.

A catalog of your own is justified in three cases: the data cannot be put on someone else's platform; the hub does not accept your type of files or volume; you are a government or industry programme that must keep track of and distribute datasets from many organisations. Even then the card is worth duplicating where people will look for the dataset. How to justify such a decision is worked through in the chapter "Build, Open Source or SaaS".

How it differs from its neighbours

Neighbour	Main difference
Research repository	the main thing is an immutable version with a DOI and a citation from a paper; in an ML catalog, loading into code and comparing models. A copy of an ML dataset release is often put into Zenodo precisely for the DOI and preservation
Open data portal	the dataset is not split, the description has no section on biases; a public body publishes, not the dataset author
Microdata catalog	survey data about people with access on request. Government AI data platforms with "safe zones" are organised similarly
Marketplace	datasets are sold; training data vendors such as Datatang are marketplaces rather than ML catalogs
Git repository	many benchmarks live in a GitHub repository with a website on GitHub Pages; this is publishing without a portal, covered in the chapter on Git repositories
A company's internal model registry	tracking your own models for development and release, for example in MLflow; this is an enterprise tool, not a public catalog

A summary table of all types is in the chapter Comparing Types and Hybrids.

Sources

- dataportals-registry export, release 1.27.0 of 15 October 2026: number of ML catalogs, owners, countries, platforms, share with an API and statuses. The classification of records by kind was made by hand for the book from the list of records. The queries are in the appendix.
- Number of datasets and models: Hugging Face Datasets and models, the OpenML API, the UCI archive API; 12 October 2026

- Croissant: website and list of supporting platforms, specification 1.1 of 29 January 2026, the RAI 1.0 extension, GeoCroissant 1.0, the mlcroissant library and validator
- Cards: T. Gebru et al., Datasheets for Datasets, 2018; M. Mitchell et al., Model Cards for Model Reporting, 2019; dataset cards on Hugging Face
- NeurIPS 2026: Evaluations & Datasets track rules, dataset hosting and RAI field requirements, 4 May 2026
- Hugging Face: gated access, DOI, storage limits, Parquet conversion and its 5 GB limit, Croissant for datasets, DuckDB and hf:// addresses
- Kaggle: dataset documentation, with the 200 GB per dataset limit; Croissant and dataset page markup checked 12 October 2026
- Own installations: Croissant in ckanext-dcat and extension releases; Croissant in Dataverse and the former external exporter; InvenioRDM export formats in the source code, without Croissant; OpenML, Codabench
- CodaLab and Codabench: announcement of CodaLab's closure, moving to Codabench, 2025 summary
- DOIs of ML platforms: the DataCite API, prefixes 10.34740 (Kaggle), 10.57967 (Hugging Face), 10.24432 (UCI), 12 October 2026
- The EU AI Act, Regulation 2024/1689; the training content summary template, 24 July 2025, and FAQ about it
- Government platforms: AI Hub terms of use and the safe zone; AIKosh; NAIRR Pilot; AI-on-Demand; the CIT registry and a report on its update (in Russian), December 2025; number of datasets, licences on CIT registry cards and the HTML of the ai.mos.ru datasets section checked 12 October 2026
- S. Longpre et al., A large-scale audit of dataset licensing and attribution in AI, Nature Machine Intelligence, 2024; 2023 preprint
- X. Yang et al., Navigating Dataset Documentations in AI: A Large-Scale Analysis of Dataset Cards on Hugging Face, ICLR 2024
- LAION: LAION-5B taken down, 19 December 2023; Re-LAION-5B release, 30 August 2024
- MNIST in torchvision: loader and mirrors, switch to own mirror, 403 error on the original website
- BIG-bench: the "canary" string in task files; the GLUE dataset on Hugging Face, CoLA test part checked 12 October 2026
- Closure of Papers with Code: discussion of the redirect to Hugging Face; the address paperswithcode.com/datasets leads to huggingface.co/papers/trending, checked 12 October 2026
- I. Begtin, Telegram (in Russian): "code first" in ML catalogs, 21 March 2024; AIKosh, 26 August 2026; national catalogs for AI, 30 September 2026
- Cloud Storage and Shared Hubs, AI, Data Spaces and Regulation

Metadata Catalogs and Registries

A metadata catalog describes not the data itself but what data is described with and where to look for it: vocabularies of terms, code lists, field definitions, structures of statistical tables, descriptions of datasets held elsewhere, and lists of other catalogs. People come to such a site not to download a table but to find out the code for the theme "Transport", what the field "author speciality" means, or which repositories accept climate data.

According to the dataportals-registry export (release 1.27.0 of 15 October 2026), there are 172 such catalogs, 0.4% of all catalogs. The type is small, but all records are active, and 67% have programmatic access: much more than indicator catalogs (37%) and almost as many as open data portals (69%).

The main conclusion of the chapter: the value of a metadata catalog lies in others referring to it. That is why every record needs a persistent address, a version and a note of what replaced it. Descriptions should be served to programs in a standard (SKOS, SDMX, DCAT), and the whole database should be downloadable. A registry that is only read by eye stops being cited after a few years and turns into one more website with a list.

How a record is organised

On an open data portal the main thing is a file; in an indicator catalog, a number with breakdowns. In a metadata catalog the unit is a description that others refer to: a term, a code, a data element, a description of a dataset or a catalog. Three live examples show how this works.

A term as an address. In the EU Publications Office vocabularies the theme "Transport" has the address `http://publications.europa.eu/resource/authority/data-theme/TRAN`. From it a program gets a description of the theme in RDF with labels in the EU languages (checked 12 October 2026). Portals following the DCAT-AP profile put this address in the theme field, not the word "Transport". That is why data.europa.eu brings together transport datasets from different countries without translating or guessing words. Any good vocabulary works this way: a term is an address, and a name is one of its labels.

A code from a shared code list. In the key of the European Central Bank's dollar exchange rate series `EXR.D.USD.EUR.SP00.A`, the letter `D` is not the bank's invention. It is the code for "daily" from the frequency code list `CL_FREQ`, which sits in the global SDMX registry. As of 12 October 2026 this code list, version 2.1, has 34 codes: `A` for annual, `Q` quarterly, `M` monthly, `W` weekly, `D` daily and others. The code list is the same for all statistical offices, so `M` means "monthly" at both Eurostat and the IMF. How a series key is organised is covered in the chapter on indicator catalogs.

A data element with a passport. The Swiss platform I14Y describes the concepts, code lists, datasets and services of public bodies. The first concept in its open API on 12 October 2026 is the field `DocumentEntry.author.authorSpeciality` from eHealth Suisse: the speciality of the author of a medical document. It has a description in four languages, a type (code list), a value type

(string), a maximum length (30 characters), a publisher, a version, a registration status, a validity start date and the fields "replaces" and "replaced by". This is exactly the kind of field passport that the ISO/IEC 11179 standard describes; more on it below.

What gets its own page

Depending on the kind of catalog, the following get their own address:

1. **A concept or code**: one theme, one heading, one code-list code.
2. **A whole vocabulary or code list**: a thesaurus, classification, ontology, code list.
3. **A data element**: the definition of a field with permitted values.
4. **A structure**: the description of a table: which breakdowns it has and which code lists are attached to them.
5. **A description of a dataset without the data**: what an organisation has and how to get it.
6. **A description of a catalog, standard or infrastructure**: a card for a repository, vocabulary, network of observing stations.

That is why the number of records in a metadata catalog cannot be compared with the number of datasets on a portal. On 12 October 2026 the global SDMX registry had 451 code lists, 248 table structures and 1,167 dataflow descriptions including all versions, and not a single statistical value.

Who builds them

Metadata catalogs are split almost equally between science, international organisations and central government.

Owner	Catalogs	Share
Universities and research organisations	54	31%
International organisations	52	30%
Central government	41	24%
Regions	8	5%
Civil society organisations	6	3%
Companies	6	3%
Communities and others	5	3%

- **Research organisations and libraries** run vocabularies and ontologies: the National Library of Finland runs the Finto vocabulary service, EMBL-EBI the life sciences ontology search OLS, Stanford BioPortal. Medical centres and rare disease research networks set up FAIR Data Points to describe their cohorts and patient registries.
- **International organisations** keep shared code lists and structures: FAO the AGROVOC thesaurus, UNESCO its own thesaurus, the Bank for International Settlements the global SDMX registry, the World Meteorological Organization the OSCAR registries of observing stations and satellites, the secretariat of the Convention on Biological Diversity its clearing-house mechanisms.

- **Central government** runs registries of data elements and code lists: the Australian Institute of Health and Welfare runs METEOR, the New South Wales government a metadata registry, Switzerland I14Y, the Dutch agency Logius the Stelselcatalogus, a catalog of the concepts of the base government registers, and the US Library of Congress the linked data service id.loc.gov.

By country the leaders are the Netherlands (24), the United Kingdom (20), Spain (9), the US and Germany (7 each), Australia (6); another 40 catalogs cover the whole world and 11 the European Union. As with other types, the number shows which software has taken root in a single community, not how mature the country is. In the Netherlands 19 of 24 are FAIR Data Points of medical centres and research networks. In the UK 15 of 20 are catalogs on the Metadata Browser cloud service. In Spain 7 of 9 are also FAIR Data Points, six of them at seed banks of botanical gardens and universities. In Australia 4 of 6 are registries on Aristotle Metadata Registry.

The boundary of the type is blurred, and 172 is a lower estimate. The registry does not include re3data itself, Research Vocabularies Australia is recorded as a research repository, and the Irish catalog of protected data as an open data portal on CKAN. On the other hand, catalogs of other objects have ended up here: of software packages (Libraries.io, Google Open Source Insights), journals (the ISSN portal), commodity codes (WTO HS Tracker). They also describe rather than store, but relate to public data only indirectly.

What they publish

- **Vocabularies, thesauri, classifications and ontologies.** Themes, subject headings, country and language codes, economic activities, scientific terms. Other catalogs use them to fill in fields the same way. An ontology differs from a vocabulary in its rigour: besides "broader–narrower" relations it defines classes, properties and rules from which a program draws inferences.
- **Structures and data elements.** Which breakdowns an indicator has and which codes are permitted in each. What the field "registration date" means, in what format it is written, who is responsible for the definition and which revision is currently in force.
- **Schemas.** Description templates: which fields a record about a dataset, sample or object must have. This is how the CEDAR schema registry and the buildingSMART Data Dictionary for construction work.
- **Descriptions of datasets without the data itself.** An organisation says what it has, and the data stays with it: medical cohorts, patient registries, biobank collections, protected public sector data.
- **Lists of catalogs, standards and infrastructures.** A card for a repository, vocabulary or standard: address, subject, terms, certifications, interfaces.

What a metadata catalog usually lacks: the values, files and records themselves. If a website lets you download a table, it is already a portal or an indicator catalog, even if the table is a code list. This boundary is covered in the section on code list portals.

Standards

SKOS (W3C, 2009) is a way to describe a vocabulary in RDF. Each concept has an address, a preferred label and synonyms in different languages, a code, "broader–narrower" and "related" relations, and mappings to concepts in other vocabularies. Almost all open vocabularies are served in SKOS. For statistical classifications there is the **XKOS** extension (DDI Alliance, version 1.0 of 2019): classification levels, correspondence tables between revisions, "includes" and "excludes" notes.

OWL is the W3C ontology language. It is used to describe domains of knowledge with classes and rules: diseases, genes, plant species, crops. Ontology catalogs (OLS, BioPortal, AgroPortal) accept ontologies in OWL and show them to people and programs.

ISO/IEC 11179 "Metadata registries" describes how to define a single data element. An element consists of a concept (an object and a property: "person – sex") and a representation (permitted values: a code from a list, a date, a number with a unit). An element has a definition, a steward responsible for it, and a registration status, from candidate to standard, and then "superseded" or "retired". The current edition of the metamodel is ISO/IEC 11179-3:2023, and in 2026 an amendment on mapping elements was published. The standard is used where different systems must understand a field in the same way: in health care, official statistics and exchange between agencies.

SDMX is a model of statistical structures: code lists, concept schemes, table structures, dataflows. An SDMX registry stores these structures so that all exchange participants take them from one place. The standard is covered in detail in the chapter on indicator catalogs.

DCAT and its profiles. A FAIR Data Point serves descriptions of catalogs and datasets in DCAT, so they are taken by the same harvesters as open data portals. For health data the European Commission is developing the **HealthDCAT-AP** profile: Release 7 was published in May 2026 as a draft specification. It adds to DCAT-AP the fields needed for health data: the health data access body, the category of health data, the number of records and the number of unique individuals, age coverage, retention period, purpose of processing and legal basis.

Describing repositories. re3data has its own schema (version 4.0): subjects, data types, access conditions for the data and for the repository itself, certifications, interfaces. The questions about access in the chapter on classification follow the same schema.

For pages read by search engines, schema.org has the types `DefinedTermSet` and `DefinedTerm`: a vocabulary and a term. This is not a replacement for SKOS but a way to make a vocabulary findable.

Description catalogs required by law

In the European Union, catalogs describing closed data are turning from a voluntary matter into an obligation.

- **The Data Governance Act** (Regulation 2022/868, Article 8) obliges each EU country to set up a single information point. It must contain a searchable list of protected public sector data (personal, commercially confidential, statistically confidential) with the format, size and

conditions for reuse. Access requests are submitted through the point. The European Commission collects the national lists into the European Register for Protected Data held by the Public Sector (ERPD) on data.europa.eu.

- **The European Health Data Space Regulation** (EHDS, Regulation 2025/327, Article 77) requires health data access bodies to keep a public, machine-readable catalog of descriptions of datasets available for secondary use: source, coverage, main characteristics and access conditions. The catalog must be available to the single information point under the Data Governance Act. The rules on secondary use apply from 26 March 2029 for most data categories, and from 26 March 2031 for five categories, including genomic data. A 2026 draft implementing act of the European Commission proposes describing datasets according to HealthDCAT-AP.

For a developer this means: a catalog describing closed data must serve DCAT-AP or its sector profile and be harvestable, otherwise it will not reach the national point and the European register.

Platforms

Custom builds make up 38% (65 catalogs), less than for indicator catalogs (44%). Only 17% of custom metadata catalogs have programmatic access, while almost all specialised platforms do.

Platform	Catalogs in the registry	What it is	Programmatic access
Custom build	65	catalog registries, vocabularies, agency code lists	for 17%
FAIR Data Point	47	open-source (MIT) point for an organisation's descriptions in DCAT; release 1.22.2 of 24 September 2026	for all: DCAT in Turtle, API description in OpenAPI
Metadata Browser	16	commercial cloud service from the British company MetadataWorks for catalogs of health and administrative data	for all
Skosmos	10	open-source (MIT) software from the National Library of Finland for publishing SKOS vocabularies; release 3.4 of 31 August 2026	REST API in the core
Aristotle Metadata Registry	8	a registry of data elements following ISO/IEC 11179; cloud service	API for all
Fusion Metadata Registry	7	the Bank for International Settlements' SDMX structure registry; free application, release 12.4.2 of 19 September 2026	SDMX REST API in the core
MOLGENIS	5	open-source (LGPL-3.0) platform from Groningen; catalogs of biobanks and cohorts	GraphQL for all
OntoPortal	2	open-source (BSD) platform for ontology catalogs; BioPortal and AgroPortal run on it	REST API
Sector systems	7	WMO OSCAR, the Convention on Biological Diversity's clearing-house mechanisms, SeaDataNet vocabularies	for most
Other	5	one each: DataHub, EntryScape, PublishMyData, CEDAR, Drupal	—

The type divides into three worlds. The first is narrow open-source tools, each for one kind of description: Skosmos and OntoPortal for vocabularies and ontologies, Fusion Metadata Registry for SDMX structures, FAIR Data Point for dataset descriptions in DCAT. They have an API and a standard format in the core. The second is cloud registries: Aristotle for data elements and Metadata Browser for catalogs of sensitive data. Convenient, but the descriptions live with the vendor. The third is agencies' and projects' own developments, where machine access is the exception rather than the rule.

The registry records some well-known installations as custom builds, although reusable software stands behind them. For example, AGROVOC responds through the standard Skosmos REST API, and the ontology search OLS is open-source software from EMBL-EBI (Apache 2.0, release 4.0.3 of 8 October 2026) that can be installed on your own servers.

The fate of Aristotle's open code

Aristotle Metadata Registry began as open-source software on Django built according to ISO/IEC 11179. In August 2020 the code was removed from the public repository: only a file remained announcing that the product was being reworked and offered as a cloud service. Those who had installed the open version were left with code without updates. All eight Aristotle registries in the registry export are cloud-hosted. Before choosing a platform for a registry, check who maintains it and when the last release came out, just as with extensions.

Core and add-ons

Metadata catalog platforms have no extension ecosystem like CKAN's. Each does one job, and whatever is missing is added with a neighbouring product.

Capability	FAIR Data Point	Skosmos	Fusion Metadata Registry	Aristotle (cloud)
What it stores	descriptions of catalogs, datasets and distributions	SKOS vocabularies	SDMX structures: code lists, concepts, tables	data elements following ISO/IEC 11179
Machine format	DCAT in RDF (Turtle and others)	SKOS via REST API	SDMX-ML, SDMX-JSON	own API
Editing in the interface	core	no: the vocabulary is prepared in another editor and loaded into a SPARQL store	core	core
Versions and statuses	core	no, versions are kept in the vocabulary's source	core: each structure has a version	core: registration statuses
Data processing	no	no	validation and conversion of data between SDMX formats	no
Registration in a shared catalog	core: the point reports itself to the FAIR Data Point index	no	no	no
How it is deployed	container on your own server	web application on your own server on top of a SPARQL store	container on your own server	only at the vendor

If you need a catalog of catalogs rather than a vocabulary or a structure registry, special software is not required. The CKAN ecosystem catalog itself runs on CKAN, and the dataportals-registry is a set of YAML files in a Git repository with exports. For descriptions of closed public sector data an ordinary portal is often used too: the Irish catalog of protected data is built on CKAN.

How to choose

The choice begins with the question of what exactly you are describing and who will refer to your records.

Situation	What to take	Why, and what it costs
You need to publish your own vocabulary or classification	SKOS with persistent concept addresses; Skosmos as a showcase; for a small vocabulary, a SKOS file in Git and persistent addresses via w3id.org	everyone reads the standard format; Skosmos only displays the vocabulary, an editor is needed separately
A sector maintains many ontologies	your own OntoPortal instance, like AgroPortal, or registering ontologies in BioPortal and OLS	ready-made search and mappings between ontologies; your own instance has to be maintained
A statistical office needs shared code lists and structures	Fusion Metadata Registry or the registry within .Stat Suite	SDMX in the core, validation of data against structures; implementation requires SDMX knowledge
Agencies must understand fields in the same way	a data element registry following ISO/IEC 11179: cloud Aristotle or a national platform such as I14Y	a field passport with statuses and versions; a cloud service keeps descriptions with the vendor
An organisation wants to describe its closed data	FAIR Data Point or a portal with a DCAT-AP export; for health data in the EU, with HealthDCAT-AP fields	descriptions are harvested by the national point and the European register; the data itself stays with the owner
A list of catalogs on a topic or country is needed	first check re3data, FAIRsharing, BARTOC and dataportals-registry; if your own list cannot be avoided, files in Git with CSV and JSON export	a list that can be downloaded will be checked and reused by others

A custom build is justified when the descriptions are more complex than ready-made products can handle and there is a team that will maintain them for years. Models are I14Y and id.loc.gov: both have an open API and persistent record addresses. If the team is not ready to provide machine access, its own registry will share the fate of custom metadata catalogs, of which programs can read only 17%. The general logic of choice is in the chapter "Build, Open Source or SaaS".

When you do not need your own catalog

Most organisations do not need a metadata catalog of their own. A vocabulary can be published as a SKOS file in a Git repository and registered in BARTOC or Linked Open Vocabularies. A repository can be entered in re3data, a standard or database in FAIRsharing, a catalog of any type in dataportals-registry. Descriptions of your own datasets can simply be served in DCAT from the portal where they already sit. A separate registry is needed by those whom many will refer to: a statistical office, a national library, a sector infrastructure, a body that the law requires to keep a list. Ways to publish without a portal are in the part "Publishing Without a Portal".

What makes a metadata catalog useful

Six signs that a catalog can be relied on.

1. **Persistent addresses.** Each concept, code and description has its own address that does not change when the website moves. The address of the theme `data-theme/TRAN` in the EU vocabularies is the same for all portals in Europe.
2. **Versions and replacements.** An obsolete record is not deleted but marked, and you can see what replaced it. In I14Y each concept has a version, a status, a validity start date and "replaces" and "replaced by" links.
3. **Machine access in a standard.** A record is served to a program in SKOS, SDMX or DCAT, not only as a page. A good sign is when the same address serves a page to a person and RDF or JSON to a program depending on the request header.
4. **Full download.** The whole vocabulary or the whole list can be downloaded as one file. re3data serves the list of all repositories in one request, the dataportals-registry in one file.
5. **A clear steward.** Each record names who is responsible for its definition and changes, and the catalog has rules for including records. In the Dutch Stelselcatalogus the holders of the registers themselves are responsible for the descriptions, and they have agreed on delivery deadlines and completeness.
6. **Registration one level up.** A vocabulary is listed in BARTOC, an ontology in OLS or BioPortal, a FAIR Data Point in the index of points, a catalog of closed data in the national single information point.

Examples

Vocabularies and ontologies

- EU Publications Office vocabularies: themes, file types, languages, countries, access rights and other code lists on which DCAT-AP relies. Datasets and services of the Swiss I14Y refer to them too.
- id.loc.gov: the linked data service of the US Library of Congress: subject headings, names, languages. Any heading is served in JSON-LD if you append `.json` to its address.
- AGROVOC: FAO's multilingual thesaurus on agriculture and food, published on Skosmos.
- Finto: Finland's national vocabulary service on Skosmos; the software itself came from its team.
- Ontology Lookup Service: EMBL-EBI's search across life science ontologies: 287 ontologies as of 12 October 2026.
- BioPortal and AgroPortal: catalogs of medical and agricultural ontologies on OntoPortal.
- Research Vocabularies Australia: Australia's shared catalog of research vocabularies.

Structures and data elements

- Global SDMX Registry: shared structures and code lists of international organisations on Fusion Metadata Registry. The registries of Eurostat, the IMF, UNICEF, the Gulf Cooperation Council statistical centre and Thailand's statistical office run on the same software.
- METEOR: Australia's registry of health and welfare data elements, run by the Australian Institute of Health and Welfare.
- I14Y: the Swiss interoperability platform: concepts, code lists, datasets, data services and public services with an open API.

- Stelselcatalogus: the catalog of concepts, data elements and information models of the Dutch base government registers: population, the trade register, the cadastre and others. Updated in June 2026; descriptions of datasets and services remained on data.overheid.nl and developer.overheid.nl.
- schema.gov.it: Italy's national catalog of ontologies, vocabularies and data schemas; more below.

Descriptions of datasets without the data

- FAIR Data Points of the Dutch Health-RI network, the Leiden and Nijmegen medical centres and a European rare disease programme.
- BBMRI-ERIC Directory: the European list of biobanks and their collections on MOLGENIS.
- Catalogs on Metadata Browser: the UK Office for National Statistics, the ADR UK programme for administrative data research, Genomics England, regional NHS trusted research environments.
- Ireland and the UK Data Service are covered in the section on open descriptions of closed data.

Catalogs of catalogs

A catalog of catalogs answers the question "where should I even look for data on this topic or country". It describes portals, repositories or vocabularies, not datasets. It is not a search engine: you will have to search on the portal you find. But it helps when a general search engine returned nothing because the catalog you need simply was not in it.

Catalog	Who runs it	What it describes	How many records
dataportals.org, formerly datacatalogs.org	Open Knowledge Foundation	open data portals; the list can be exported to CSV and JSON	631, October 2026
Open Data Inception	Huwise (formerly OpenDataSoft)	open data portals by country	"2600+" on the website, 3,763 records in the source dataset, October 2026
datacatalogs.ru	Infoculture	data catalogs and repositories in and about Russia: type, themes, software, API availability, access mode	245 at launch in November 2021, 263 in June 2022
re3data	DataCite and partners	research repositories	3,535, 12 October 2026
FAIRsharing	a University of Oxford group	standards, databases and policies of journals and funders	1,973 databases, 1,646 standards, 159 policies, January 2023
BARTOC	the VZG library centre, Germany	vocabularies, thesauri, classifications and ontologies	4,162, October 2026
Linked Open Vocabularies	Universidad Politécnica de Madrid	linked data vocabularies: DCAT, schema.org, FOAF and others	920, October 2026
FAIR Data Point index	the FAIR Data Point community	FAIR Data Points that have reported themselves	230, of which 74 active, October 2026
ecosystem.ckan.org	the CKAN community	extensions and portals on CKAN with core versions	about 1,400 extensions and 1,300 portals, October 2026

Catalog	Who runs it	What it describes	How many records
dataportals-registry	the Dateno project	catalogs of all types, one YAML file per catalog	more than 43,000, release 1.27.0, October 2026

The numbers in the table are not comparable: each list counts its own thing and decides in its own way what counts as a catalog. Why the number of catalogs in the world depends on the effort spent searching is covered in the chapter The World of Catalogs in Numbers. More on the CKAN ecosystem catalog is in the chapter on CKAN.

Open descriptions of closed data

A dataset description can be open even when the data itself is closed. Such a catalog says honestly: the data exists, this is what is in it, this is how to get access if you have grounds. Without it a person cannot request data simply because they do not know it exists. In the EU such catalogs are now required by law; see the section Description catalogs required by law.

- Ireland's Public Service Data Catalogue describes public bodies' data that cannot be published openly: personal data, trade secrets, statistically confidential data. Under the Data Governance Act this is "protected data": it can be obtained for reuse under conditions that preserve the protection. The catalog serves as a single point where agencies list such data and where anyone interested submits a request. As of 12 October 2026 it holds 1,543 datasets.
- UK Data Service, the British archive of social and economic data, maintains three access levels. Open data is downloaded without registration under open licences. Safeguarded data, after registration and agreeing to the end user licence. Controlled data, detailed and confidential, is available only to experienced researchers on application and only inside the SecureLab trusted research environment: from an approved workstation, without the right to download the data itself. In 2023 the shared open catalog listed datasets of all three levels, and the documentation for the closed ones was open.
- FAIR Data Points work the same way, but for research: a medical centre lists cohorts and patient registries in DCAT, and access to the records themselves is usually obtained on application.

More on microdata catalogs is in a separate chapter, and on licences and access modes in the chapter on licences and access.

Code list portals: a borderline case

Reference data (in Russian practice, NSI, normative reference information) means code lists and classifications used by information systems: codes, lists, value dictionaries. In Russia there are separate government portals for it: the unified reference data system on Gosuslugi, the Ministry of Health reference data portal, the Federal Compulsory Medical Insurance Fund's code lists, and the reference data systems of Moscow and St Petersburg. In June 2024 the Ministry of Health portal had about 1,800 code lists with export in XML, JSON, XLS and CSV.

Outwardly such portals look like open data portals: many datasets, machine-readable export, an API. The difference is that they have a single owner, uniform terms of use instead of a licence for each dataset, and they were created for exchange between government systems, not for publication. In other countries similar resources are more often organised as catalogs of schemas

and vocabularies. The Italian schema.gov.it, run by the Department for Digital Transformation and the statistical office ISTAT, is a national catalog of ontologies, vocabularies and data schemas in RDF with a SPARQL query endpoint. The Swiss I14Y stores code lists as concepts with versions and statuses, not as files.

There is no clear answer as to where a code list portal belongs. By content it is closer to a metadata catalog: a code list describes the values with which other data is filled in. By design, closer to an open data portal. The deciding question is what a person does on the website: downloads the code list as data, or checks codes in their own system against it.

Where it breaks

The card is alive, the catalog is gone. A registry describes a catalog or vocabulary that no longer exists: the address has changed, the website is closed, but the card lives on. For a catalog registry this is an occupational hazard. On 12 October 2026 the FAIR Data Point index knew 230 points, but only 74 were active. Another 80 did not respond, 47 were listed as inactive, and 28 failed validation of their descriptions. Points are often set up for a project, and when it ends the server is switched off. A card should be marked as inactive rather than silently deleted: someone may have linked to the old address.

The list mixes catalogs with things that are not catalogs. In 2023 Open Data Inception records included data laws, policy pages and individual datasets. In October 2026 the website still says "2600+", although the source dataset has 3,763 records: the counter and the content live separately. The larger the list, the more it needs an explicit inclusion criterion and a note of verification.

The list cannot be downloaded. In 2022 datacatalogs.ru was a website on top of an Airtable database: convenient for manual editing, but the service is proprietary, and there was no export of the catalog as open data. A list that cannot be downloaded whole cannot be checked or reused by other projects. The dataportals-registry, started in 2023, is organised the other way round: each record is a file in an open repository, and the whole list is exported as one file.

The vocabulary changes without versions. A theme code was renamed, and all catalogs that referred to it now show an empty field. A vocabulary needs persistent term addresses and a note of which term is obsolete and what replaced it. Registries following ISO/IEC 11179 and SDMX keep versions in the model itself; simple vocabularies on a website often have none at all.

The platform moves to the cloud. Aristotle Metadata Registry's open code was removed from the repository in 2020, and the product is developed only as a cloud service. A cloud registry is convenient as long as the contract is alive. An export of all elements in an open format must be written into the contract before work begins, not at the time of migration.

The national list does not reach the shared one. On 12 October 2026 the European register of protected data on data.europa.eu held 10,757 descriptions, but only from 12 countries. 69% of the records (7,408) came from Norway, and among them were ordinary open geodata (orthophotos and terrain data marked "open access"). Ireland, whose own catalog has 1,543 datasets, passed 137

records there. Countries still understand differently what counts as protected data and what to pass up, and harvesting works only for those who serve DCAT-AP according to the harvester's rules.

The vocabulary is closed to programs. A vocabulary is needed above all by programs, but websites sometimes refuse them along with bots. On 12 October 2026 Finto and METEOR responded to requests from a checking program with a 403 error, including with an ordinary browser header, while the REST API of other Skosmos installations responded. If a metadata catalog is closed by a robot filter, only people can use it, which means no portal will refer to it. The general analysis is in the anti-pattern "The door is closed to programs".

How to check a metadata catalog

Four steps for a quarter of an hour.

1. **Take one record and open its address with a program.** Does the address return RDF, JSON or SDMX, not only a page? Has it stayed the same after the website moved, according to the web archive?
2. **Find an obsolete record.** Can you see that it is obsolete and what replaced it, or was it simply deleted?
3. **Download the whole catalog.** Is there a full export or only paginated search?
4. **Check where the catalog is registered.** Is it in BARTOC, re3data, FAIRsharing, the FAIR Data Point index, and does the address match there?

Typical addresses for different platforms (checked 12 October 2026):

```
# FAIR Data Point: description of the point in DCAT (Turtle)
curl -s -H "Accept: text/turtle" "https://fdp.healthdata.nl/"
# FAIR Data Point index: all registered points and their state
curl -s "https://home.fairdatapoint.org/index/entries/all"
# Skosmos: list of vocabularies of an installation
curl -s "https://data.ub.uio.no/skosmos/rest/v1/vocabularies?lang=en"
# OLS: how many ontologies are in the catalog
curl -s "https://www.ebi.ac.uk/ols4/api/ontologies?size=1"
# SDMX registry: the frequency code list from which the code D in the ECB series key comes
curl -s "https://registry.sdmx.org/sdmx/v2/structure/codelist/SDMX/CL_FREQ/latest?
format=sdmx-json"
# EU vocabularies: the theme "Transport" as an address, response in RDF
curl -sL -H "Accept: application/rdf+xml"
"http://publications.europa.eu/resource/authority/data-theme/TRAN"
# I14Y: concepts with versions and registration statuses
curl -s "https://api.i14y.admin.ch/api/public/v1/concepts?page=1&pageSize=1"
# re3data: all repositories in one response
curl -s "https://www.re3data.org/api/v1/repositories"
```

A response in RDF, SKOS or SDMX at a record's address is a good sign. If the catalog responds only with a page for people, and with an error or a robot check to a program, it is hard to rely on. The summary list is in the checklists, and general commands are in the chapter on APIs and bulk download.

How it differs from its neighbours

Neighbour	Main difference
Open data portal	a portal serves files and tables, a metadata catalog serves descriptions, vocabularies and lists; a code list portal is a borderline case
Indicator catalog	an SDMX registry stores code lists and table structures, while an indicator catalog serves the values themselves
Aggregator and dataset search engine	an aggregator collects as many datasets as possible for search; a catalog of catalogs describes the catalogs themselves and does not replace a search across all data
Microdata catalog	a microdata catalog describes surveys and provides files on application; a catalog describing closed data only says that the data exists and whom to contact
Research repository	a repository stores data and issues DOIs; a metadata catalog stores descriptions, and the data stays with the owner
Enterprise data catalog	describes tables and pipelines inside a company for its employees; a public metadata catalog is open to all and describes shared vocabularies and the data of many organisations

A summary table of all types is in the chapter Comparing Types and Hybrids.

Sources

- dataportals-registry export, release 1.27.0 of 15 October 2026: number of catalogs, owners, countries, platforms with the share having an API, interfaces. The queries are in the appendix.
- SKOS, W3C Recommendation of 2009; XKOS, DDI Alliance, version 1.0; OWL 2; ISO/IEC 11179-3:2023, fourth edition of 16 January 2023; SDMX
- Global SDMX Registry: the `CL_FREQ` code list and number of structures via the API, 12 October 2026; the theme "Transport" in the EU vocabularies; the I14Y open API; checked 12 October 2026
- Data Governance Act, Regulation 2022/868, Article 8; European Register for Protected Data on data.europa.eu: number of records and countries via the search API, 12 October 2026; harvesting rules for national points
- European Health Data Space Regulation, summary on EUR-Lex: deadlines; draft implementing act under Article 77, 2026; HealthDCAT-AP Release 7, draft specification of May 2026
- FAIR Data Point specification, source code and releases, index of points: state of points via the index API, 12 October 2026
- Skosmos and its releases; OntoPortal; Ontology Lookup Service; MOLGENIS; Metadata Browser; checked 12 October 2026
- Fusion Metadata Registry and the BIS project page
- Aristotle Metadata Registry, public repository: last change 28 August 2020, the repository contains only an announcement of the move to a cloud service; checked 12 October 2026
- re3data: number of repositories via the API; BARTOC: number of vocabularies via the API; Linked Open Vocabularies: counter on the home page; OLS: number of ontologies; all 12 October 2026; FAIRsharing
- Catalogs of catalogs, checked 12 October 2026: dataportals.org, counter on the home page; Open Data Inception and its source dataset, number of records via the API; the CKAN ecosystem catalog; dataportals-registry, release 1.27.0 of 15 October 2026

- Public Service Data Catalogue, Ireland: counter on the home page, 12 October 2026; UK Data Service, Access conditions: three access levels, checked 12 October 2026
- Logius, Stelselcatalogus: purpose and the responsibility of register holders; Stelselcatalogus updated, 3 June 2026
- schema.gov.it: Italy's national data semantics catalog, checked 12 October 2026; Russian reference data portals did not open from abroad on 12 October 2026, information about them is from posts
- I. Begtin, Telegram (in Russian): launch of datacatalogs.ru, 3 November 2021; datacatalogs.ru on Airtable, 263 catalogs, 23 June 2022; Open Data Inception and datacatalogs.org, 15 December 2021; meta-catalogs: Geolode, DataShades, Open Data Inception, 15 May 2023; FAIRsharing, 2 January 2023; Ireland's open catalog of non-open data, 27 May 2022; UK Data Service and three access modes, 26 October 2023; reference data portals in Russia, 22 May 2023; schema.gov.it, 3 June 2023; the Ministry of Health code list portal, 14 June 2024

Search Engines and Aggregators

A dataset search engine does not publish datasets of its own. It takes descriptions from many catalogs and offers one search across them. The files themselves stay with the source, and a person follows a link to get there.

The practical conclusion for both sides. For someone looking for data, a search engine is useful as a first step when it is not known who holds the data. But "nothing found" does not mean the data does not exist: each search engine sees only its own part of the world. For a catalog owner, search engines bring users almost for free, provided the catalog serves descriptions in a standard way. An organisation rarely needs a search engine of its own: it is expensive to run, and such services close down more often than others.

According to the dataportals-registry export (release 1.27.0 of 15 October 2026), there are 124 search engines. This is the most blurred type in the registry: more than half of the records are showcases of two systems, and many of the largest search engines are recorded as other types. The analysis is in the section The number of search engines is not a measure.

Aggregator or search engine

Both take other people's descriptions and publish nothing themselves. The difference is whom the pages are made for. An aggregator has its own page with a persistent address for each dataset, and shows it deliberately to ordinary search engines like Google: there is a sitemap, and the page opens without scripts. The aggregator wants to be found. A search engine shows only results for a query, and has no separate pages that someone else would index: a person goes straight to the source.

- **NIAID Discovery Portal** of the US National Institute of Allergy and Infectious Diseases collects descriptions of biomedical datasets from dozens of repositories. As of 12 October 2026 it holds 14.2 million records, each with its own page, and `robots.txt` lists 26 sitemaps. It is an aggregator.
- **find.data.gov.scot** calls itself a search engine for Scottish data, but each of its 26,500 datasets from 70 sources has its own page, and all are listed in the sitemap (October 2026). By design it is also an aggregator.
- **Google Dataset Search** keeps no dataset pages of its own: a card in the results leads to the source's website. It is a search engine.

For a data owner the difference is practical. A search engine creates no extra copies. An aggregator does: in Google an aggregator's page with the same dataset may appear next to yours. How to make the search engine treat yours as the main one is covered in the chapter on discoverability.

From here on in the chapter both kinds are called search engines unless the difference matters.

Who builds them

Dataset search engines are built mainly by the research world. Research infrastructures and funders pay for search across research outputs because they require scientists to publish data and want to see what has been published.

Owner	Search engines	Share
Universities and research organisations	84	68%
International organisations	17	14%
Companies	8	6%
Communities and civil society organisations	7	6%
Central government	6	5%
Other	2	2%

Research networks: OpenAIRE and the community showcases built on its graph, EUDAT with the B2FIND search engine, national research data search engines of Sweden, Australia, China and Turkey.

International organisations search within their sector: the International Aid Transparency Initiative (IATI) search engines for development aid projects, the OECD's eChemPortal for information on chemicals, Ocean InfoHub for ocean data, GARDIAN of the CGIAR research consortium for agricultural data.

Companies: Google with Dataset Search, GeoSeer with search across geoservices, Transitland with a catalog of transit schedules.

Governments rarely build search engines. More often they build a national portal that itself harvests agencies and regions: the US data.gov, the European data.europa.eu. In substance this is the same work one level up, but the registry records such websites as open data portals. A notable exception is Scotland, where the search engine appeared before the portal.

By territory: 37 search engines cover the European Union as a whole, 21 the whole world, 23 are in Italy, 8 in Germany. Other countries have one or two each.

The number of search engines is not a measure

Of 124 records, about forty are separate services with their own index. The rest are as follows:

- **50 are OpenAIRE CONNECT showcases.** This is an OpenAIRE cloud service: a community, a university alliance or a research infrastructure gets its own search website that shows a slice of the shared OpenAIRE graph according to that community's rules. A showcase does no harvesting of its own. In essence it is one search engine with fifty entrances.
- **28 are VIVO systems.** VIVO is open-source software for an organisation's researcher profiles: people, publications, projects, sometimes datasets. Of the 28 installations, 17 are Italian UNIFIND systems installed for Italian universities by a single consortium. Hence Italy's 23 "search engines".
- **6 are Idra.** All six are demo and test instances, and none is active, although the software itself is still being developed.

On the other hand, many large search engines are recorded differently in the registry. Dateno, NIAID Discovery Portal, Research Data Australia, GESIS Search and DataONE are listed as research repositories. B2FIND, NASA's Earthdata Search, OmicsDI, data.europa.eu and data.gov as open data portals. This is partly fair: many of them both store something of their own and collect from others. But the world of search engines cannot be counted from one registry type. There are opposite errors too: the search engine of the Dutch archive DANS EASY is recorded as active, although it closed on 2 December 2024 and its datasets moved to the new DANS Data Stations.

What they collect

The main unit of a search engine is a record about someone else's dataset. It contains exactly what the source provided: title, description, publisher, a link to the dataset page, sometimes a licence, territory, dates and links to files. If the source did not specify a licence, the search engine will not have one either: it has no right to invent it.

On top of this the search engine adds its own work:

- **normalises fields**: country, format, licence, theme, so that filters work on them;
- **merges duplicates**: one dataset from a city, regional and national portal should become one record;
- **remembers the source and harvest date**, so that it is clear where a record came from and how fresh it is;
- **sometimes looks into the data itself**: downloads the file and parses its columns, like Auctus, or puts a copy of the data into its own storage.

What exactly gets into a search engine depends on its sources. Research search engines collect DataCite records and repositories, government ones open data portals, geosearch layer catalogs and geoservices, and statistics aggregators such as DBnomics the time series themselves (in October 2026, 94 sources and about 47,000 series datasets). There is no general search engine that sees all types of data equally well.

How they work

Outwardly search engines look alike: a search box, filters, a list of cards. Inside they are organised differently, and the design determines what will be in them and what never will. Seven models are the most common.

Search by page markup

The search engine makes no agreements with anyone. It crawls the web like an ordinary search engine and takes descriptions from schema.org/Dataset markup on dataset pages. This is how Google Dataset Search works: in February 2023, more than 45 million datasets from 13,000 websites; Google has published no newer official figure. The coverage is huge, but only of sites that have added markup, and the search engine cannot verify it. Geoportals and indicator catalogs rarely add markup, so their data is almost absent here.

Harvester using catalog protocols

The search engine keeps a list of sources and maintains a harvester for each type of platform: the CKAN API, DCAT files, OAI-PMH, CSW, the ArcGIS interface. This is how Dateno, B2FIND, BASE and OpenAIRE work. On 12 October 2026 Dateno showed 22 million datasets from 5,200 sources; at the end of 2024 it had harvesters for more than 15 types of platform. B2FIND of the EUDAT consortium collects European research repositories: in October 2026 it held 1.47 million records.

This model needs a registry of sources with manual checking and a harvester for each platform. A custom portal with its own interface is seen last or not at all. Dateno's registry is dataportals-registry, from which the figures in this book were calculated. The registry was originally created for the search engine, and Dateno indexes portals from their descriptions in the registry: it chooses the harvester by the recorded platform and API addresses.

Sector aggregator

It collects the sources of one sector and knows its description schemas, so it offers filters that general search engines lack: organism and experiment type, instrument and satellite, substance and regulation. Examples:

- **NIAID Discovery Portal**: biomedicine, 14.2 million records, 62% from a single database, NCBI GEO;
- **OmicsDI**: data from "omics" research (genomics, proteomics and the like), 4.9 million datasets from 29 databases (October 2026);
- **DataONE**: Earth and environmental sciences, about 1.08 million current descriptions excluding superseded versions, from 83 registered member repositories;
- **Earthdata Search**: NASA Earth observation data, 45,000 collections; each collection is a series of products that may contain millions of files;
- **the OECD's eChemPortal**: information on chemicals from government databases of various countries;
- **Transitland**: public transport schedules in GTFS format from around the world.

A sector aggregator is the most durable kind: it has a client who understands why it is needed.

National search engine

Instead of one common portal, a country or region collects descriptions from hundreds of local sources. find.data.gov.scot: 26,500 datasets from 70 sources; the data.gov.scot portal appeared later, in 2026. Research Data Australia is a search across Australian universities, research organisations and agencies: in July 2025, about 224,000 records, of which about 96,000 had data available online. Similar services exist in Sweden (researchdata.se), China (Findata) and Turkey (the academic archive Harman).

This model needs agreements with sources, harvesting via DCAT and a link to the original source in every card. The Scottish path of "search first, portal later" is worked through in the chapter "Federation and Harvesting".

Service crawler

The search engine finds servers of a certain standard by itself and queries them without asking the owners. GeoSeer crawls WMS, WFS, WCS and WMTS geoservices: 3.5 million layers from 4,200 servers. STAC Index keeps a list of catalogs of satellite imagery and other spatial files in the STAC standard: in October 2026, 149 catalogs, of which 63 are servers with an API and 86 are static sets of files. A crawler sees what is not marked up and not listed in any catalog, but only in its own standard.

Showcase over a shared index

It collects nothing itself but shows a slice of a large index according to a community's rules: OpenAIRE CONNECT showcases, the shared FAIR Data Point index, VIVO profile systems. Launching such a showcase is cheap, but its content is the same as in the shared index, with all its gaps.

Search by content

The search engine downloads the files themselves, parses columns, their types, dates and coordinates, and searches on these. This is how New York University's Auctus works: in it you can find a dataset that complements your table through a shared column, for example a district code or a date. The price is coverage: in October 2026 Auctus held about 55,000 datasets from 64 sources, half of them from the Humanitarian Data Exchange. Downloading and parsing every file is expensive, so search by content still remains a niche.

What follows from this

Model	Example	What it cannot do without	Weak spot
By page markup	Google Dataset Search	schema.org markup at the sources	takes markup at its word, does not see unmarked pages
Harvester using protocols	Dateno, B2FIND, OpenAIRE	a source registry, harvesters for platforms, deduplication	custom portals without a standard interface
Sector aggregator	NIAID, DataONE, Earthdata	sector description schemas and filters	sees only its own sector
National search engine	find.data.gov.scot	agreements with sources, harvesting via DCAT	depends on one team and one budget
Service crawler	GeoSeer, STAC Index	finding servers, parsing responses of standard services	sees only its own standard
Showcase over an index	OpenAIRE CONNECT	a shared index, configuring selection rules	content no richer than the shared index
By content	Auctus	downloading and parsing files	small coverage

Standards

A search engine has almost no standards of its own: it reads what the source serves to machines. So its standards are the sources' standards:

- **schema.org/Dataset** on the dataset page: this is what Google Dataset Search reads. Without markup the dataset does not exist for it.
- **DCAT and its profiles**: this is how the pan-European and national portals harvest each other, and how independent search engines harvest open data portals.
- **OAI-PMH** following the OpenAIRE guidelines and **the DataCite schema**: this is how research aggregators harvest repositories and DOI records.
- **CSW**: this is how layer catalogs of geoportals are harvested.
- **STAC**: this is how imagery catalogs are listed and linked.
- **FAIR Data Point**: this is how research catalogs publish descriptions in RDF and register in a shared index.

Then the descriptions are normalised, deduplicated and ranked. How such a chain works is covered in the chapter "Federation and Harvesting".

There is no common standard for search results: each search engine has its own API, open for some (DataCite, OpenAIRE, data.europa.eu), paid for others (GeoSeer) or key-based (Dateno), and Google Dataset Search has none at all.

The obligations of the parties are simple. A source should provide a stable export address and state the licence for each dataset. A search engine should not show a description whose original page no longer opens, or at least should mark it.

Platforms

Platform	What it is	Search engines in the registry	API
OpenAIRE CONNECT	cloud service for community showcases on top of the OpenAIRE graph	50	the shared OpenAIRE API
Custom build	Google Dataset Search, GeoSeer, find.data.gov.scot, STAC Index and others	35	for 20%
VIVO	open-source researcher profile system, BSD-3-Clause licence	28	for 25%
Idra	open-source platform for federating open data portals, AGPL-3.0 licence	6	all six are inactive instances
FAIR Data Point	indexes of FAIR Data Points	2	for both

One search engine each runs on DataLad (an index of datasets stored in Git repositories managed by DataLad) and on the software of the European geophysical infrastructure EPOS.

Idra is the closest to a "ready-made search engine". It is open-source software from the FIWARE catalog: it federates portals on CKAN, DKAN and Socrata and Orion data brokers, accepts DCAT-AP exports, can parse pages of portals without an API, normalises descriptions to DCAT-AP and serves them through a REST API and SPARQL. Release 4.0.0 came out in March 2025, and changes were made to the code as recently as October 2026. But the registry has not a single active installation: all six are demo instances of developers and of projects that have closed. When choosing Idra, it is worth finding a live installation at a similar organisation first.

A portal platform with harvesting. A national or regional aggregator is more often built on a portal platform that can harvest other catalogs:

Platform	Harvesting other catalogs	What else is needed
Piveau	in the core: harvesting and quality checking of DCAT descriptions are the basis of the platform; data.europa.eu runs on it	—
CKAN	through extensions: the ckanext-harvest framework, RDF harvesting in ckanext-dcat, CSW harvesting in ckanext-spatial	harvesters for other protocols are separate extensions or custom work
Idra	in the core: CKAN, DKAN, Socrata, DCAT-AP exports, page parsing	your own check that the project is alive

B2FIND, for example, runs on CKAN, on the old 2.9.8 branch that no longer gets updates, with its own harvesting extension. The registry records such a website as an open data portal, although in substance it is an aggregator. Details on extensions are in the chapter on CKAN.

A custom build means harvesters plus a search engine such as Elasticsearch, OpenSearch or Solr. Almost all large search engines are built this way. There is no ready-made software that harvests CKAN, DSpace, GeoNetwork and ArcGIS out of the box, so a large general search engine is always a custom build.

Examples

General

- Google Dataset Search: searches schema.org markup on any websites.
- Dateno: a harvester of open data portals, geoportals, research repositories and statistics, 22 million datasets; API with a key.

Research

- OpenAIRE Explore: research outputs from Europe and the world, including 108 million records about datasets.
- DataCite Commons: all records with DataCite DOIs, including datasets.
- B2FIND: European research repositories, runs on CKAN.

Sector

- NIAID Discovery Portal: biomedicine.
- OmicsDI: genomics, proteomics and similar data.
- DataONE: Earth and environmental sciences.
- Earthdata Search: NASA Earth observation data.
- Transitland: public transport schedules.

National and regional

- find.data.gov.scot: Scotland.
- Research Data Australia: Australian research data.
- data.europa.eu: national portals of European countries; formally an open data portal, in practice an aggregator of aggregators.

Geodata

- GeoSeer: layers of WMS, WFS, WCS and WMTS geoservices.
- STAC Index: catalogs of imagery and other spatial files.

By content

- Auctus: searches by columns and values inside files.

How to evaluate a search engine

The number of datasets on the home page says little. A search engine into which two or three large research aggregators have been loaded gets tens of millions of records and remains a search engine for research data. It is more useful to check five things.

1. **Variety of data.** Do government open data, geodata, research datasets, statistics and machine learning data turn up side by side? If almost everything is of one type, it is a sector search, even if it calls itself general.
2. **Country coverage.** Does anything turn up for a small country, not only for the US and Western Europe? A simple check is a query with the name of a small state in its own language.
3. **Sources.** How many catalogs have been harvested, and is their list published? If the list is closed, you cannot tell what is missing from the search.
4. **Links to files.** Does a card lead to the file itself or an API, or only to the dataset page at the source, or does it merely report that the dataset exists?
5. **Filters.** Can the results be narrowed by country, source type, format, licence, theme? There is almost nothing to rank datasets by links, so without filters long result lists are useless.

These criteria pull in different directions: chasing the number of records is easier than chasing variety and links to files. A good search engine shows honestly what it has bet on.

How much is in the large indexes

The figures below cannot be compared directly: each index counts its own thing as a "dataset". They show the order of magnitude and what these magnitudes are made of.

Index	What it collects	"Dataset" records	Date	Caveat
Google Dataset Search	pages with schema.org markup	more than 45 million from 13,000 websites	February 2023, according to Google	no newer official figure
DataCite	datasets with a DOI	76.4 million	12 October 2026	42.8 million from a single Japanese fusion science institute; a link to the content is given for about 12,000 records
OpenAIRE Graph	research outputs, including data	108 million	12 October 2026	the link leads to the dataset page at the source, not to a file
BASE	research repositories via OAI-PMH	25 million, of which 22 million from DataCite	November 2024, from a post by I. Begtin	in October 2026 the website is closed to programs with a robot check, rechecking was not possible

Index	What it collects	"Dataset" records	Date	Caveat
Dateno	open data portals, geoportals, research repositories, statistics	22 million from 5,200 sources	12 October 2026, from the home page counter	in November 2024 there were 19 million from more than 5,000 catalogs
NIAID Discovery Portal	biomedical repositories	14.2 million	12 October 2026	62% of records from a single database, NCBI GEO
OmicsDI	genomics, proteomics and similar research databases	4.9 million from 29 databases	12 October 2026	some of these databases are also harvested by NIAID
GeoSeer	layers of WMS, WFS, WCS, WMTS geoservices	3.5 million from 4,200 servers	12 October 2026	sees only services of these standards
B2FIND	European research repositories	1.47 million	12 October 2026	—
DataONE	Earth and environmental science repositories	1.08 million	12 October 2026	excluding superseded versions; with all versions 1.64 million
Earthdata (CMR)	Earth observation data of NASA and partners	45,000 collections	12 October 2026	a collection is a series of products, not a file

Three caveats matter more than the numbers themselves. The first is granularity: more than half of DataCite's "datasets" came from one institute that assigns a DOI to every experiment, while NASA, on the contrary, counts a whole series of files as one collection. Why the "number of datasets" field differs by orders of magnitude between catalogs is covered in the chapter The World of Catalogs in Numbers. The second is duplication: some indexes harvest others, so 22 of BASE's 25 million records are DataCite. The third is links to files. Research indexes collect descriptions via OAI-PMH and from DataCite, which usually have no list of files. Of 76.4 million DataCite records of type "dataset", only about 12,000 have a link to the content (the `contentUrl` field). The card reports that the dataset exists, but it cannot be downloaded from the search.

Searching for geodata

Google Dataset Search hardly sees geodata: most geoportals do not serve schema.org markup. That is why there are separate tools for them, and each has its own blind spot.

GeoSeer crawls servers of standard geoservices. In October 2026 it held 3.5 million layers from 4,200 servers in 97 countries, most of them in the German domain zone. Search on the website is free, the API is paid. The list of sources is not published, and according to a 2024 observation there are no filters by country or theme. The main limitation is that GeoSeer sees only WMS, WFS, WCS and WMTS services. A layer published on an open data portal as a file or served via ArcGIS REST does not exist for it. ArcGIS Hub search, conversely, sees only sites on Esri's cloud platform.

Such services are also short-lived. The Finnish company Spatineo ran the open Spatineo Directory: in 2023 it had more than 87,000 geoservice addresses, almost all from Europe and North America. By October 2026 the directory had closed, replaced by a paid search for ArcGIS Pro. How geodata is searched for inside the geoportals themselves is covered in the chapter on geoportals.

Where it breaks

An outdated copy. The search engine took a description a year ago, the source has since moved or removed the dataset, but the card in the search remains and leads to a dead link.

Triple counting. One dataset ends up in the search three times: from a municipal, a regional and a national portal. The aggregator looks richer than its sources and at the same time inflates the number of datasets.

An aggregator harvests an aggregator. It is easier for a search engine to connect one large aggregator than a hundred of its sources. But then it gets secondary copies with fields already lost, and duplicates: 22 of BASE's 25 million records came from DataCite. Statistics aggregators face a similar choice: take a series from the national statistical office or from the World Bank, which has already republished it. The rule is simple: harvest the original source, and connect an aggregator only for sources you cannot reach yourself.

Someone else's copy instead of the source. In the Dateno index in 2024, 244,000 datasets were attributed to Russia, 195,000 of them from the geoportal of one research centre. This turned out to be a clone of dataset descriptions of the US weather service NOAA, not Russian data. An aggregator needs a source registry with manual checking, otherwise one such website distorts all the statistics for a country.

Search by description only. Neither Google Dataset Search nor DataCite Search can find a dataset that has a column with an organisation code or coordinates: they do not index fields and values inside files. Auctus shows that this is possible, but so far on tens of thousands of datasets, not millions.

Markup is taken at its word. Everything Google Dataset Search knows about a dataset was written by the owner of the page, including the "free to access" flag. The search engine cannot check it. schema.org/Dataset markup is also added by data vendors for promotion, and their cards are mixed with open datasets. In November 2024 a query about the population of Andorra, even with the "free" filter, returned paid statistics services first, with UN data somewhere below. The same query in Spanish returned nothing, although data on Andorra exists on statistical and geoportals that do not add markup.

Nothing to rank by. Ranking datasets by links, like web pages, hardly works: datasets are rarely linked to. That is why data search relies on filters (country, catalog type, format, licence, theme), and the quality of filters depends on how carefully the sources filled in their descriptions.

Search engines close down. Harvesting has to be maintained for years: sources move, change platforms and APIs. Spatineo Directory has closed, all six Idra installations in the registry are inactive, the DANS EASY search closed together with the archive in December 2024, and BASE has closed itself to programs with a robot check. If your work relies on a search engine, keep a link to the original source, not to the search engine's card.

In the rough trust score that the registry calculates, search engines get a lower score: they do not store data and are not responsible for the source's licence. This is not a verdict but a reminder not to confuse search with the original source. More in the chapter on methodologies.

How to find data through search engines

Start not with the largest search engine but with the one that knows your type of data.

What you are looking for	Where to look first
Government data	the country's national portal, in Europe data.europa.eu; then Dateno
Statistics	the statistical office website and its indicator catalog, DBnomics; then Dateno
Geodata	the national geoportal, GeoSeer, Dateno, ArcGIS search; for imagery, STAC Index
Research data	DataCite Commons, OpenAIRE Explore, a sector aggregator (NIAID, OmicsDI, DataONE, Earthdata)
Data from any websites, including commercial	Google Dataset Search
A dataset that complements your table	Auctus

A few habits save time:

- **Write the query in the country's language and in the source's words.** Spain's statistical office calls population "población", not "population". Andorra showed in November 2024 that the same search engine finds different things in different languages.
- **Narrow with filters rather than scrolling.** Country, format, licence and source type cut out more noise than refining the words in the query.
- **Check at the source.** Look at the licence, update date and the file itself on the publisher's dataset page: the search engine may have taken the description long ago and with losses. Cite the source too, not the search engine's card.
- **Do not trust empty results.** If nothing was found, go directly to the portal or indicator catalog of the country you need. Many geoportals and statistical websites are not in any general search engine.

Several search engines have an open API, and you can check what each of them knows about a topic with one command. The responses were checked on 12 October 2026:

```
# DataCite: DOI records of type "dataset", the count is in meta.total
curl -s -g "https://api.datacite.org/dois?query=Andorra&resource-type-
id=dataset&page[size]=1"
# OpenAIRE Graph: datasets, the count is in header.numFound
curl -s "https://api.openaire.eu/graph/v1/researchProducts?
type=dataset&search=Andorra&pageSize=1"
# data.europa.eu: datasets of European portals, the count is in result.count
curl -s "https://data.europa.eu/api/hub/search/search?q=Andorra&filter=dataset&limit=1"
# NIAID Discovery Portal: biomedicine, the count is in total
curl -s "https://api.data.niaid.nih.gov/v1/query?q=Andorra&size=0"
# Earthdata (CMR): the number of collections is in the CMR-Hits header
curl -sI "https://cmr.earthdata.nasa.gov/search/collections.json?keyword=Andorra"
# Auctus: search by columns and values inside files
curl -s -X POST "https://auctus.vida-nyu.org/api/v1/search" \
  -H 'Content-Type: application/json' -d '{"keywords": ["Andorra"]}'
```

For this query DataCite found about 9,500 datasets, OpenAIRE about 9,200, data.europa.eu 585, NIAID 101, Earthdata 60 collections. The numbers say not where there is more data about Andorra, but how each search engine searches: research indexes find the word in the description

of any dataset where the country is mentioned even in passing, while data.europa.eu sees only European portals.

For search engines to find your own catalog, you need a description export, markup and registration in relevant registries. This is covered in the chapter on discoverability, and for publishing without a portal in the chapter "Making a Publication Findable".

When you need your own search engine, and when you do not

An organisation almost never needs its own "worldwide" dataset search engine: such services already exist, and even they struggle to maintain coverage. One of the cheaper options is usually enough:

- **get into existing search engines**: serve descriptions in DCAT or OAI-PMH, add schema.org markup, register in relevant registries;
- **set up a showcase over a shared index**: for a research community or university alliance whose outputs are already in OpenAIRE, an OpenAIRE CONNECT showcase is enough;
- **collect datasets on a portal platform with harvesting**: for a region or country where sources already run on CKAN and other standard platforms.

A search engine of your own is justified in two cases. The first is a sector one: general search engines do not know the sector's description schemas and do not provide the needed filters, and the sector has a client ready to pay for years, as with NIAID or DataONE. The second is a national one, when data is already scattered across a hundred websites and cannot quickly be brought into one portal, as in Scotland. In both cases three questions are worth answering before starting: who keeps the source registry, who fixes the harvesters when sources change platform, and what will happen to the card addresses if the project closes. How to record such a decision with a justification is in the chapter "Justifying a Decision".

How it differs from its neighbours

Neighbour	Main difference
Open data portal	publishes its own datasets and is responsible for them. If a portal both accepts publications and harvests other catalogs, like data.gouv.fr, it is a hybrid
Research repository	stores files and assigns DOIs to them. A search engine stores nothing and issues no DOIs
Metadata catalog	describes catalogs, structures and code lists. A catalog of catalogs such as re3data answers the question "where to look", a search engine "what exists"
Geoportal	shows layers on a map and serves services. GeoSeer searches across layers but keeps no services itself
Indicator catalog	stores the series themselves and their methodology. A statistics aggregator such as DBnomics is closer to an indicator catalog, because it copies series, not just descriptions
A copy of someone else's cards	republishes descriptions without a link to the source and without a harvest date. This is not an aggregator but a copy

A summary table of all types is in the chapter Comparing Types and Hybrids.

Sources

- dataportals-registry export, release 1.27.0 of 15 October 2026: owners, territories, platforms and share with an API of search engines, the list of records, and large search engines recorded as other types. The queries are in the appendix.
- Dataset markup in Google's documentation, OpenAIRE guidelines, STAC, FAIR Data Point
- OpenAIRE CONNECT: description of the community showcase service; OpenAIRE Graph API: the query `researchProducts?type=dataset`, 12 October 2026
- VIVO and the VIVO repository: BSD-3-Clause licence, release 1.15.1 of 25 June 2026
- Idra: description of capabilities in the README, AGPL-3.0 licence, release 4.0.0 of 27 March 2025, latest changes 8 October 2026; Piveau
- B2FIND: platform version from the status_show response and number of records via the CKAN API, 12 October 2026
- Dateno: counters of datasets, sources and countries on the home page, 12 October 2026
- NIAID Discovery Portal: number of records and distribution by source via a query to the portal API, sitemaps in robots.txt, 12 October 2026
- OmicsDI: number of datasets via the search API and the list of databases via the databases API, 12 October 2026
- DataONE: number of descriptions via the search API and the list of member repositories via the node registry, 12 October 2026
- NASA Common Metadata Repository: number of collections in the `CMR-Hits` response header, 12 October 2026
- STAC Index: list of catalogs via the API, 12 October 2026
- Auctus: number of datasets by source via the statistics API, source code, 12 October 2026
- DBnomics: number of sources and datasets via the API, 12 October 2026
- find.data.gov.scot: counters on the home page and the sitemap, 12 October 2026
- DANS EASY: notice of closure on 2 December 2024 and transfer of datasets to DANS Data Stations, checked 12 October 2026
- Datasets at your fingertips in Google Search, Google Research blog, 28 February 2023: 45 million datasets from 13,000 websites
- DataCite REST API: queries to `api.datacite.org/dois` filtered by the "dataset" type, by publisher and by the presence of `contentUrl`, 12 October 2026
- GeoSeer statistics and API terms, 12 October 2026
- API responses to the query "Andorra" in DataCite, OpenAIRE, data.europa.eu, NIAID and CMR, 12 October 2026
- I. Begtin, Dataset search engines as global data discovery systems (in Russian), May 2022: search by description only
- I. Begtin, Open data in the world, in facts and theses (in Russian), March 2023: data vendors in Google Dataset Search results
- I. Begtin, Updates in Dateno (in Russian), November 2024: filters instead of link-based ranking; Why data on Russia is so hard to find (in Russian), November 2024: the NOAA description clone

- I. Begtin, Telegram (in Russian): criteria for a global dataset search engine, 30 June 2023; index coverage indicators, 22 March 2024; aggregator and search engine, using DataMed as an example, 14 September 2024; NIAID Discovery Portal, 28 November 2024; find.data.gov.scot, 3 November 2024; an overview of search engines with figures, 25 November 2024; DataCite Commons without links to files, 25 September 2024; GeoSeer, 17 August 2024; Spatineo Directory, 30 July 2023; the Andorra query in Google Dataset Search, 16 November 2024; Auctus and search by content, 2 June 2022; B2FIND, 11 August 2022; DBnomics and the choice between original and secondary sources, 19 November 2024; Dateno harvesters for different platforms, 31 December 2024; Research Data Australia, 17 July 2025
- Federation and Harvesting

API Catalogs, Marketplaces and Dataset Lists

The three types in this chapter are rare. According to the dataportals-registry export (release 1.27.0 of 15 October 2026), there are 115 API catalogs, 56 marketplaces and 107 dataset lists: together less than 1% of 43,600 catalogs. They are easy to miss in the overall statistics and easy to confuse with a "real" portal.

Each of them answers its own question about the boundary of the concept. What if a catalog entry is a programming interface rather than a dataset? What if a dataset is a product, and access to it is sold under a contract? What if there is no catalog at all, just a page of links?

The main conclusion of the chapter: all three types work well as a complement to a data catalog and badly in its place. An API catalog without dataset cards hides what data stands behind the interfaces. A marketplace with closed descriptions hides the very fact that the data exists. A list without licences and dates quickly becomes a dead end. The remedy is the same in all three cases: an open description in a common standard next to the main showcase.

	API catalog	Marketplace	Dataset list
What gets its own page	a programming interface	a product: a file, a subscription, access to a table	a link to a file or someone else's catalog
Catalogs in the registry	115	56	107
Main owner	central government, 45%	companies, 70%	none predominates
Custom build	74%	98%	89%
Programmatic access to the catalog itself	51%	21%	14%
Restricted access to at least part of the data	14%	91%	1%
Description standard	OpenAPI, AsyncAPI, DCAT data service	none common; ODRL and the Dataspace Protocol in data spaces	most often none; schema.org, Data Package

The "programmatic access" row is about the catalog itself: whether a program can get the list of interfaces, products or links. For an API catalog it is not about the interfaces it lists.

API catalogs

What it is. A developer portal: a list of the programming interfaces of a public body, a transport company or a bank with documentation, sample requests, and often key issuance and a test console. This is not the same as the API of an open data portal itself from the chapter on APIs and bulk download. There the API is a door into the catalog. Here the interface is what the catalog describes.

A good interface card asks the same questions as a dataset card, plus its own: what it serves, the base address, a machine-readable description, the access method (open, with a key, on request), limits, the data licence, service status, a contact and (the most common omission) which datasets stand behind it.

Who builds them

Almost half of API catalogs are run by central government (45%), another 23% by regions, and 10–11% each by companies and municipalities. The most records are from the US (24), Finland (8), the United Kingdom (6), Sweden and Norway (5 each).

About a third of the API catalogs in the registry (35 of 115) are transport developer portals: US metro and bus companies (MBTA, BART, WMATA, CTA), Scandinavian operators (Entur, Trafiklab, Digitraffic), London, New Zealand, Taiwan. The reason is practical: a timetable app needs real-time delay data, and the transport company issues a key for that. Schedules are usually served as a GTFS file, and vehicle movements as a GTFS Realtime feed.

What kinds there are

- **A national catalog of government APIs.** api.gov.uk in the United Kingdom, API Setu in India, the gov.br/conecta catalog in Brazil, api.gov.au in Australia, Singapore's developer portal, catalogs in Italy, Lithuania, Japan, the Dutch API register. They are often made separately from the open data portal: access to an API almost always requires a key, a test console and documentation, and the audience is developers, not analysts.
- **A single organisation's developer portal.** Central banks (the Riksbank, the Bank of Finland, Bank Negara Malaysia), statistical offices (Stats NZ, Poland's GUS), weather services (the UK Met Office, India's IMD), transport.
- **An API section in an open data portal.** In France the separate api.gouv.fr catalog was merged into data.gouv.fr: the first cards moved on 20 November 2024, in 2025 the old website began to be retired, and now its address leads to the portal's data services section. On 12 October 2026 it lists 1,265 interfaces: 1,119 open, 37 with registration and 109 with restricted access. 1,162 have a link to a machine-readable description, 1,038 to the datasets the interface serves. Norway's API catalog in the common national catalog is organised the same way: 1,071 data services next to datasets. The Finnish Suomi.fi interface catalog is built on CKAN.
- **A registry of government system interfaces.** The Estonian RIHA and X-tee, Laos' GDX. This is a record of which systems exchange what, mainly for the agencies themselves. The public can see the description but not access.
- **Community collections.** APIs.guru collects OpenAPI descriptions of public interfaces into an open repository: on 12 October 2026, 2,529 APIs and 3,992 descriptions. Next to it are publicapis.dev and the commercial Postman Public API Network and RapidAPI Hub. The German bund.dev is a volunteer project that finds undocumented German government APIs and writes documentation for them on the state's behalf.
- **MCP server registries.** The official MCP registry has been running in preview since 8 September 2025. The task is the same, to find an interface, only such a catalog is read not by a programmer but by an AI assistant. More on MCP in data catalogs is in the chapter on trends.

Standards

OpenAPI describes an ordinary request–response interface: which addresses, which parameters, what comes back. From the description, tools build a documentation page, a test console and client libraries automatically. The latest version is 3.2.0 of 19 September 2025. **AsyncAPI** does the

same for event streams: subscriptions to delay messages, sensor readings. Version 3.1.0 came out on 31 January 2026.

A data service in DCAT. Since DCAT 2 (2020) the catalog vocabulary has had a separate entity for an interface: an address, a link to a description such as OpenAPI, and a list of the datasets the service serves. This is the bridge between an API catalog and a data catalog: one record says "this dataset can be obtained through this interface". The API sections of the French and Norwegian national catalogs are built on this bridge.

An address for programs. RFC 9727 (June 2025) defines the standard address `/.well-known/api-catalog`. From it a program gets the list of a website's interfaces in linkset format without knowing in advance where the developer portal is. The standard is new. On 12 October 2026 none of the eleven national and large catalogs checked (the British, Australian, Dutch, Italian, Singaporean, Norwegian and French ones, api.data.gov, Digitraffic, Entur, bund.dev) served a list at this address. Where the response code was 200, an ordinary page of the website came back.

Sector standards. In transport these are GTFS and GTFS Realtime. In "smart cities" there is NGSI-LD, an interface for real-time sensor data; India's IUDX runs on it. Many countries have their own rules for designing government APIs; an example is the Dutch API Design Rules.

Platforms

Three quarters of API catalogs are custom builds. Among ready-made solutions the most common is a developer portal attached to an API gateway. A gateway is software that sits in front of an organisation's interfaces: it checks keys, counts requests and limits load. A developer portal comes with it, so the catalog appears almost "for free".

- **Azure API Management**: 22 catalogs in the registry, from the Bank of Finland and Fingrid to New York City's API portal and Washington's transport.
- **WSO2 API Manager** (open source, Apache 2.0): the Met Office, Latvia's VDAA. **Gravitee**: the French Polynesia portal.
- **A static site generated from files.** api.gov.uk is built from a single CSV table in an open GitHub repository. APIs.guru is also a repository of OpenAPI descriptions from which the website is generated.
- **Data catalog platforms.** In udata, on which data.gouv.fr runs, data services are a separate entity in the core: the card has fields for the OpenAPI description, limits, availability, the access request address and related datasets. CKAN has no separate entity: the ckanext-dcat extension lets you record an access service inside a dataset resource and export it to DCAT.

Capability	API gateway portal	Static site from files	udata	CKAN
Interface page with documentation	core	site generator	core, link to the OpenAPI description	no; a link from a resource
Key issuance and limit tracking	core	no	no; there are "limits" and "how to get access" fields	no
Linking an interface to datasets	no	by hand	core	the ckanext-dcat extension: an access service on a resource

Capability	API gateway portal	Static site from files	udata	CKAN
Visibility to search engines	weak, see "Where it breaks"	good	good	good

The choice follows from the task. If an organisation issues keys to its interfaces, it already has a gateway portal, and the only question is making the descriptions visible from outside. If you need to list other people's interfaces, a static site generated from a table is enough. If datasets stand behind the interfaces, they should be described in a data catalog with the interface as the way to obtain them, following the French and Norwegian model.

Examples

- The API section on data.gouv.fr: a national API catalog inside an open data portal, with the access type and related datasets on each card.
- api.gov.uk: the catalog of UK public sector APIs, built from a table in an open repository.
- API Setu: India's catalog of interfaces at all levels of government.
- Digitraffic and Trafiklab: transport data of Finland and Sweden: real-time interfaces and schedule files.
- APIs.guru: an open collection of OpenAPI descriptions.
- bund.dev: volunteer-written documentation for undocumented German government APIs.

Where it breaks

The description lags behind the interface. The documentation says one field, the response contains another. The scale can be seen from APIs.guru: on 12 October 2026, of 3,992 collected descriptions, 688 failed format validation, and 166 were unavailable at their original address.

The catalog ages. The British catalog has 233 records, and for 70 of them the last update was in 2020. A data licence is given for 36 (12 October 2026). A catalog compiled once stops reflecting live interfaces, just as an abandoned portal stops reflecting data.

The interface is described, but not the access. The catalog lists APIs that need a key but does not say how to get it, who gets one and how long to wait. In April 2024 India's IUDX described 189 datasets from 47 cities, mostly real-time streams, and not one was open: in effect it is a developer portal for partners, not an open data catalog.

The search engine cannot see the catalog. A gateway developer portal often builds the page with a script in the browser. On 12 October 2026 the HTML of the API list page of the Bank of Finland on Azure API Management did not contain a single interface name, only a menu and a login button. Neither search engines nor the web archive find such a catalog. A sitemap is marked in the registry for only 26 of 115 API catalogs.

The interface count is inflated. Of the 1,265 "APIs" on data.gouv.fr, about 800 are automatic interfaces to individual datasets from regional portals on the Data Fair platform, one for each table. This is not an error: such an interface is useful. But the number of APIs in a catalog, like the

number of datasets, says something about its structure, not about how many developer services the state has.

There is an API, but no catalog. A common case: interfaces exist but are not listed or described anywhere. They break with every website update, and the web archive does not preserve them. Russia has no national catalog of government APIs: inter-agency exchange goes through closed systems, and in 2021 Infoculture collected the public interfaces of government websites, more than a hundred. Moscow's apistore.mos.ru in 2021 contained 15 API prototypes and had not been updated since 2019; in October 2026 it does not open from abroad. In Germany the same work is done by bund.dev volunteers, in France by the state.

An interface instead of an export. If data can only be obtained by requests for one record at a time, someone who needs the whole database has to pull it down for weeks and runs into limits. An interface is a good complement to bulk download, not a replacement; more in the section "Bulk download is not a luxury".

How to check an API catalog

1. **Find the machine-readable description.** Does the interface have an OpenAPI file, and does it match the actual response to a test request?
2. **Find the access procedure.** Does it say whether a key is needed, who issues it, what the limits are and what the data licence is?
3. **Find the dataset.** Does the data the interface serves have a card in a data catalog with a link to the interface?
4. **Look as a program would.** Are interface names visible in the page HTML, and is there a list at the RFC 9727 address?

```
# RFC 9727: a website's list of interfaces in linkset format
curl -s -H "Accept: application/linkset+json" "https://example.org/.well-known/api-catalog"
# data.gouv.fr: how many interfaces are open; access_type=restricted shows how many are
closed
curl -s "https://www.data.gouv.fr/api/1/dataservices/?page_size=1&access_type=open"
# Norway: data services in the national catalog
curl -s -X POST -H "Content-Type: application/json" -d '{}' \
  "https://search.api.fellesdatakatalog.digdir.no/search/data-services"
# the British API catalog is a single table
curl -s "https://raw.githubusercontent.com/alphagov/api-catalogue/main/data/catalogue.csv"
# APIs.guru: how many descriptions are collected, how many are unavailable and invalid
curl -s "https://api.apis.guru/v2/metrics.json"
```

When it is needed

An API catalog is needed when an organisation has many interfaces and a developer needs one entry point: where to get a key, where the documentation is, what has changed. For a single agency API, a documentation page with an OpenAPI description is enough. If a dataset stands behind the interface, it should also be described in an ordinary catalog, with a link to the API as the way to obtain it. Then it will be found both by an analyst who is not searching for "API" and by a harvester such as data.europa.eu that reads DCAT.

Data marketplaces

What it is. Platforms where data is sold or provided under a contract. The product can be a file, a subscription to updates, access to an interface or to a table in the buyer's cloud storage. What distinguishes a marketplace from an open data portal is not payment as such, but a contract for each release: who received it, on what terms, what they may do with the data.

Who builds them

70% of the marketplaces in the registry belong to companies, 21% to central government. Access to at least part of the data is restricted for 91%, and programmatic access to the platform itself is marked for 21%. By country the leaders are South Korea (16), the US (12) and China (6); another 10 platforms work worldwide.

The country figures show that in many countries a data marketplace is a government project, not just a business:

- **South Korea.** Most Korean records are the sector big data platforms launched by the Ministry of Science and ICT in 2019: ten platforms covering finance, the environment, culture, transport, oncology, retail, telecommunications, small business, the regional economy and forestry. In 2020 six more were added, including fire safety and marine fisheries. All of them are gathered in a common data map: by the end of 2021 it had 6,842 kinds of data. The platforms mix free and paid data.
- **China.** State data exchanges: the first opened in Guiyang in 2015, the Shanghai Data Exchange on 25 November 2021, with 20 products on the first day. According to an August 2025 review, 49 such platforms were operating with 3,944 data providers, and another 24 had been liquidated or never opened.
- **Japan.** JDEX and KYOTO DATA MARKETPLACE.
- **The United States.** Cloud platforms (AWS, Snowflake, Databricks, Google, Microsoft) and exchanges that sell their own market data: CME DataMine, Cboe DataShop, Nasdaq Data Link. The Korean KRX and Hong Kong's HKEX do the same.
- **Data for AI.** Defined.ai, Nexdata and Centific sell datasets for training models: speech recordings, annotated images, texts.
- **European data spaces.** The Pontus-X catalog in the Gaia-X ecosystem, the Language Data Space catalog, the Dutch DMI. Data spaces are covered below and in the chapter on trends.

What kinds there are

Four different models live under the single word "marketplace".

- **Government and sector platforms.** The far end of the scale is data.europa.eu: a public infrastructure without fees or deals, so not a marketplace but what one is compared with. The Shanghai Data Exchange conducts paid deals between companies. Catena-X in the automotive industry combines non-profit governance with paid onboarding services. The British Rail Data Marketplace, opened in November 2023, is run by the Rail Delivery Group: in its first year it gained more than 140 products, 1,200 organisations and 4,000 data sharing agreements, and in October 2026 it lists 269 products, free and paid. The platform is less about selling than about

managing usage agreements: at launch the organisers promised that most data would be free. Here the government or the sector also promotes the platform and may be the regulator. That is why data appears on it that is available nowhere else.

- **Data inside a professional product.** Bloomberg and LSEG sell not datasets but a bundle of data, identifiers, licences and a trader's workstation. This is a way to embed data in a product rather than a separate market.
- **Cloud provider platforms.** AWS, Snowflake, Google, Databricks. They differ in whether they take payment themselves and how they deliver data. The common logic is the same: the platform increases consumption of the main cloud platform.
- **Vendor showcases.** Datarade (more than 2,000 vendors on its home page in October 2026) helps find a vendor and pass a request to them, while the contract, payment and delivery remain between the parties. This is a catalog that brings in buyers, not an exchange.

Standards

Product description. There is no common standard; each platform keeps its own card. The closest thing to a common language is the description of a data product: owner, terms, delivery methods, pricing plans. Templates such as ODPS are covered in the chapter "Units of Publication and Formats".

Terms of use. Where access is negotiated between organisations, terms are described in machine-readable form: ODRL is a W3C language for rules such as "may be used for research, may not be passed to third parties".

The data space protocol. The Dataspace Protocol (version 2025-1, approved by the Eclipse Foundation on 23 July 2025) connects these parts. A participant publishes a catalog of their datasets in DCAT, attaches terms in ODRL to each, and then the protocol describes contract negotiation and data transfer. The open-source implementation is the Eclipse Dataspace Components connector, on which Catena-X exchange is built. An important detail: the dataset description here is ordinary DCAT. That means it can also be shown outside, even if the data itself is closed.

Delivery without download. Cloud platforms often do not hand over a file but give the buyer access to a table directly in the vendor's storage. AWS Data Exchange knows five kinds of delivery: files, interfaces, access to Redshift tables, direct access to S3 storage and permissions in Lake Formation. Databricks Marketplace works on the open Delta Sharing protocol, which on 10 June 2026 became OpenSharing and moved to the Linux Foundation.

Platforms

In the registry 55 of 56 marketplaces are custom builds, and this is an honest picture: there is almost no reusable marketplace software. Cloud platforms are part of the cloud, and data exchanges have their own systems. Open catalog platforms (CKAN, Dataverse, InvenioRDM) do not accept payments. For data spaces there is the open-source EDC connector, but it is an exchange channel between participants, not a showcase with payment.

The boundary with open data portals is blurring. On 30 September 2025 OpenDataSoft renamed itself Huwise and began selling the same platform as a data product marketplace, including for exchange within an organisation.

Why platforms rarely take off

Owners of data that is genuinely in demand do not need an intermediary: customers come to them anyway, on their terms. Owners fear that the intermediary will use or resell the data. And a "platform" is less useful for data than for goods: data is bought rarely, often as a one-off purchase or a long subscription, and it is more convenient to negotiate directly with the vendor.

Working platforms are usually run by the main data owner or by someone who already has buyers. Exchanges sell their own quotes. Rail Data Marketplace is run by the railway industry organisation. Cloud platforms sell where the buyer already stores data. An independent showcase of other people's data has to spend on promotion for years.

Examples

- Rail Data Marketplace: the UK railway sector platform: free and paid products, with usage agreements managed by the platform itself.
- AWS Data Exchange, Snowflake Marketplace, Databricks Marketplace: cloud platforms with delivery into the buyer's storage.
- Nasdaq Data Link: financial and alternative data from the exchange and third-party vendors.
- Datarade: a vendor showcase.
- JDEX and KYOTO DATA MARKETPLACE: Japanese platforms.
- Catena-X: the automotive data space on the Dataspace Protocol.

Where it breaks

The description is closed until registration. The buyer does not understand what they are buying, and the search engine does not see that the data exists. The working scheme is different: descriptions are open, while payment and delivery go through a separate system.

The platform is not visible from abroad. On 12 October 2026 the Korean big data platforms, their common data map and the Chinese data exchanges did not open from abroad. For a domestic market this may be acceptable, but for a foreign buyer, an aggregator and the web archive such platforms do not exist.

"Marketplace" as a sign on the door. The Korean platforms and Rail Data Marketplace largely give data away for free, while Huwise calls a platform for open data portals a marketplace. The name does not say whether anything is sold there. The type is determined by what the user gets: a file without conditions means an open data portal; a contract for each release means a marketplace.

Platforms close down. Of the Chinese data exchanges, 24 had been liquidated or were not operating by August 2025. data.world started as an open cloud catalog, and on 13 July 2026 it closed its open community; the story is covered in the chapter on cloud storage and hubs.

Nothing remains after the subscription. With delivery "without download", access to a table disappears the day the subscription ends. The fate of files the buyer managed to export is decided by the contract: in AWS Data Exchange exported files stay with the subscriber, but the vendor's contract may require their deletion. This is worth reading before buying, not after.

How to check a marketplace

1. **Open a card without logging in.** Are the description, field composition, period, update frequency and a data sample visible?
2. **Find the terms.** Are the terms of use written on the card, not only in a contract shown after payment?
3. **Find out the delivery method.** A file, an interface or access to a table in the cloud, and can you move the data to your own systems?
4. **Ask what will remain.** What may you keep after the subscription ends, and may you cite the data in publications?

When it is needed

A marketplace is needed when data is actually sold or provided under contract and each release needs terms. If the goal is to open data, a marketplace is superfluous: an open data portal or publishing without a portal is enough. It is almost never worth building your own platform for other people's data; see why in the decision record. If data is sold through a cloud platform, its description should also be kept open in an ordinary catalog: then a buyer will find it through search, not only inside the cloud.

Dataset lists

What it is. A page, table or repository with links to files or to other catalogs. A list has no search, filters or cards: a list row is a link and one sentence of description. The registry has 107 such lists, about 14% with programmatic access.

Who builds them

There is no main owner here: 20% of lists belong to companies, 19% to central government, 14% to municipalities, 13% to communities, 12% to universities. 89% are custom builds, 16 are on GitHub or GitLab. The leaders by country are the US (20) and international projects (12). Five records are attributed to the Seychelles: these are crypto exchanges registered there that publish their trading history.

What kinds there are

- **A community's thematic selection.** awesome-public-datasets on GitHub has about 79,000 stars in October 2026. Next to it are country lists: awesome-nigeria-datasets, open-data-turkey. The latter in 2023 was still the most complete list of portals in Turkey: there is no national portal, and city portals and GIS are scattered.

- **An "Open data" page on an agency's or city's website.** Cardiff, Halle, Arnsberg, Łódź, Riyadh, the Saudi tax authority. Files sit on pages of an ordinary CMS (TYPO3, Drupal, SharePoint) without a separate portal.
- **A single publisher's download page.** New York City taxi trips: one Parquet file per month and field dictionaries. Wikipedia dumps, OpenStreetMap extracts by region, transit schedule files. 13 of the 107 records are crypto exchange trading histories. For example, data.binance.vision is a cloud storage listing where each archive is accompanied by a checksum file.
- **A list built from description files.** Registry of Open Data on AWS is a website generated from 1,238 YAML files in a GitHub repository (12 October 2026). Each file has the same fields: title, description, licence, update frequency, contact, storage address. awesome-public-datasets itself works the same way: its README is assembled automatically from YAML files in the neighbouring apd-core repository.
- **Personal indexes on GitHub Pages.** Catalogs of clinical data, Indian geodata, Hong Kong open data, kept by one person. Useful as a starting point, but they have no owner who will answer for completeness.

Standards

Most often none. The following help bring a list to a machine-readable form:

- schema.org markup: DataCatalog for the page and Dataset for each row;
- a `datapackage.json` file next to the data; its template is among the book's templates;
- the list itself in CSV, JSON or YAML in a repository from which the page is generated, as with the AWS registry;
- checksum files next to archives, as at Binance;
- a DCAT file listing the datasets: this is how a static site becomes a catalog that harvesters read. How JKAN and other generators do this is covered in the chapter on static sites.

Platforms

An ordinary web page on a CMS, a README file in a Git repository, a static site on GitHub Pages, a shared spreadsheet or simply a cloud storage listing. When there are a few dozen datasets, the next step is not CKAN but a static catalog: the same files, but each dataset gets its own page with a description and markup.

Examples

- awesome-public-datasets: a thematic list of open datasets with broken-link flags.
- Registry of Open Data on AWS: a list of datasets in cloud storage, built from description files.
- TLC Trip Record Data: a download page used for years: one publisher, a clear series of files, field dictionaries.
- Wikimedia Downloads: dumps of all Wikimedia projects.

Where it breaks

Links die. This is the main disease of lists, and awesome-public-datasets fights it: each row has a status, and on 12 October 2026, 96 of 1,062 records were flagged as needing repair. A sample check of 300 random links from it the same day: 258 opened, 16 returned an error or did not respond, and another 26 refused a program (codes 401, 403, 429). The latter are probably alive for a person, but a harvesting robot can no longer see them.

No licence and no date. A list row says the data exists, but not whether it can be used or when it was last updated.

The list duplicates other catalogs. A selection of links to portals and datasets that are already in catalogs goes out of date faster than they do: a catalog is updated by the data owner, a list by whoever found the time.

The list depends on one person. A personal index on GitHub Pages or a page on a city website stops being updated as soon as the author leaves. An "Open data" page on an agency website has all its addresses broken at once when the website is redesigned.

How to turn a list into a findable publication

1. **Each row has a licence, update date, format and size.** Without this the list can only be browsed, not used.
2. **The list itself is data.** Keep it as a CSV, JSON or YAML file in a repository and generate the page from it. Then the list can be downloaded, checked by a program and harvested into an aggregator.
3. **A robot checks the links.** Scheduled link checking is easy to add to a repository, for example with the lychee action in GitHub Actions. Flag broken rows, as in awesome-public-datasets, rather than silently deleting them.
4. **Markup and registration.** Add schema.org to the page and register the list in your country's or topic's catalog.

This path is covered in detail in the chapters "Publishing Without a Portal" and "Making a Publication Findable", and a summary list of steps is in the checklists.

When it is needed

A list is a normal way to publish a dozen datasets or to compile an index on a topic. It becomes a dead end when there are no licences, persistent addresses and markup. There is no need to set up CKAN for such a list: from a list to a static catalog is one step, and it is cheaper than any portal.

How they differ from their neighbours

Type	Neighbour	Main difference
API catalog	Open data portal	describes services, not datasets; for a portal, an API is a way to get a dataset, for an API catalog the interface is the subject of description
API catalog	Metadata catalog	a registry of government system interfaces describes who exchanges what; an API catalog gives a developer access

Type	Neighbour	Main difference
Marketplace	Open data portal	a contract and terms for each release; an open data portal provides data without requests
Marketplace	Data space	a data space is the rules and protocol for exchange between participants; a marketplace is a showcase that can run on top of it
Marketplace	ML catalog	training datasets are sold under contract; hubs such as Hugging Face distribute them under open licences
Dataset list	Open data portal	no cards, search or API; a row is a link and a sentence
Dataset list	Catalog of catalogs	lists datasets, not portals; awesome selections often mix the two
Dataset list	Aggregator and dataset search engine	a list is compiled by hand, an aggregator by a program from other catalogs

A summary table of all types is in the chapter Comparing Types and Hybrids.

Sources

- dataportals-registry export, release 1.27.0 of 15 October 2026: number of catalogs, owners, countries, platforms, shares with an API and with restricted access, transport API catalogs, crypto exchange lists. The queries are in the appendix.
- OpenAPI 3.2.0, 19 September 2025; AsyncAPI 3.1.0, 31 January 2026; RFC 9727: api-catalog, June 2025; DCAT 3: data service; the Dutch API Design Rules
- Responses at `/.well-known/api-catalog` for eleven API catalogs and the HTML of the Bank of Finland API list page on Azure API Management, checked 12 October 2026
- Data services on data.gouv.fr: the API section and the section's API, export of all 1,265 cards, 12 October 2026; api.gouv.fr on beta.gouv.fr: transfer to data.gouv.fr and retirement in 2025; DILA on the transfer of cards, 21 November 2024; the transfer issue in the data.gouv.fr repository
- Norway's API catalog and the national catalog's data service search; the api.gov.uk catalog table; APIs.guru metrics; checked 12 October 2026
- WSO2 API Manager; linking an access service to a resource in ckanext-dcat
- The official MCP registry, 8 September 2025
- ODRL Information Model; Dataspace Protocol 2025-1 and release 1.0.0 at the Eclipse Foundation, 23 July 2025; Eclipse Dataspace Components
- Delivery types in AWS Data Exchange and terms after the subscription ends; Databricks on Delta Sharing becoming OpenSharing, 10 June 2026
- Rail Data Council on Rail Data Marketplace and the product catalog, checked 12 October 2026
- Korean big data platforms: launch of ten platforms (in Korean), ZDNet Korea, 22 July 2019; six new platforms (in Korean), korea.kr, 22 February 2021; 16 platforms and 6,842 kinds of data (in Korean), IT Chosun, 29 March 2022
- Chinese data exchanges: opening of the Shanghai exchange, Xinhua, 25 November 2021; the Guiyang exchange as the first, 2015 (in Chinese); a review of platforms as of August 2025 (in Chinese), CWW
- Announcement of OpenDataSoft's renaming to Huwise, 30 September 2025

- awesome-public-datasets and apd-core: record statuses and a sample check of 300 links, 12 October 2026; Registry of Open Data on AWS and its description files; TLC Trip Record Data; Binance Data Collection; checked 12 October 2026
- I. Begtin, On data markets and exchanges (in Russian), September 2026: four platform models and reasons for failure
- I. Begtin, Telegram (in Russian): Registry of Open Data on AWS, 18 March 2021; government API catalogs, 4 February 2022; why there is no api.gov.ru, 2 February 2022; apistore.mos.ru, 19 November 2021; bund.dev, 24 August 2022; API catalogs as a product for developers, 10 February 2023; India Urban Data Exchange, 17 April 2024; company data catalogs, 17 April 2023; open data in Turkey, 7 February 2023
- When You Do Not Need a Portal

Comparing Types and Hybrids

Catalog types differ not by name but by their main unit, their description standard and what a catalog cannot do its job without. That is why a catalog should be compared with catalogs of its own type, and a platform chosen for the main product.

A hybrid is a website or organisation that runs catalogs of several types at once. This is common: roughly one in eight central agencies and one in eight international organisations run catalogs of two or more types. But only a handful use one piece of software for all types. The practical conclusion of the chapter: name the type of each part by what people open it for, choose a platform for each part, and decide separately how to connect them: shared navigation, shared search or harvesting descriptions into one catalog.

The figures in the chapter come from the dataportals-registry export (release 1.27.0 of 15 October 2026), which contains 43,614 catalogs. The queries are in the appendix "Using the Registry". Live websites were checked on 12 October 2026.

Types in numbers

Type	Catalogs	Most often run by	Custom builds	Have an API	Restricted access	Dead	Main platforms
Geoportal	24,752	local government, 58%	4%	70%	1%	2.0%	ArcGIS Server, ArcGIS Hub, e-mapa, GeoServer
Research repository	8,404	universities and research organisations, 85%	34%	63%	6%	1.4%	DSpace, Pure, IPT, Figshare
Open data portal	5,967	local government, 48%	19%	69%	1%	4.1%	CKAN, OpenDataSoft (Huwise), Gosweb, Socrata
Indicator catalog	3,405	central government, 46%	44%	37%	4%	0.9%	Swing, WordPress, Drupal, Knoema
Microdata catalog	326	central government, 55%	13%	70%	49%	18.1%	NADA, REDATAM, OBiBa Mica, Nesstar
Metadata catalog	172	universities and research organisations, 31%	38%	67%	10%	0%	FAIR Data Point, Metadata Browser, Skosmos
Dataset search engine	124	universities and research organisations, 68%	28%	55%	1%	4.8%	OpenAIRE, VIVO
API catalog	115	central government, 45%	74%	51%	14%	0%	Azure API Management
Dataset list	107	companies, 20%	89%	14%	1%	6.5%	site generators, CMS
ML catalog	100	universities and research organisations, 37%	78%	23%	5%	2.0%	site generators, competition platforms
Marketplace	56	companies, 70%	98%	21%	91%	0%	custom builds for almost all

"Restricted access" is the share of catalogs where at least part of the data is provided after registration, on application or for a fee. "Dead" means catalogs that do not open or have been officially decommissioned. Knoema appears in the table according to the registry records: the service itself has not opened since January 2026, as covered in the chapter on indicator catalogs.

What the table shows:

- **Custom builds and APIs go in opposite directions.** Where a platform is widely accepted (geoportals, open data portals, microdata), two thirds or more of catalogs have an API. Where everyone writes their own (marketplaces, ML catalogs, lists), a fifth or fewer do. A custom build rarely gets as far as a programming interface: it has to be not only written but also maintained.
- **Restricted access is a feature of the type, not a sign of trouble.** Marketplaces almost always have it, and half of microdata catalogs: surveys contain information about people. For an open data portal, closed access is a reason to investigate.
- **The share of dead microdata catalogs is a trace of abandoned installations.** 47 of the 59 dead ones are old NADA catalogs of statistical offices that were installed as part of support programmes and stopped being maintained. The analysis is in the chapter on microdata catalogs.
- **A specialised platform hardly ever goes beyond its type.** CKAN means 1,906 open data portals and only 39 research repositories. GeoNode (770), Dataverse (257) and PxWeb (110) each appear in exactly one type, NADA and DSpace with one exception. Only WordPress spans nine types out of eleven, Drupal seven, Liferay and TYPO3 six each: a general-purpose CMS says nothing about the type.

How the types differ in substance

The numbers show how the market is organised. The substantive difference lies in what a catalog publishes and what it is obliged to preserve.

Type	Main unit	Description standard	Cannot work without	Must not lose	Passes descriptions to
Open data portal	a dataset: a file or table	DCAT	DCAT description export, open licence	the dataset address and licence	the higher-level portal, general search engines
Geoportal	a layer and a map service	ISO 19115, CSW, OGC API Records	a catalog service and the layer services themselves	the ability to download, not just view	the national spatial data infrastructure
Research repository	a dataset version with a DOI	DataCite	file storage, DOIs, versions, citation	the version that was cited	DataCite, research aggregators via OAI-PMH
Indicator catalog	an indicator and a time series	SDMX	a series API, methodology and units of measure	the series methodology and its persistent address	international organisations via SDMX
Microdata catalog	a survey and its files	DDI	access levels, variable descriptions	access conditions and the codebook	social data archives

Type	Main unit	Description standard	Cannot work without	Must not lose	Passes descriptions to
ML catalog	a dataset for training and evaluating models	Croissant, dataset card	splits, licence, card	the version and split on which models were measured	platforms and search engines via markup
Marketplace	a product: a dataset or subscription	none common	contract, payment, delivery	contract terms	nobody: the catalog is closed to harvesting
Search engine or aggregator	a record about someone else's dataset	whatever it harvests	harvesting via protocols, deduplication	the link to the original source	people; sometimes the next aggregator
Metadata catalog	a vocabulary, schema, catalog description	SKOS, SDMX, ISO/IEC 11179	persistent concept addresses, versions	old versions of concepts	systems that refer to it
API catalog	a programming interface	OpenAPI	interface description, access conditions	the interface's version history	the open data portal, general API catalogs
Dataset list	a link to a file	most often none	working links	—	usually nobody

Two conclusions from the table. First, what neighbouring types "cannot work without" hardly overlaps. A portal without DCAT loses federation, a repository without DOIs loses citation, and an indicator catalog without a series API turns into a table builder in the browser. That is why a platform good for one type is rarely good for its neighbour. Second, descriptions of different types go into different chains. A portal passes DCAT to the national portal, a geoportal passes ISO 19115 to the spatial data infrastructure, a repository passes to DataCite. If a website does two jobs, it has two chains, and both have to be maintained. Which capabilities are mandatory for which scenario is covered in detail in the chapter "Capability Map".

Hybrids

The book uses "hybrid" for two different cases. The first is **one website** on which two products live side by side: datasets and layers, datasets and API descriptions. The second is **one organisation** running several catalogs of different types: a statistical office with indicators and microdata, an international organisation with a portal, a geoportal and a survey library. The most developed version of the second case is a data hub: several catalogs under one banner with a shared search.

A hybrid is not a new type. Each part of a hybrid is described and evaluated by its own type: a statistical office's indicators are compared with other offices' indicators, not with its own microdata catalog.

How many organisations run catalogs of different types

The registry has 32,595 owners, if an owner is counted as an organisation with one name in one country. 2,203 of them have catalogs of two or more types, 254 of three or more. These owners account for 7,358 catalogs, or every sixth catalog in the registry. The count by name is approximate: the same agency can be recorded in different ways, so the real number of "multi-type" owners is probably higher.

The larger and more centralised the organisation, the more often it has several types:

Owner	Share of owners with catalogs of two or more types
Central government	12%
International organisations	12%
Regional government	10%
Universities and research organisations	7%
Local government	5%
Companies	2%

Pairs of types recur. Among owners with exactly two types:

Pair of types	Owners	Who it usually is	Common software combinations
Geoportal and open data portal	953	cities and regions	ArcGIS Server and CKAN, GeoServer and CKAN, ArcGIS Hub and Socrata
Geoportal and research repository	344	universities and research institutes	ArcGIS Server or GeoServer next to DSpace, THREDDS, ERDDAP
Geoportal and indicator catalog	227	statistical offices, regions	a map server next to an indicator database
Open data portal and research repository	86	science agencies, universities	CKAN next to Dataverse or DSpace
Indicator catalog and open data portal	80	statistical offices, regions	—
Indicator catalog and microdata catalog	72	statistical offices	an indicator database and NADA or REDATAM

1,127 organisations run a geoportal and an open data portal together, if those with a third type are also counted. 56% of them are local government, 22% regional. The most such pairs are in the US (174), Italy (90), Slovakia and Spain (58 each).

One piece of software for two types is rare

Of 2,203 owners with several types, 118 manage all their catalogs with one piece of software, with a custom build counted here as one piece of software. The "geoportal and open data portal" pair shows this most clearly. Of the 953 owners who have only these two types, four have the same ready-made platform on both sides: three ArcGIS Hub, one WordPress. Another 39 have custom builds for both parts. The rest run two different programs: a map server for layers and a portal platform for datasets.

This is not carelessness but a consequence of the table in the previous section. A geoportal needs a catalog service and layer services; a portal needs DCAT and file download. Few programs do both equally well, and owners choose a specialised program for each task.

Five kinds of hybrid

A second product as an add-on

The most common and simplest case. There is one main product, and the second complements it.

- **An open data portal with a map** remains a portal if the map shows tables that are already published. It becomes a geoportal if without WMS and WFS services the website has nothing to show. In the registry, 43 open data portals have map layers marked as content.
- **A geoportal with tables** remains a geoportal if the tables are layer attributes or a couple of files next to the map.
- **A research repository with a map** remains a repository if the map is a way to find a dataset by the place of observation, not a product.

The rule: the type is determined by what gets its own address and its own card. A map, chart or preview without its own address is a feature, not a second catalog. Such a hybrid is solved with one platform and one extension: for example, CKAN with ckanext-spatial for search by territory and ckanext-geoview for layer previews. If maps become the main product, an extension is no longer enough: the chapter on CKAN covers this as a separate build with a caveat on when it is more honest to take a geoplatform.

One platform, several kinds of records

The platform was designed from the start to keep records of several kinds, and each kind has its own address and its own card. This is a true hybrid in one piece of software.

- **data.norge.no**, Norway's national catalog, keeps five catalogs under one banner: datasets, APIs, concepts, information models, and public services and events. As of 12 October 2026 it has 9,066 datasets, 1,071 APIs, 6,383 concepts and 1,393 services. Only 1,644 of the 9,066 datasets are marked as open data. The rest are descriptions of data the state has but which are not necessarily available to all: in effect, a metadata catalog about government data. It runs on its own development, Felles datakatalog, with open code under the Apache 2.0 licence, and harvests descriptions from organisations' catalogs in the DCAT-AP-NO profile. By type it is an open data portal, an API catalog and a metadata catalog at once. The registry counts it twice, as a portal and as an API catalog.
- **data.gouv.fr**: 74,376 datasets, 1,265 data services (as API descriptions are called here) and 507 harvest sources. The former API catalog api.gouv.fr has been merged into the portal, as covered in the chapter on API catalogs. Data services and harvesting descriptions from CKAN and DCAT are in the core of its software, udata (release 17.12.1 of 9 October 2026).
- **GeoNode** keeps layers, maps, documents, map stories (GeoStories) and dashboards in its core, and also harvests resources from remote map services, according to the documentation of release 5.1.0 of 18 August 2026. It is a geoportal with a small document archive inside.
- **NADA**, according to its project description, can describe not only surveys but also indicators, geodata, documents and images, in DDI Codebook, Dublin Core, ISO 19115 and 19139, IPTC. But a platform capability does not in itself make a catalog a hybrid. In the World Bank Microdata Library all 7,145 records are surveys, with no other kinds. The catalog remains a microdata catalog, and the bank publishes indicators and geodata on other platforms.

- **ArcGIS Hub** is a geoportal that behaves like an open data portal: each layer can be downloaded as a file, and the catalog is served in DCAT-US and DCAT-AP. The registry records Hub as a geoportal 3,464 times and as an open data portal 19 times. More in the chapter on geoportals.

Such a hybrid works when the kinds of records are linked to each other. In Norway a dataset refers to the concepts that describe its fields and to the API through which it is obtained. On data.gouv.fr a data service refers to the dataset whose rows it serves. If the kinds of records are not linked, one piece of software for them is merely a saving on maintenance, justified only when the software handles each kind no worse than a specialised one.

A portal that also harvests

The portal publishes its own datasets and at the same time harvests descriptions from other catalogs. data.gouv.fr accepts publications directly and harvests descriptions from 507 sources. data.europa.eu harvests national portals, and since 1 July 2026 the address of the closed INSPIRE geoportal also leads to it. For the user there is no difference: they search across everything at once. For the owner these are two different obligations: they are responsible for their own datasets, and for the harvested ones only for not showing dead records and not multiplying duplicates. How an aggregator differs from a search engine and how harvesting works is covered in the chapters on search engines and aggregators and on federation and harvesting.

Several catalogs of one owner

An organisation runs catalogs of different types on different software, and only the website navigation connects them. Most statistical offices are organised this way. Mexico's INEGI has 22 records of four types in the registry: an indicator bank and interactive tables, tables on PxWeb, an SDMX service, a microdata catalog on NADA (825 studies as of 12 October 2026) and REDATAM for census calculations, a map portal and an open data page.

This is a normal setup if the website says plainly what is where. An "indicator" and a "survey file" are different shelves, and the user needs to be told which one to go to. Why two catalogs are better than one "universal portal" for a statistical office is worked through in the statistical office example.

Hub

A hub is several catalogs of different types under one banner, connected by shared navigation and often a shared search. The portals of international organisations are organised this way.

- **Pacific Data Hub** of the Pacific Community has, according to the registry export, more than 30 records of five types: a dataset portal on CKAN, microdata on NADA, indicators on .Stat Suite with an SDMX service, geoportals on GeoNode and a map showcase on TerriaJS, THREDDS servers with oceanographic data, a STAC imagery catalog of the Digital Earth Pacific programme and PopGIS demographic geoportals for fifteen countries.
- **The World Bank** keeps separate indicators, a microdata library and a data catalog that harvests descriptions from the others: as of 12 October 2026 it holds 7,900 datasets, including from the microdata library and the finance and energy platforms. The data.worldbank.org

home page says that the open data site "is expanding into Data360", and the address data360.worldbank.org leads to the same place. The registry has 28 records of five types for the bank, most of them separate indicator databases.

- **FAO**: a dataset catalog on CKAN, more than a dozen thematic maps on TerriaJS at data.apps.fao.org, sites on ArcGIS Hub and indicator databases on its own FENIX platform.

A hub does not make catalogs the same. It solves a different problem: a person who came for "data about Fiji" does not need to know that indicators are in .Stat Suite and surveys in NADA. A hub's shared search is usually built by harvesting descriptions from all catalogs into one, as the World Bank data catalog does. Then the hub gets one more harvesting chain, with all its problems.

How to determine the type of a hybrid

The type of each part of a hybrid is determined by the same questions as the type of an ordinary catalog; they are covered in the chapter on classification. For a hybrid, four checks are added.

1. **What gets its own address.** If a layer, series or API has its own card with a persistent address, it is a separate part with its own type. If not, it is a feature of the main catalog.
2. **What goes into federation.** Look at what the website serves in DCAT, CSW or OAI-PMH. If the DCAT export contains only datasets while layers go via CSW to the spatial data infrastructure, the website has two chains and, in effect, two catalogs.
3. **Which search is the main one.** If the home page has one search across everything and the results lead into different software, it is a hub. If there are several searches and each covers its own shelf, these are several catalogs side by side.
4. **Which software stands behind each part.** Different paths on one domain often mean different software: `/catalog` is CKAN, `/microdata` is NADA, `/geoserver` is a map server.

The classification of a hybrid is recorded not as one line but as a line per part, using the template from the chapter on classification. For INEGI these are four lines: an indicator catalog on a custom build, PxWeb and an SDMX service; a microdata catalog on NADA with REDATAM; a geoportal on a custom build; an open data portal. The owner, level and access are shared, but the type, standard and platform differ.

How to build a hybrid: three setups

If you need two types at once, the choice comes down to three setups.

Setup	When it fits	What it costs	Examples
One platform with extensions	the second product is small and its standard is supported by an extension or the core	the second product is weaker than in specialised software; the extension has to be checked with every upgrade	CKAN with ckanext-spatial, Dataverse for microdata, GeoNode with documents
Two specialised platforms and shared navigation	both products matter, each has its own standard and its own chain	two maintenance efforts and two addresses; the user needs to be told what is where	a statistical office: PxWeb or .Stat Suite and NADA; a city: ArcGIS Server and CKAN
Hub: several platforms and a shared search	three or more catalogs, and searching across all of them matters to the user	one more harvesting chain, duplicates, identifier matching, a central team	Pacific Data Hub, the World Bank data catalog, data.norge.no

What specialised platforms can do outside their type:

Platform	Own type	Second product	How
CKAN 2.12	open data portal	search by territory and harvesting geodata descriptions via CSW	the ckanext-spatial extension
CKAN 2.12	open data portal	WMS and GeoJSON layer previews	the ckanext-geoview extension, release 0.3.2 of 2 October 2026
CKAN 2.12	open data portal	layer services, microdata access levels, SDMX	no: a second piece of software is needed
ArcGIS Hub	geoportal	file download and DCAT export, as for an open data portal	core
GeoNode 5.1	geoportal	documents, map stories, dashboards	core
Dataverse	research repository	microdata: DDI description, restricted-access files	core
Dataverse	research repository	geodata: a coverage description block	core; the map via pluggable previewers
NADA	microdata catalog	indicators, geodata, documents	core, but without an SDMX service or layer services
udata 17	open data portal	an API catalog (data services) and harvesting other catalogs	core

The capabilities of each platform are covered in detail in the platform profiles, and summary tables are in the appendix.

Example decision record: a region with maps and datasets

Context. A region runs an open data portal on CKAN and passes descriptions to the national portal in DCAT-AP. Separately, the regional land agency runs a map server with layers and a GeoNetwork catalog that passes descriptions via CSW to the national spatial data infrastructure. Residents complain that layers cannot be found on the portal, and management proposes to "bring everything together".

Options.

1. Move the layers into CKAN as files and close the geocatalog.
2. Replace both websites with one geoplatform that also handles datasets, for example ArcGIS Hub.
3. Keep both systems. CKAN with ckanext-spatial harvests layer descriptions from GeoNetwork via CSW and shows them in the shared search with links to the services. Layers go to the national portal through one chain only.

Criteria. Do not lose obligations to the spatial data infrastructure. One search for the resident. Do not create duplicates in the national and European catalogs. How many programs to maintain.

Decision. Option 3.

Why. Option 1 breaks an obligation: the spatial data infrastructure expects descriptions in ISO 19115 and layer services, not files. Option 2 is possible, but changes both systems at once and ties the region to one vendor's cloud; three of the 953 owners of this pair in the registry do this. Most keep two programs, most often a map server and CKAN. Option 3 keeps each program in its place and gives the resident one search.

Consequences. CKAN gets one more extension that has to be checked with upgrades: ckanext-spatial passes automated tests on 2.12, as recorded in the chapter on CKAN. Harvested layer descriptions must be excluded from the DCAT export if the national portal harvests the geocatalog itself, otherwise the same layer will reach data.europa.eu twice. How such chains break and get tangled is covered in the chapter on federation.

When to revisit. If layers become the region's main product and datasets an appendix to them. If the national portal stops harvesting geocatalogs directly. If ckanext-spatial stops being released for a new version of CKAN.

Where hybrids break

Duplicates in federation. The same layer goes up twice: through the geoportal via CSW and through the open data portal via DCAT. The aggregator shows two records with different addresses, and both look like the main one. To avoid this, each part of a hybrid should have its own chain, and what has been harvested from the neighbouring part is not passed up again.

One part outweighs everything. If both parts are connected to federation, the count is determined by the one that slices data more finely. On data.europa.eu, of 850,000 German datasets, 681,000 come from a single geocatalog, almost 450,000 of them development plans for individual plots. When comparing countries or organisations, check which part of the hybrid the records came from.

Only one part is visible. The opposite problem: one part of a hybrid was connected to federation and the other forgotten. In June 2024 data.europa.eu saw 324 datasets from Estonia because it harvested only the national geoportal. After the open data portal was connected, there were 8,700.

One thing, different cards. A dataset in the portal and a layer in the geoportal describe the same data, but with different titles, licences and update dates. The user does not understand which card is correct, and the search engine sees two.

Two searches without explanation. A statistical office website with a search across indicators and a separate search across surveys, where neither mentions the existence of the other. A person looks for a survey in the indicator search, finds nothing and leaves.

A universal platform does the second product badly. CKAN with layers loaded as files provides no WMS and WFS services. An indicator catalog on a CMS serves tables as images and PDFs. "Can do this too" in a platform description does not mean it can do it the way the second type needs.

The hub is closed to programs. The whole value of a hub lies in the links between catalogs, but some hubs close them to programs. On 12 October 2026 the API of the Pacific Data Hub portal, its GeoNode and the FAO catalog on CKAN responded to programs with a Cloudflare check, and the

INEGI tables on PxWeb with a 403 error. A person in a browser sees the data, but an aggregator and a researcher with a script do not. How this affects discoverability is covered in the chapter on discoverability.

One part of the hub dies, the rest live on. Hub components age differently. The 110 owners who run both indicators and microdata have 60 indicator catalogs recorded in the registry on Knoema, a service that has not opened since January 2026. A hub has to be maintained as a set of separate programs, each with its own lifespan.

Type by a single record. Aggregators and registries record a hybrid as one type, and the type counts drift as a result. ArcGIS Hub in the registry is almost always a geoportal, data.norge.no is an open data portal and an API catalog at the same time. When comparing figures by type, remember that the boundary between them was drawn by a person, not a program.

How to check a hybrid

You can see what is actually inside with API requests. All commands were checked on 12 October 2026.

```
# data.gouv.fr: datasets, data services (APIs) and harvest sources in one program
curl -s https://www.data.gouv.fr/api/1/site/ | jq '.metrics | {datasets, dataservices,
harvesters}'

# data.norge.no: how many records are in each catalog
for t in datasets data-services concepts services; do
  printf "%s: " "$t"
  curl -s -X POST -H 'Content-Type: application/json' \
    -d '{"pagination":{"size":1,"page":0}}' \
    "https://search.api.fellesdatakatalog.digdir.no/search/$t" | jq .page.totalElements
done

# NADA: which kinds of records actually exist in the catalog
for t in survey geospatial timeseries table document; do
  printf "%s: " "$t"
  curl -s "https://microdata.worldbank.org/index.php/api/catalog/search?ps=1&type=$t" | jq
.result.found
done
```

For the World Bank Microdata Library the last command gives 7,145 surveys and zero records of other kinds. Old versions of NADA do not understand the `type` parameter and return the same number for any kind; the INEGI catalog responds this way. In that case the kinds of records are checked in the catalog itself, in the data type filter.

For CKAN it is useful to compare the DCAT export with search: if search contains layers but `/catalog.ttl` does not, the layers were harvested from the geocatalog and go up through another chain. Commands for individual types are in the "How to check" sections of the chapters on each type and in the checklists.

The rule

Name the type of each part by what people open it for. Choose the platform for that part, not the "most universal" one. Cover the second task with an extension if it is small, with a separate program if it matters, or honestly cut it back. The link between the parts (navigation, shared search or harvesting descriptions) is a separate decision with its own cost, and it is worth writing it down in a decision record, as in the example above.

Sources

- dataportals-registry export, release 1.27.0 of 15 October 2026: types, owners, platforms, shares with an API, with restricted access and of dead catalogs, owners with catalogs of several types and software combinations. The queries are in the appendix.
- data.norge.no and the Felles datakatalog search API, portal code; data.gouv.fr, udata 17.12.1; World Bank Microdata Library, World Bank data catalog, data.worldbank.org; INEGI microdata catalog; Pacific Data Hub; checked 12 October 2026.
- NADA: kinds of records and description standards; GeoNode documentation and release 5.1.0; ckanext-spatial, ckanext-geoview; checked 12 October 2026.
- I. Begtin, Public data catalogs of international organisations (in Russian), January 2023.
- How to Classify Catalogs, Capability Map, Federation and Harvesting, Worked Examples, Justifying a Decision.

PART IV.

Catalog Capabilities

Which capabilities data catalogs have, why they are needed and when you can do without them: metadata export, storage, DOIs, citation, metadata depth, deployment speed and more.

1. Capability Map
2. Metadata Export
3. Metadata Completeness and Depth
4. APIs and Bulk Download
5. Built-in Storage
6. Persistent Identifiers, DOIs and Versions
7. Citation Export
8. Access, Roles and Workflows
9. Deployment Speed and Maintenance
10. Other Capabilities

Capability Map

Portals differ not by name but by their set of capabilities. This chapter is the map for the whole part: what capabilities exist, which any catalog needs, which are switched on by the catalog type or a specific task, and on which platforms each capability is in the core, in an extension or nowhere. Details on each capability are in separate chapters.

In brief:

- **Six things are needed by any catalog**, even a static site with a dozen datasets: a persistent address, a licence for each dataset, an update date and a contact, an export of all descriptions, markup for search engines, and access to descriptions without registration.
- **Everything else depends on the catalog type.** A research repository needs DOIs and versions, a geoportal a catalog service, an indicator catalog SDMX, a microdata catalog closed files with an open description, an ML catalog versions and a Croissant description.
- **Type is not the only reason.** A higher-level portal, a funder, the law, the number of datasets and publishers make a capability mandatory for a catalog of any type.
- **A mandatory capability is better taken from the platform core.** On live catalogs it is present in most installations if it is in the core, in about a third if it comes from an extension, and in a few percent of custom-built sites.

How to use the map

1. Tick off the minimum: it is the same for everyone.
2. Find your catalog type in the table "What is needed in which cases" and write down the "mandatory" and "desirable" rows.
3. Go through the external reasons: each can raise a capability from "desirable" to "mandatory".
4. For each mandatory capability, check on which platforms it is in the core, in an extension, or absent.
5. Write down the rejections too, with a reason. How to turn the list into a requirements specification and a platform choice is covered in the chapters "Selecting Features" and "From Features to a Platform".

If the catalog type is not yet clear, start with the route map or with the classification of catalogs.

What a capability is

A catalog capability is something a catalog can do for someone: show a dataset to a person, serve descriptions to a program, issue an identifier for a paper, hide a file from outsiders. It is not a menu item or a module name. The same capability can be built in different ways: a DOI can be issued by the platform itself, by an extension, or by someone else's repository where a copy is deposited.

Each capability has an addressee, whose task it solves:

- **a person on the website**: search, preview, a clear description, a "Cite" button;
- **a program**: a higher-level portal, a dataset search engine, a harvester, a service built on the data. It needs exports, protocols and APIs;
- **the owner and editors**: roles, review before publication, statistics, the cost of maintenance.

A catalog made only for the first addressee looks good, but neither aggregators nor search engines can see it. That is why half the items in the minimum below are for programs.

Groups of capabilities

Group	What it includes	What question it answers	Chapter
Describing data	mandatory and domain fields, custom fields, table field dictionary	can you tell from the description what is inside and whether you may use it	Metadata Depth
Exchanging descriptions	DCAT, schema.org, domain formats: ISO 19115, DataCite, DDI, SDMX, Croissant	will other systems be able to read the descriptions	Metadata Export
Machine access	search API, export of all descriptions, access to part of the data, change feed	can a program take the catalog and data without a person	API and Bulk Download
Storage	files in the catalog, external links, a table in a database	is the file available as long as the catalog lives, not the source website	Storage
Citability	persistent address, DOI, versions, ready-made citation	will the link open the same dataset years later	Identifiers, Citation
Governance	organisations, roles, moderation, closed files, registration	who publishes, who reviews, who sees the file	Access and Workflows
Operations	deployment method, upgrades, extensions, people	what it costs to launch and to maintain for years	Deployment
Other	search and filters, preview, languages, harvesting other catalogs, statistics, feedback	is it convenient for people and does the catalog grow with the task	Other Capabilities

The minimum for any catalog

These capabilities are needed both by a national portal and by a research group's page. Without them a catalog is either invisible or useless.

What	Why	Chapter
A persistent address for the dataset and file	papers, reports and other catalogs link to the dataset; the address does not change when the description is edited or the website moves	Identifiers
A licence for each dataset	without it the dataset cannot be used legally, even if it can be downloaded	Licences
An update date and a contact	the date shows whether the dataset is alive; the contact gives someone to report an error to	Metadata Depth
An export of all descriptions	the higher-level portal and aggregators harvest the catalog from it; paging through search is inconvenient and unreliable for them	API and Bulk Download
schema.org markup and a sitemap	ordinary search engines and Google Dataset Search find the dataset through them	Metadata Export
Descriptions without registration or captcha	otherwise automatic harvesting does not work or breaks with every update of the protection	API and Bulk Download

The export can be simple. For a small site it is one `data.json` or `datapackage.json` file, not a full DCAT-AP profile. The same minimum is recorded as the first section of the requirements checklist.

A catalog without this minimum is a common story even among large portals. In November 2024 the Moscow portal data.mos.ru served neither schema.org markup nor DCAT, and its datasets were absent from dataset search, while small civil society portals with markup ranked first. More in the chapter on metadata export.

What is needed in which cases

"Mandatory" means the catalog will not do its job without it. "Desirable" means it helps noticeably, but the catalog is useful without it. "Not needed" means the capability is superfluous for this type or another tool covers it. "From hundreds of datasets" means that with dozens, pages and a description file are enough; a search API of your own is set up when the number of datasets grows large. The same thresholds are recorded in the requirements checklist.

Capability	Open data portal	Research repository	Geoportal	Indicator catalog	Microdata catalog	ML catalog	Small site
Export of all descriptions	mandatory	mandatory	mandatory	mandatory	mandatory	mandatory	desirable
schema.org markup	mandatory	mandatory	desirable	desirable	desirable	mandatory	mandatory
DCAT export	mandatory	desirable	desirable	desirable	desirable	not needed	desirable
Domain format	not needed	mandatory: DataCite and OAI-PMH	mandatory: ISO 19115 and CSW or OGC API Records	mandatory: SDMX or machine access to tables	mandatory: DDI	mandatory: Croissant	not needed
Search API	from hundreds of datasets	from hundreds of datasets	from hundreds of datasets	desirable	from hundreds of datasets	from hundreds of datasets	not needed
Access to part of the data without downloading	desirable	not needed	mandatory	mandatory	desirable	desirable	not needed
Files in the catalog itself	desirable	mandatory	not needed	not needed	desirable	desirable	not needed
DOI	not needed	mandatory	not needed	not needed	desirable	desirable	not needed
Versions	desirable	mandatory	not needed	desirable	desirable	mandatory	not needed
Citation export	desirable	mandatory	not needed	desirable	desirable	desirable	desirable
Closed files with an open description	not needed	desirable	not needed	not needed	mandatory	desirable	not needed

Why:

- **An open data portal** lives in the chain "city → region → country → general search engine". Without DCAT and a description export, the higher-level portal will not harvest it. A search API is needed when there are hundreds of datasets or more: a city with thirty datasets can manage with pages and a description file. Storing files itself is desirable because external links die: in a

full download of data.gov.uk in 2023, about 10% of file links did not open. A DOI is usually excessive here: the data is updated, and instead of versions with identifiers an archive of past files is enough.

- **A research repository** exists so that data can be cited in a paper. That is why DOIs, versions and a ready-made citation line are mandatory, for a small repository too if a version is cited. Files must sit in the repository itself: a link in a paper cannot depend on someone else's website. Closed files are not needed by all, but often: an embargo until the paper is published, patient data.
- **A geoportal** describes layers and services. The domain language is ISO 19115 and the CSW catalog service or OGC API Records; this is mandatory even with a small number of layers. Access to part of the data means the WMS, WFS and OGC API map services: without them a layer can only be viewed on a map. Storing files itself is not necessary: the data usually already sits in the map server.
- **An indicator catalog** serves not files but indicators with breakdowns, so the main thing is selection by year and territory. Without SDMX, or at least machine access to tables with an indicator description, the catalog remains a table builder in the browser. The description export here is replaced by a full list of tables or series with methodology. Versions are desirable: indicators are revised, and an analyst needs to know which release they relied on.
- **A microdata catalog** describes surveys and provides the records about people by access level: the DDI description is visible to all, the file on application. Storing files itself is necessary if you provide them yourself; the World Bank Microdata Library mostly points to external repositories. Access to part of the data here means online tables that calculate distributions without releasing records, as in REDATAM.
- **An ML catalog** serves datasets on which models are compared. Results are comparable only if the dataset and its split into training and test parts have not changed, so versions are mandatory. A Croissant description is required by the NeurIPS data track for every submission with a new dataset, and Kaggle, OpenML, Hugging Face and Dataverse generate it themselves. Closed files mean access on request, as with some datasets on Hugging Face.
- **A small site** with operational data can do without a search API and without DOIs. Search engines need schema.org markup, otherwise the dataset will not be found. If the data needs to be cited as an immutable version, that is already a task for a research repository: a DOI is mandatory at any size, and the easiest way to get one is a generalist repository.

What makes a capability mandatory for any type

The table above is a guideline, not a law. A regional portal that must pass metadata to the national one makes DCAT mandatory, even if its type says "desirable". Below are reasons that raise a capability to "mandatory" regardless of the catalog type.

If	Then mandatory	Chapter
A higher-level portal harvests your descriptions	DCAT in the profile adopted in the country (DCAT-AP, DCAT-US) and a stable export address	Metadata Export
The data is part of the EU high-value datasets (Regulation 2023/138)	API and bulk download, a licence no stricter than CC BY 4.0	API and Bulk Download

If	Then mandatory	Chapter
The layers are part of the country's spatial data infrastructure	ISO 19115 descriptions and a catalog service	Geoportals
A funder or journal requires a link to the data	DOI, versions, citation export	Identifiers
The dataset accompanies a paper in the NeurIPS data track	a Croissant description	ML Catalogs
There are hundreds of datasets or more	a search API, full-text search of descriptions and filters	Other Capabilities
More than one team publishes	organisations and roles	Access and Workflows
External people publish	moderation before entering search, registration by invitation	Access and Workflows
The data is personal, licensed or sensitive	closed files with an open description, an application log	Access and Workflows
The country has several official languages or the audience is international	interface and description languages	Other Capabilities
You harvest other catalogs	scheduled harvesting from DCAT, the CKAN API, CSW, OAI-PMH	Federation and Harvesting
A law or a sector code list requires an attribute not in the schema	custom description fields	Metadata Depth
The catalog must live for ten years	files stored locally, a platform with a clear upgrade policy, a closure plan	Deployment

The reverse also works: a capability from the "mandatory" column can be crossed out if someone else covers it. A geoportal does not store files itself if layers live in a map server. A research group does not set up its own DOIs if it deposits a release in Zenodo. A city does not build its own search API if a regional portal already harvests its datasets.

Where a capability lives in platforms

The same capability is in the core of one platform, appears only with an extension in another, and is absent from a third. The tables in this part use four marks:

- **core**: the capability is in the platform itself;
- **extension**: a separate module adds it, with its own author and its own risks on upgrade;
- **no**: the platform cannot do this; another platform or custom development is needed;
- **not checked**: the cell could not be filled from the October 2026 documentation.

A summary across the chapters of this part:

Capability	In the core	Via an extension or configuration	No
DCAT export	DKAN, udata, Piveau, Huwise (OpenDataSoft), Socrata, ArcGIS Hub, InvenioRDM, GeoNetwork	CKAN: ckanext-dcat; Dataverse: a third-party DCAT 3 exporter	DSpace, NADA, PxWeb, GeoNode; Dataverse has no DCAT export in its core

Capability	In the core	Via an extension or configuration	No
schema.org markup	Dataverse, InvenioRDM, GeoNetwork, NADA, Socrata; Huwise has it in the core, but the administrator enables it	CKAN: ckanext-dcat	PxWeb; udata, Piveau, ArcGIS Hub and GeoNode had no markup on the page in a check on 13 October 2026; on a static site the template author writes the block
Export of all descriptions	Dataverse, InvenioRDM, DSpace (OAI-PMH); GeoNetwork (CSW); DKAN and cloud services (DCAT); JKAN (a `data.json` file)	CKAN: ckanext-dcat	a static site, until the author builds a description file
Search API	CKAN, DKAN, cloud services, Dataverse, InvenioRDM, DSpace, GeoNetwork, NADA, PxWeb	—	JKAN, a static site
Files in the catalog	CKAN, Dataverse, InvenioRDM, DSpace, NADA, cloud services	CKAN: cloud storage instead of disk, via configuration or an extension	PxWeb and .Stat Suite: they store tables, not files
Access to part of the data	CKAN (DataStore), cloud services, ArcGIS Hub, Dataverse (tabular files), PxWeb, .Stat Suite	CKAN: loading CSV into DataStore is done by the XLoader extension	GeoNetwork (links to other people's services), a static site
DOI	Dataverse, InvenioRDM, Figshare; DSpace and GeoNetwork after being enabled in configuration. A contract with DataCite is needed everywhere	CKAN: ckanext-doi; the documentation names only CKAN 2.9 and 2.10, not tested on 2.11 and 2.12	DKAN, Huwise (OpenDataSoft), Socrata, ArcGIS Hub, PxWeb; NADA has only a field for a DOI minted separately
Versions	Dataverse, InvenioRDM, Figshare, DSpace	—	CKAN (an activity log is not a version), cloud open data portals
Citation export	Dataverse, InvenioRDM, Figshare, GeoNetwork 4.4; NADA via DataCite if the study has a DOI	CKAN: ckanext-doi (only with a DOI) or the young ckanext-citeproc	DSpace up to and including 10.1 (only meta tags for Google Scholar), cloud open data portals
Custom description fields	Dataverse, InvenioRDM, DSpace, DKAN (schema replacement); Huwise (OpenDataSoft) and Socrata: field templates in settings	CKAN: ckanext-scheming; GeoNetwork: schema plugins for country profiles	ArcGIS Hub: one style from ready-made standards per organisation; JKAN and a static site
Closed files with an open description	Dataverse, InvenioRDM, DSpace, NADA, Huwise (OpenDataSoft)	CKAN's core can close only a whole dataset, description included; fine-grained control requires extensions	a static site; Socrata hides a private dataset entirely
Harvesting other catalogs	Piveau; GeoNetwork (CSW, OAI-PMH, OGC services, STAC collections and others); Dataverse (OAI-PMH client); udata (six harvesters shipped, disabled by default); Huwise (harvesters for CKAN, Socrata, ArcGIS, data.json and others; CSW enabled on request); Socrata (Catalog Connector: data.json and public ArcGIS Server)	CKAN: ckanext-harvest, ckanext-dcat, ckanext-spatial	JKAN, a static site; ArcGIS Hub not checked in the October 2026 review

Who maintains each CKAN extension and which core versions it works with is recorded in the chapter "CKAN: Core and Extensions". A full summary by platform is in the appendix, and platform profiles in the chapter "Platform Profiles".

The "core" mark does not mean "enabled". DOIs in Dataverse, DSpace and GeoNetwork will work only after a contract with DataCite and configuration, and preview in Dataverse only after the administrator installs previewers. Check the specific site, not only the platform documentation.

What this looks like on live catalogs

The difference between core and extension is visible not only in documentation. The dataportals-registry records which interfaces were found for each catalog. Below is how many catalogs provide a capability mandatory for their type, depending on the platform. Registry release 1.27.0, all records of the type including inactive ones; research repositories are counted together with generalist ones.

Capability and catalog type	Platforms where it is in the core	Where an extension or configuration provides it	Custom builds
DCAT in open data portals	Socrata 239 of 245, OpenDataSoft 348 of 454, DKAN 103 of 161	CKAN with ckanext-dcat 633 of 1,906	24 of 1,144
DOI in research repositories	Figshare 266 of 288, Dataverse 172 of 257	DSpace, where DOIs are enabled by configuration, 218 of 829	158 of 2,840
OAI-PMH in research repositories	DSpace 665 of 829, Dataverse 177 of 257	—	19 of 2,840
CSW or OGC API Records in geoportals	ArcGIS Hub 3,257 of 3,464, GeoNetwork 686 of 762, GeoNode 628 of 770	—	4 of 1,043

What follows from this:

- If a capability is in the core, it is present in roughly two thirds of installations or more, and in the cloud Socrata and ArcGIS Hub almost in all: there it does not need to be installed or enabled.
- If it has to be installed separately or enabled by configuration, it is present in a third or a quarter: ckanext-dcat in CKAN, DOIs in DSpace. Dataverse has DOIs in its core, but the platform can also issue other identifiers (Handle and the installation's own permalinks), so DOIs are marked for two thirds of sites.
- Custom builds almost never have the mandatory capability: 2% of custom open data portals serve DCAT, and fewer than 1% of custom repositories OAI-PMH.
- A map server is not yet a catalog. Of 4,674 geoportals on ArcGIS Server, a catalog service was found on one, and of 1,927 municipal e-mapa portals on none: they show layers but serve no descriptions for harvesting.

Across all catalogs the picture is more modest than the table "What is needed in which cases" demands. DCAT was found in 26% of open data portals, OAI-PMH in 25% of research repositories, DOIs in 17%, a catalog service in 20% of geoportals, SDMX in 5.2% of indicator catalogs. Of these 177 indicator catalogs, 113 are Knoema showcases, and Knoema itself has not opened since January 2026 (more in the chapter on indicator catalogs). What the map calls mandatory is in fact present in a minority, and the reason is usually the platform: neither the core nor an installed extension provided the capability.

The numbers are a lower bound: the registry marks an interface only if it found it at known addresses. The registry hardly checks schema.org markup, so it is not in the table. The queries are in the appendix "Using the Registry".

How to cover capabilities without a portal

Many capabilities on the map do not require a portal. A static site, a Git repository, a generalist research repository such as Zenodo and a cloud hub cover the whole minimum and part of the rest.

Capability	How to cover it without a portal	What you will not get
Persistent address	your own domain with an unchanging address structure; a release DOI in Zenodo	—
Licence	a `LICENSE` file and the licence field in `datapackage.json`	—
Export of all descriptions	a `data.json` or `datapackage.json` file next to the pages; a generator such as JKAN builds it automatically	harvesting protocols such as OAI-PMH and CSW
schema.org markup	a JSON-LD block in the dataset page template	field validation: the template author takes care of it
Search API	not needed with dozens of datasets	search for programs
Access to part of the data	Datasette, a program that publishes an SQLite database with a web page and a JSON API in one command	pure static hosting: Datasette runs as a small server
Files	next to the pages, as release attachments, in cloud storage	large files in Git: GitHub does not accept files larger than 100 MiB
DOI and versions	release tags in the repository and automatic archiving of releases in Zenodo	—
Citation export	a `CITATION.cff` file: GitHub shows a button to cite the repository based on it	—
Roles and review	repository permissions and review of changes before merging	moderation of external publishers
Closed files	—	this way you publish only open data
Harvesting other catalogs	—	you will not have it

If the "What is needed" table requires only the minimum for your case, you probably do not need a portal. When this happens and how to make such a publication findable is covered in the part "Publishing Without a Portal", especially in the chapters on Git repositories and on discoverability.

How to quickly check a live portal

A few requests in ten minutes show which capabilities from the map a particular site has. Substitute the portal address and the address of any dataset page:

```
P=https://portal.example

# Export of all descriptions: DCAT-US and DCAT in RDF
curl -sI "$P/data.json" | head -1
curl -sI "$P/catalog.ttl" | head -1

# Sitemap
curl -sI "$P/sitemap.xml" | head -1
```

```
# schema.org markup on a dataset page: number of JSON-LD blocks
curl -s "$P/dataset/some-dataset" | grep -c 'application/ld+json'

# CKAN search API and the list of installed extensions
curl -s "$P/api/3/action/package_search?rows=0"
curl -s "$P/api/3/action/status_show"

# Research repository and geoportal
curl -sI "$P/oai?verb=Identify" | head -1
curl -sI "$P/csw?service=CSW&request=GetCapabilities" | head -1
```

A `200` response is a good sign; `404` at all export addresses is a reason to look for it in the portal's documentation or to assume it does not exist. Then three checks by eye: do five random datasets have a licence and an update date; do the files open; is a DOI and a "How to cite" block visible, if the portal promises them. Detailed checks are in the "How to check a live portal" section of each chapter in this part and in the summary checklists.

Common mistakes

- **"Mandatory" for everything.** If everything is mandatory, comparing platforms becomes meaningless, and the project cost grows because of capabilities nobody uses.
- **A list from someone else's specification.** A national portal needs harvesting of other catalogs, roles for a hundred agencies and a showcase of use cases. A city with thirty datasets does not.
- **A showcase instead of the minimum.** Search with filters and charts on the home page are built before the description export and markup. As a result, people use the site, but programs cannot see it.
- **Preview instead of an API.** A table on the page helps a person, but it cannot be cited and cannot be reliably retrieved by a program.
- **A DOI for operational data.** A transit schedule that changes every day needs a persistent address and an update date, not a new DOI for every export.
- **An extension without checking.** "There is an extension" does not mean "it works on your version". Before choosing, check who maintains it and on which core versions tests run; the method is in the chapter "Extension Ecosystems".

How the chapters of this part are organised

Each chapter about a capability is organised the same way: what it is, why it is needed, what variants exist, when it is needed and when not, how it is done in platforms, how to check a live portal. The platform tables use the same four marks as above: core, extension, no, not checked.

Sources

- The chapters of this part and the platform documentation they refer to.
- The guidelines by type were checked against how real catalogs are organised in the dataportals-registry. dataportals-registry export, release 1.27.0 of 15 October 2026: shares of interfaces and DOIs by type and platform, queries of 15 October 2026 to this export, given in

the appendix.

- DCAT 3, W3C Recommendation, schema.org/Dataset
- Commission Implementing Regulation (EU) 2023/138 on high-value datasets: API, bulk download and licence
- NeurIPS 2026: Evaluations & Datasets track rules: Croissant for every submission to this track that introduces a new dataset
- Harvesting other catalogs in GeoNetwork 4.4: CSW, OAI-PMH, OGC services, STAC collections, checked 12 October 2026
- I. Begtin, Why data on Russia is so hard to find (in Russian), November 2024: data.mos.ru and dataset search; Good and bad data publishing practices (in Russian), November 2024: the share of broken links on data.gov.uk

Metadata Export

A catalog is useful not only to people on its website. Other portals, search engines and programs must be able to read the dataset descriptions. For this the catalog serves metadata in a format these systems already understand.

The main conclusion of the chapter: responding at `/data.json` does not yet mean serving descriptions. An export has a profile, and profiles are not interchangeable: data.europa.eu expects DCAT-AP, data.gov DCAT-US, Google markup on the dataset page. A platform usually supports several formats at once, but the required profile has to be enabled in configuration, and the description fields must be filled in the way that particular profile requires. The standards themselves are covered in the chapter "Metadata Standards". This chapter is about a platform's ability to serve them.

What it is

Metadata export is a ready-made description of a dataset or of the whole catalog in a format that someone else's system reads. It has three addressees, each with its own format.

Other catalogs. They need a common description vocabulary. For open data this is DCAT, a W3C Recommendation and the vocabulary of data catalogs. Using it, a national portal harvests regional ones, and the common European portal harvests national ones. DCAT itself defines only the meaning of fields. Which of them are mandatory and what words to use is decided by a profile, a refinement of the vocabulary for a country or sector:

- **DCAT-AP** is the profile for Europe. The current version is 3.0.1, a SEMIC recommendation of 27 October 2025. The 3.0.2 draft of 17 June 2026 has not yet become a recommendation. On top of it there are sector refinements: GeoDCAT-AP for geodata, StatDCAT-AP for statistics, DCAT-AP HVD for high-value datasets, HealthDCAT-AP for health data;
- **DCAT-US** is the profile for US federal agencies. An agency publishes one `data.json` file, and data.gov harvests it. The new version is DCAT-US 3.0, built on DCAT 3. data.gov still accepts the old version 1.1 during a transition period.

Search engines. They need markup right in the dataset's HTML page: an `application/ld+json` block with the schema.org vocabulary and the Dataset type. Google Dataset Search reads it. It is not a replacement for DCAT: the markup has a different audience, and Google states plainly that it understands both schema.org and DCAT, but looks for markup on the dataset page.

Sector systems. They have their own formats and protocols:

- geodata: ISO 19115 and the CSW catalog service, as well as OGC API Records;
- research: the DataCite schema and the OAI-PMH protocol;
- microdata: DDI;
- statistics: SDMX;
- machine learning: Croissant on top of schema.org.

A protocol and a format are different things, and they are often confused. CSW and OAI-PMH answer the question "how to ask the catalog". DCAT, ISO 19115 and DataCite answer the question "in what language the answer will come". The same CSW can answer both with an ISO 19115 record and with a DCAT description, if the catalog software supports it.

Why it is needed

Without a standard export, a catalog is visible only to those who open its website. Here is who is left out.

The higher-level portal. It harvests descriptions according to the profile adopted in the country and at known addresses. A catalog with "almost DCAT" or its own JSON will not be harvested, even if the set of fields is similar. How such chains work and where they break is covered in the chapter on federation and harvesting.

Dataset search engines. In November 2024 Google Dataset Search, for Russian-language queries, ranked first datasets from hubofdata.ru and ngodata.ru, two small civil society portals on CKAN. Not because they are the most complete, but because their pages have schema.org markup. The Moscow open data portal data.mos.ru, large and working, served neither schema.org nor DCAT, and its datasets were absent from both Google Dataset Search and the Dateno search engine. In 2022 the Indian portal data.gov.in dropped out in the same way: ordinary Google search indexed it, dataset search did not. Markup matters more than the size of the catalog.

Aggregators and archives. They need a complete export that can be repeated: one file or a protocol with selection by date. A sitemap is not suitable for this: it contains only page addresses, without licences and files. How a description export differs from a search API is covered in the chapter on APIs and bulk download.

The law. For public bodies in the EU, the DCAT-AP profile is a practical requirement for exchange with data.europa.eu. For high-value datasets under Regulation 2023/138, the DCAT-AP HVD profile is added on top.

An export does not fix empty fields. If a description has no licence and no publisher, DCAT will not have them either: the software carries over what is filled in. Which fields profiles require and how to check that they are filled in meaningfully is covered in the chapter on metadata depth.

Implementation options

Descriptions are served in five ways. They are not mutually exclusive: a mature catalog needs several at once.

Method	What the consumer gets	Strength	Weak spot
Markup on the page	schema.org in the HTML of each dataset	a search engine reads it while crawling the site	only one dataset at a time; the server itself must serve the page, not a script finishing it in the browser
One file for the whole catalog	`data.json`, DCAT, a CSV table	taken with one command	grows with the catalog; becomes heavy for large catalogs

Method	What the consumer gets	Strength	Weak spot
Paginated export	the same DCAT, with a link to the next page	works for a catalog of any size	all pages must be traversed; if the catalog changes during the traversal, records may be lost
Protocol	OAI-PMH, CSW, OGC API Records	selection by modification date	a separate request for each protocol
Registration in an external registry	metadata goes to DataCite together with the DOI	the dataset is found by its identifier, without crawling the site	only for datasets with a DOI

Two decisions within these methods deserve separate discussion.

A profile is enabled, not received "for free". In ckanext-dcat the DCAT-AP 3 profile is enabled by default, while DCAT-US 3, HealthDCAT-AP and schema.org are separate profiles that the administrator names in the configuration. That is why a CKAN portal with the extension can serve DCAT and still fail its country's profile validator. The same applies to cloud services: an ArcGIS Hub site serves DCAT-US 1.1 and DCAT-AP 2.1.1 by default, while the DCAT-US 3.0 and DCAT-AP 3.0.0 feeds are enabled separately.

The export and the page are different representations. The same description can be requested in different file formats: Turtle, RDF/XML, JSON-LD. This is called content negotiation: the program specifies what it needs in the `Accept` header, or takes the address with the right extension. For a person it is invisible, for a harvester it matters. But the dataset page and the export file live separately. On 13 October 2026 a dataset page on data.gouv.fr contained no schema.org markup at all: the HTML had only a link to a Turtle file. The catalog serves DCAT, but a dataset search engine will not read its pages.

Another trap is a page drawn by a script. If the description and markup are loaded after the page opens, a robot that does not execute scripts sees an empty shell. The dataset page on data.europa.eu is built this way. More in the chapter on discoverability.

When it is needed and when not

Situation	What is needed	What usually covers it
A portal harvested by the national portal or data.europa.eu	DCAT in the country's profile, without registration, at a stable address	the core of DKAN, udata, Huwise, Socrata, ArcGIS Hub; for CKAN, ckanext-dcat with the right profile
High-value datasets in the EU	the DCAT-AP HVD profile on top of DCAT-AP	a profile in ckanext-dcat, GeoNetwork formatters, feed configuration in Hub
A catalog that must be found through search	schema.org markup on every dataset page and a sitemap	the core of Dataverse, InvenioRDM, GeoNetwork, NADA; for CKAN, the same ckanext-dcat
A geoportal in the national infrastructure	ISO 19115 and CSW or OGC API Records; for an open data portal, also GeoDCAT-AP	the GeoNetwork core
A research repository	DataCite and OAI-PMH; schema.org for search	the core of Dataverse, InvenioRDM, DSpace
A microdata catalog	DDI for each survey; schema.org for search	the NADA core

Situation	What is needed	What usually covers it
An indicator catalog	SDMX for international partners	the .Stat Suite core; PxWeb has no full SDMX
A catalog for machine learning	Croissant	the core of Hugging Face, Kaggle, OpenML; for CKAN, a plugin in ckanext-dcat; for Dataverse, in the core since version 6.10
A dozen datasets that nobody harvests	schema.org and one `data.json` file	a static site, JKAN
An internal showcase for your own staff	nothing	no export needed

Almost everyone should add schema.org markup: it costs nothing on a standard platform and decides whether a dataset reaches dataset search. A full DCAT-AP profile is needed only by those who are actually harvested. For a catalog of a dozen datasets it is excessive, but a file with all descriptions is useful even there: an aggregator will pick up the catalog from it when its turn comes.

In brief, this is what a justification looks like using the decision record template:

- **Context.** A regional open data portal on CKAN 2.11. The national portal harvests regions via DCAT-AP 3, and staff ask "to make us visible in Google".
- **Options.** Install ckanext-dcat with the default profile; enable the DCAT-AP 3 profile and schema.org markup in it; write a custom JSON export "inspired by DCAT".
- **Criteria.** Which profile the national portal expects, whether datasets reach search, what maintenance costs when CKAN is upgraded.
- **Decision.** ckanext-dcat with the DCAT-AP 3 profile and the markup plugin. Do not write a custom export.
- **Why.** The extension is maintained by the CKAN team and tested on CKAN 2.10, 2.11 and 2.12. A custom JSON would have to be explained to each harvester separately, and the default profile may not match what the national portal expects.
- **When to revisit.** If the country adopts a national profile on top of DCAT-AP or if the portal moves away from CKAN.

How it is done in platforms

The marks are core, extension, no, not checked, as in the summary table. Checked against documentation and live installations on 13 October 2026.

Platform	DCAT	schema.org	Domain format	What is important to know
CKAN 2.10–2.12	extension ckanext-dcat: DCAT-AP 2 and 3, DCAT-US 3, HealthDCAT-AP	extension: the structured data plugin in the same ckanext-dcat	the ckanext-dcat extension serves Croissant; CSW is the ckanext-spatial extension, only for harvested records	DCAT-AP 3 profile by default; paginated export with selection by date
DKAN	core: a `data.json` file	not checked	no	the core schema is DCAT-US 1.1; live installations on the new DKAN also serve DCAT-US 3.0

Platform	DCAT	schema.org	Domain format	What is important to know
udata	core: paginated DCAT and a file per dataset	no: the page HTML has no markup, only a link to a file	no	a nightly CSV export of the whole catalog is a separate core task
Huwise (OpenDataSoft)	core: the catalog as one file in RDF, Turtle and DCAT	core, but the administrator enables it: opendata.paris.fr has markup, data.economie.gouv.fr does not	no	some installations have a national profile: on data.bs.ch it is DCAT-AP-CH
Socrata	core: `data.json` as one file	core: JSON-LD on the dataset page	no	the markup on data.cityofnewyork.us contains the licence, publisher and list of files
ArcGIS Hub	core: DCAT-US 1.1 and DCAT-AP 2.1.1; DCAT-US 3.0 and DCAT-AP 3.0.0 enabled separately	no: the opendata.dc.gov dataset page has no markup	core: OGC API Records	the feed includes all site content, not only datasets
Piveau	core: DCAT in several formats, SPARQL	no: the dataset page on data.europa.eu is drawn by a script, the HTML has no markup	core: SPARQL	on data.europa.eu a dataset description in Turtle opens at an API address
Dataverse 6.x	not in the core; DCAT 3 is a third-party exporter for Dataverse 6	core: since version 6.10 a short Croissant markup is embedded in the page, the earlier schema.org remains in the export	core: OAI-PMH, DataCite, DDI, Dublin Core	Croissant was built into the core in version 6.10; earlier it was a separate exporter
InvenioRDM v14, Zenodo	core: a record in DCAT via the `Accept` header	core: JSON-LD on the record page	core: OAI-PMH, DataCite, Dublin Core, MARCXML	there is no single file with all descriptions, only the protocol
DSpace 7 and later	no	no	core: OAI-PMH and Dublin Core	the DSpace@MIT dataset page responds to programs with a check, so the markup there could not be verified
GeoNetwork 4.4	core: DCAT, DCAT-AP and GeoDCAT-AP via CSW and separate formatters	core: schema.org JSON-LD on the record page; Croissant fields are in the branch of the upcoming 4.4.13	core: ISO 19115 and CSW; OGC API Records is a separate service	the EEA catalog lists DCAT, DCAT-AP HVD and mobilityDCAT-AP among its CSW formats
GeoNode 5.1	no: the catalog serves an API and CSW, but not DCAT	no: the catalog page is a shell, the HTML has no markup	core: CSW and ISO 19115, pycsw inside; sitemap in the core	some older installations have a `data.json` file, but it is not a GeoNode 5 core export
NADA	no	core: JSON-LD on the study page	core: DDI Codebook for each study; no OAI-PMH	the World Bank Microdata Library markup has no list of files and no licence
PxWeb	no	no	core: table descriptions in JSON-stat; SDMX-ML 2.0 only in the old API	PxWebApi 2 has no SDMX format
JKAN	core: a `data.json` file declaring the DCAT-US 1.1 schema	no: the page template adds no schema.org markup; the page has RDFa `dcat:Dataset`	no	the template outputs a fixed list of fields; on 14 October 2026 opendata.scot served six, without the publisher
A hand-made static site	no: the author puts the file in place	no: the author writes the block in the template	no	the book provides a ready-made markup template

Three rows in the table contradict what is usually said about the platforms.

GeoNode does not serve DCAT. Its catalog is pycsw: CSW responds with ISO 19115 records, Dublin Core and several other formats, and DCAT is not among them. This is how live GeoNode installations in Nicaragua and Bolivia responded on 13 October 2026. The `data.json` file that the

registry finds at some GeoNode installations does not belong to the version 5.1 core: there is no such address in its code, and the file that opened contained its own list of fields, not a DCAT-US catalog.

udata does not mark up pages. data.gouv.fr on udata 17.12.1 serves a dataset description as a Turtle file, but the page HTML has no schema.org block. The catalog is open to a harvester, but not to Google Dataset Search.

Dataverse's markup has changed. Up to version 6.10, schema.org was embedded in the page, and it can be brought back by configuration. Since version 6.10 a short Croissant markup has taken its place: without the list of files, but Google asks for it so that a dataset appears in the "Croissant supported" filter. The full markup with the list of files is still served through the export.

What the registry shows

According to the dataportals-registry export (release 1.27.0 of 15 October 2026), at least one standard way of serving descriptions (DCAT, OAI-PMH, CSW, OGC API Records, SDMX or STAC) is recorded for 9,330 of 43,614 catalogs, or 21%. A DCAT export exists for 5,233. Of these, 3,152 serve both DCAT-US and DCAT-AP: mostly ArcGIS Hub sites, where both feeds are enabled by default.

The registry hardly records schema.org markup (there are 38 such addresses in it), so its share cannot be calculated from the registry. It can only be seen on the page itself.

The share depends on the platform more than on the owner's wishes. Among open data portals DCAT is marked for 239 of 245 on Socrata, 348 of 454 on Huwise, 103 of 161 on DKAN and 633 of 1,906 on CKAN. Among custom builds, 24 of 1,144. A third of CKAN portals install the ckanext-dcat extension, although without it the catalog is not harvested. The queries are in the appendix on the registry.

Extensions and their risks

CKAN has no description export in its core. All the work is done by ckanext-dcat. It is maintained by the CKAN team, under the AGPL licence; the latest release is 2.4.4 of 11 August 2026. The documentation declares support for CKAN 2.10 and 2.11, and automated tests also run on 2.12. Until support for 2.12 is announced, check the extension's behaviour on that version on your own test instance. The method for evaluating an extension is in the chapter on extension ecosystems.

There are several plugins inside the one extension, and each is enabled separately:

- **dcat** serves the catalog in RDF and reads other people's exports;
- **structured_data** adds schema.org markup to the dataset page. Without it there is DCAT, but a search engine cannot see the catalog;
- **croissant** serves a description for machine learning. In 2026 platforms serve Croissant 1.0, although version 1.1 has come out.

The default profile is DCAT-AP 3. Next to it are the DCAT-AP 2, DCAT-US 3, HealthDCAT-AP and schema.org profiles. A national profile is written as your own plugin on top of a ready-made one: this is how DCAT-AP_IT works. The risk here is not that the extension is abandoned, but in configuration. A portal with the extension installed "to tick a box" serves the default profile and fails the check of its national portal.

A separate extension, ckanext-spatial, provides CSW through pycsw, but only for records harvested from other geoportals, not for the whole CKAN catalog.

In Dataverse, DCAT export is also an extension, and a younger one. The DCAT 3 exporter from the GDCC community, release 0.8.7 of 31 August 2026, produces DCAT 3 and the Dutch DCAT-AP-NL 3.0 profile. It is designed for Dataverse 6, works best on 6.2 and later, and is itself marked as a prototype. It is worth installing for those who already know which profile is expected of them.

How it breaks: the file opens, but the profile is wrong

A check on 13 October 2026. Three exports were run through the European Commission's SHACL validator, DCAT-AP 3.0.1 profile, base rules level.

data.europa.eu, a dataset on air quality in Bristol. The Turtle export passed without errors. This is expected: the portal itself harvests descriptions according to DCAT-AP and serves them itself. But the dataset page in a browser without scripts is empty, and it has no schema.org markup. The catalog is exemplarily open to a harvester and closed to a search engine that reads only HTML.

data.gov.gr, a series of weather station observations. A catalog on CKAN 2.11.3 with ckanext-dcat, DCAT-AP 3 profile; the description includes both the high-value dataset flag and a link to the regulation. The validator found two errors. The file format is written as the text "CSV", while the profile expects a link to a type from a vocabulary. The file checksum has no algorithm specified. The page itself is marked up: the HTML has a schema.org block. The export and markup are done, but the profile formally fails because of two fields.

data.gouv.fr, a dataset on subsidies. The Turtle file opens, but the validator flags the format "xls" written as text. A full check adds the publisher and update frequency: they are written as links that are not declared as objects of the required type. The page HTML has no markup at all.

Four more failures are common.

- **The default profile.** The extension is enabled but serves DCAT-AP 2 where DCAT-AP 3 is expected, or DCAT-US 1.1 where 3.0 is already accepted.
- **An export without a page.** A harvester takes the catalog, a search engine does not. udata and ArcGIS Hub work this way.
- **A page without an export.** There is markup, but no file with all descriptions. A search engine will find the datasets, the national portal will not.
- **Foreign items in the export.** The ArcGIS Hub feed describes all site content: on opendata.dc.gov it has 1,871 records, of which search counts 1,476 as datasets. A harvester that trusts the feed will publish maps and applications as datasets.

Publishing without a portal

Without a portal, descriptions do not appear by themselves: the author puts them in place. Two files are enough.

Markup on the dataset page. A JSON-LD block with the schema.org vocabulary: title, description, creator, publisher, date, licence and list of files. A ready-made template is included in the book. It is inserted into the page template once and filled in from the dataset fields.

A file with all descriptions. For a small catalog this is `data.json` following the DCAT-US 1.1 schema: both data.gov harvesters and aggregators understand it. JKAN builds such a file from the pages automatically, but the template fills in a short list of fields and does not cover the schema's mandatory fields. Check the file before handing it to a harvester; see the chapter on static sites. On your own static site it is built by a script at each build, otherwise it will be out of date the day the second dataset is published.

If a dataset is archived in Zenodo, its DataCite description goes there together with the DOI, and a separate DCAT file is not needed for that dataset. How to set up the page and register in a catalog is covered in the chapter "Making a Publication Findable".

How to check a live portal

A manual check means four questions to a live catalog:

1. Open a dataset page and find an `application/ld+json` block with the Dataset type in the source code. Or paste the page address into Google's Rich Results Test.
2. Find the file with all descriptions and see which profile it declares. In DCAT-US this is the `conformsTo` field.
3. Compare the numbers. The export should contain as many datasets as the search on the website.
4. Run the file through the validator for the required profile. For DCAT-AP this is the European Commission's SHACL validator: use the base rules check for the version your harvester expects, currently `dcatap.3_0_1_base`.

Commands

Substitute the portal address and the address of one dataset page. The commands check the markup, typical export addresses and the declared profile.

```
PORTAL=https://data.gov.gr
PAGE=https://data.gov.gr/dataset/emy-station-189

# schema.org markup on the dataset page
curl -sL "$PAGE" | grep -o '"@type": *"Dataset"' | head -1

# File with all descriptions: paginated DCAT and data.json
curl -sI "$PORTAL/catalog.ttl" | head -1
curl -sI "$PORTAL/data.json" | head -1

# Which profile the file declares
curl -s "$PORTAL/data.json" | head -c 400
```

Next, addresses for specific platforms, checked on 13 October 2026:

```
# CKAN with ckanext-dcat: one dataset description in Turtle and the number of datasets in
the export
```

```
curl -sL "https://data.gov.gr/dataset/emy-station-189.ttl" | head -20
curl -s "https://data.gov.gr/catalog.ttl" | grep -E "hydra:(totalItems|lastPage)"

# udata: a Turtle file per dataset and the overall export
curl -sI "https://www.data.gouv.fr/api/1/datasets/dataset-slug/rdf.ttl" | head -1

# Huwise: the whole catalog as one file
curl -sI "https://opendata.paris.fr/api/explore/v2.1/catalog/exports/dcat" | head -1

# ArcGIS Hub: which feed is enabled and how many records it has
for V in dcat-us/1.1 dcat-us/3.0 dcat-ap/2.1.1 dcat-ap/3.0.0; do
  echo -n "$V "; curl -s "https://opendata.dc.gov/api/feed/$V.json" | jq '.dataset |
length'
done

# InvenioRDM: one record in DCAT and in schema.org
curl -s -H "Accept: application/dcat+xml" "https://data.caltech.edu/api/records/record-id" |
head
curl -s -H "Accept: application/ld+json" "https://data.caltech.edu/api/records/record-id" |
head

# GeoNetwork 4.4: which formats CSW serves
curl -s "https://sdi.eea.europa.eu/catalogue/srv/eng/csw?
service=CSW&version=2.0.2&request=GetCapabilities" \
  | grep -o 'dcat[^<]*'

# Research repository: OAI-PMH
curl -s "https://dataverse.harvard.edu/oai?verb=Identify" | grep repositoryName

# Dataverse: embedded Croissant markup and the list of export formats
curl -s "https://borealisdata.ca/api/info/exportFormats"
```

A good result: a Dataset block in the HTML, the export file opens without login, it declares the required profile, and the validator finds no base-level errors. A bad one: the address responds, but the profile is old, formats are written as text instead of vocabulary links, and the dataset page has no markup. The summary list is in the checklists.

What follows from this

- First find out who will take your descriptions and which profile they expect. An export "in DCAT in general" does not meet this requirement.
- Provide both an export and markup. The first feeds a harvester, the second a search engine, and platforms cover them separately.
- Check the export with a validator for the right profile version, not just by the server response. A file format written as text instead of a vocabulary link is a typical reason for rejection.
- On CKAN all this is done by one extension, ckanext-dcat, but the profile and markup must be enabled explicitly. Without configuration the extension serves the default profile.
- If you are writing the catalog yourself, include in the specification both an export file in the country's profile and a schema.org block in the page template. Neither will appear by itself.

Sources

- DCAT 3, W3C Recommendation

- DCAT-AP 3.0.1, SEMIC recommendation of 27 October 2025; draft 3.0.2 of 17 June 2026; the GeoDCAT-AP, StatDCAT-AP, DCAT-AP HVD profiles
- DCAT-US 3.0 and the DCAT-US 1.1 schema
- Dataset markup in Google's documentation: schema.org and DCAT, mandatory fields, the Croissant filter
- ckanext-dcat documentation: profiles, markup for Google Dataset Search, release 2.4.4 of 11 August 2026; ckanext-spatial: CSW
- Dataverse metadata export, discoverability and Croissant, built in since version 6.10; the GDCC DCAT 3 exporter, release 0.8.7 of 31 August 2026
- InvenioRDM record serializers: DCAT, DataCite, Dublin Core, schema.org
- GeoNetwork 4.4 and search engines: JSON-LD schema.org with Croissant; DCAT formats in GeoNetwork 4.4
- The pycsw catalog in GeoNode's code and pycsw output formats
- ArcGIS Hub: catalog feeds; DCAT-US 3.0 and DCAT-AP 3.0.0 feeds on opendata.dc.gov, checked 13 October 2026
- The European Commission's DCAT-AP SHACL validator: validation types for DCAT-AP 3.0.1, 3.0.0, 2.1.1 and DCAT-AP HVD
- Live checks on 13 October 2026: data.gov.gr (CKAN 2.11.3, dataset emy-station-189), data.gouv.fr (udata 17.12.1), opendata.paris.fr, data.economie.gouv.fr and data.bs.ch (Huwise), data.cityofnewyork.us (Socrata), opendata.dc.gov (ArcGIS Hub), data.europa.eu (a dataset on air quality in Bristol), the EEA catalog (GeoNetwork 4.4), the Ifremer Sextant catalog (GeoNetwork 4.4.13), GeoNode in Nicaragua and Bolivia, the World Bank Microdata Library (NADA), Borealis (Dataverse 6.8.4), CaltechDATA (InvenioRDM), opendata.scot (JKAN), the DCAT export of an IPT installation in New Zealand
- I. Begtin, Why data on Russia is so hard to find (in Russian), November 2024, and Dataset search engines as global data discovery systems (in Russian), May 2022: which portals reach dataset search and why
- Prevalence of exports: a query to the dataportals-registry export, release 1.27.0 of 15 October 2026. The queries are given in the appendix
- Metadata Standards, Discoverability, Metadata Depth

Metadata Completeness and Depth

A dataset description does two jobs: it is how the dataset is found, and it is how people decide whether they may use it. Hence the main practical conclusion of the chapter. A short mandatory part that every publisher fills in and a program checks. Subject-specific fields only where data cannot be understood without them. A data dictionary for every table. And no placeholder values: an empty field is visible at once, while a placeholder looks filled in and misleads both people and programs.

The chapter covers what standards require and what they do not, how platforms let you add your own fields and describe table columns, and how to check in half an hour how honestly descriptions are filled in on a live portal. The check at the end of the chapter was done on two national portals, the Greek and the French, and shows how a formally complete description turns out to be empty.

What it is

Three properties of a description are often confused.

Completeness: what share of the required fields is filled in. It is counted across datasets: how many datasets out of a hundred have a licence, a period, a publisher.

Depth: how detailed the schema itself is: only a title and description, or also period, territory, methodology, list of columns. Depth is set by the platform and the catalog owner; completeness is provided by publishers.

Accuracy: whether what is written in the field is true. A period of "1900–2099" for a table covering one year, a description that repeats the title, a territory of "the whole country" for data from one district. Formally the field is filled in; in substance it is not.

A platform can be detailed and still empty: twenty mandatory fields, each holding a dash. And the other way round: a short schema filled in honestly is more useful than a long questionnaire. An extreme case is the computational chemistry catalog ioChem-BD on DSpace: the title of its datasets is often just a numeric code, for example "000112758". A chemist who knows what they are looking for will open such a dataset from a link in a paper. Neither the catalog's search nor dataset search engines will find it.

Why it is needed

To be found. Search and filters work only on filled-in fields. A dataset without a period will not appear in the "2020–2024" filter, a dataset without a theme will not appear under "Transport". Dataset search engines are stricter: Google accepts dataset markup only if it has a title and a description between 50 and 5,000 characters long.

To decide whether it may be used. Without a licence the user's rights are undefined, even if the file downloads: what may be done with the data depends on the country's law and on whether the data is protected at all. A careful user will not take such a dataset. Without a publisher and contact there is no one to ask a question or report an error to.

To understand what is inside. A `sum_2` column without a data dictionary is open in form and closed in meaning. In 2021 Russia's Federal Water Resources Agency opened the "Water Data" system with dataset passports, an API and digitally signed files. The passports had no description of the field structure, and one field in the register of water users held the owner's name, address and taxpayer number together. The data is there, the documentation is not.

So that a dataset outlives its authors. In ten years nobody will remember what code `7` in the "status" column means. That is why archives and research repositories describe things in more detail than open data portals: they work for a future reader, not for today's report. Five models of working with metadata (scientific, archival, public-sector and civic, technological and commercial) are covered in the chapter on classifying catalogs. The model determines what depth of description is normal for a catalog.

So that a harvester does not have to guess. An aggregator that collects descriptions from many catalogs is forced, when a dataset description is poor, to substitute properties of the catalog itself: if the whole catalog is about healthcare, its datasets get the theme "health". This is how the Dateno search engine handles layers from ArcGIS and GeoServer servers, which have almost no description of their own. It is better than nothing, but the precision of such search is the precision of a guess.

Four levels of description

The minimum, without which a dataset cannot be used:

- a title and a description in plain language that does not repeat the title;
- who published it and how to contact them;
- a licence;
- the date the data was last updated, not the date the record was edited;
- a permanent address.

This is enough to find a dataset and understand whether it may be used.

The recommended level adds what makes a dataset easy to filter and to keep updated:

- temporal and spatial coverage;
- how often it is updated;
- a theme from a shared vocabulary;
- file format and size;
- language.

The subject level is needed where data cannot be interpreted without it:

- statistics: methodology, unit of measure, breaks in series;
- geodata: coordinate reference system, scale, feature type;
- research: authors with ORCID, funding, link to the paper;
- microdata: sample, questionnaire, access levels.

Content description is a data dictionary: what each column means, its type and unit, what an empty cell means. This is metadata no longer about the dataset but about its contents, and it is organised separately. See the section "Data dictionary" below.

How data dates differ from record dates is covered in the chapter "Anatomy of a Catalog". In short: if the platform does not distinguish them, the "updated" date tells you about an edit to the description, not about how fresh the data is.

What standards require and what they do not

A standard's mandatory fields are the minimum for exchange between programs, not the minimum for usefulness. The most surprising thing in the table below is where the licence ends up.

Standard	Mandatory	Important but not mandatory
DCAT-AP 3.0.1, the EU portal profile	for a dataset, title and description; for a distribution, access URL	a dataset has 7 recommended properties: contact, distribution, spatial coverage, keywords, publisher, temporal coverage, theme. Another 27 are optional, including issue date, modification date and update frequency. Licence is a recommended property of the distribution, not of the dataset
The DataCite schema, used to register DOIs	identifier, creator, title, publisher, publication year, resource type	description, subject, dates, geolocation are recommended; licence, language, format, size, version, funding are optional
schema.org/Dataset per Google's requirements	name, description from 50 to 5,000 characters	licence, creator, temporal coverage, spatial coverage, variables measured, identifier are recommended

Two conclusions follow.

First: build your own list of mandatory fields from the user's needs, do not copy it from a standard. Licence, update date and period are mandatory for the user, although none of the three standards calls them that.

Second: in DCAT-AP the licence lives on the file. A platform built on the profile stores it there too, and a check of "does the dataset have a licence" will give the wrong answer on it. An example is in the section on checking a live portal.

If a value is genuinely unknown, it is more honest to say so than to invent one. The DataCite schema has standard codes for this: `:unkn` for unknown, `:none` for no value, `:null` for deliberately empty. The cost of such honesty shows in an example from the DataCite documentation: a citation of a dataset about which almost nothing is known looks like `:unkn (2020). :none. :null.` But nobody will mistake it for a complete one.

Implementation options

Platforms let you configure the description schema in five ways. The way determines who can change the fields and what it costs at upgrade time.

A fixed schema. Fields are set by the platform and cannot be changed, or only by editing code. Lightweight catalogs such as JKAN and a static site work this way: the template author writes the schema, and fields are not validated. ArcGIS Online and ArcGIS Hub sites have no custom fields,

but the administrator chooses one metadata style for the organisation from ready-made standards: ISO 19139, ISO 19115-3, INSPIRE, FGDC, the North American Profile of ISO 19115 or Dublin Core+. Organisations created after July 2025 have Dublin Core+ by default, older ones ISO 19139.

A schema from a file. The administrator describes fields in a file, and the platform builds a form and validation from it.

- In CKAN this is done by the ckanext-scheming extension. The schema is written in JSON or YAML and defines fields for datasets, resources, organisations and groups: whether they are required, lists of allowed values, ready-made field presets, validation functions. Since CKAN 2.12 required fields can be skipped for a draft until it is published.
- In DKAN the dataset schema is a JSON Schema file, based on DCAT-US by default. Your own file replaces the standard one, and everything entering the catalog is validated against it. The documentation warns: change the schema carefully when the catalog already holds datasets; an incompatible edit breaks validation of old records.
- In DSpace fields come from the metadata schema registry (Dublin Core by default), and input forms are described in the `submission-forms.xml` file. A field can be shown only for a certain type of item, for example only for a dataset. After editing the forms the server is restarted.

Blocks and profiles. The platform ships ready-made field sets for sectors; you enable them and add your own.

- Dataverse has seven supported blocks: citation, geospatial, social sciences and humanities, astronomy, life sciences, journals, 3D objects. There are experimental ones too: CodeMeta for software, computational workflows, archival information, Local Contexts labels for Indigenous knowledge. The collection administrator decides which fields are mandatory in it. A custom block is described in a TSV file and loaded through the API. The documentation reminds you separately: new fields must also be added to the Solr search index schema, otherwise they cannot be searched.
- GeoNetwork 4.4 has ISO 19139, ISO 19115-3, Dublin Core and ISO 19110 in its core; the last is a feature catalogue, that is, a data dictionary for geodata. National profiles are installed as schema plugins: Swiss, Canadian HNAP, Dutch, Italian RNDT, British GEMINI, and also DCAT-AP.
- InvenioRDM stores descriptions according to the DataCite schema. Resource type, title, publication date and creators are mandatory. Ready-made additional fields cover journal, imprint, thesis, conference details, and CodeMeta for software. Custom fields are defined in the site configuration and arranged into sections of the form.

Templates in a cloud service. The administrator creates custom fields in the web interface, without a programmer.

- In Huwise (formerly OpenDataSoft) the basic "Standard" template cannot be changed or deleted. Next to it you enable compatibility templates (DCAT, INSPIRE) and your own: the administrator creates fields with a type, allowed values and rules, and enables translation per field. Separate administrative templates are visible only in the back office.

- In Socrata (Tyler Data & Insights) custom fields are grouped into fieldsets. Each field has its own requiredness (a dataset cannot be published without a required field) and visibility: to everyone or to staff only. A field can carry a validation in the SoQL query language, including one that depends on other fields, and an audit tool shows which datasets break the rules. Only fields with predefined names from the "Common Core" set go into the `data.json` file for data.gov.

A template as part of the standard. A full standard is too big for a publisher: DDI or ISO 19115 has hundreds of elements. A good solution is in the World Bank's metadata editor, used by microdata catalogs on NADA. For each data type it has two kinds of ready-made templates, both read-only: a full one with all elements of the standard, and a recommended one with only the main ones. An organisation copies a template and edits the copy: removes what it does not need, changes labels, makes fields mandatory. The standard and the export stay shared, while each publisher has its own form. The same technique works on any platform: one schema, different forms for different publishers.

Data dictionary

A data dictionary describes the columns of a table: name, readable label, meaning, type, unit of measure, allowed values, how missing values are marked. Without it a table is open in form and unintelligible in substance. In the Parnitha weather station dataset on data.gov.gr, all 57 columns in the table store are recorded as text, including temperature and pressure, and none has a description. The record is analysed in full in the chapter "Anatomy of a Catalog".

A dictionary has several machine-readable forms; choose the one accepted in your sector:

Form	Where it is used
Table Schema from the Frictionless family, usually inside a `datapackage.json` file	open data, publishing in a Git repository, automated file validation
CSV on the Web (CSVW), a 2015 W3C Recommendation	tables with linked data; Google, in beta, reads a table's dictionary in this form directly from page markup
DDI variables	microdata: each variable has the question text, response codes, frequencies
ISO 19110, feature catalogue	geodata: feature types of a layer and their attributes
Croissant fields and record sets	datasets for machine learning
SDMX structure	indicators: dimensions, code lists, attributes

A dictionary is doubly useful if the file itself is validated against it. Then "number" in the dictionary means the column really holds numbers, not the text "n/a". In France this is organised at national level: the schema.data.gouv.fr catalog had 106 shared schemas in October 2026, 69 of them in Table Schema. A dataset resource on data.gouv.fr references a schema, and the portal validates the file against it. A shared schema for one topic, for example electric vehicle charging stations, lets files from hundreds of publishers be merged into one table. But even there a link to a schema is rare: in a sample of a thousand datasets, 36 had one.

In DCAT, the link between a dataset and a standard is recorded with the `dct:conformsTo` field: it says which schema or sector standard the file conforms to. Sector data standards themselves (OCDS, GTFS, Darwin Core and others) are covered in the chapter on what is published.

When detail does harm

A mandatory schema of fifty fields slows publication down. Fields start being filled in formally: a description is copied from a neighbouring dataset, the title is repeated in the description, today's date goes into "data period". Worst of all is when the program itself does this: it substitutes a default value so that the form passes validation. What this looks like on a national portal is shown below, in the data.gov.gr check.

Hence three rules.

- A short mandatory schema with optional subject fields is better than a long questionnaire nobody reads.
- A field the publisher may not know must not have a default value. An empty field is visible in any check; a placeholder is visible in none.
- Test whether a field should be mandatory on three to five real publishers before launch. If half of them cannot fill it in, make the field optional or help fill it in, with a vocabulary, a hint, or automatic calculation from the file.

Custom fields are needed if a sector vocabulary or the law requires an attribute that the standard schema lacks: the high-value dataset category, the legal basis for publication, a funding programme code. They are not needed if you want to keep an internal number from an accounting system "just in case". The opposite extreme also occurs: carefully describing what does not need cataloguing at all, for example every service dump of an intermediate calculation.

When it is needed and when not

Situation	What to do	What usually covers it
An open data portal in the EU harvested by the national portal	DCAT-AP and the national profile; a licence on every file; themes and frequency from EU vocabularies	CKAN with scheming and ckanext-dcat, Piveau, Huwise with the DCAT template
A small municipal portal	minimum and recommended levels; no custom fields; a data dictionary for tables	the fixed schema of a cloud service, or CKAN without scheming
A research repository	DataCite plus a discipline block; authors with ORCID, link to the paper	Dataverse with sector blocks, InvenioRDM with custom fields
A geoportal	ISO 19139 or ISO 19115-3, the country profile, a feature catalogue for vector layers	GeoNetwork with a profile plugin, an ISO style in ArcGIS
A microdata catalog	DDI with variable descriptions	NADA with the metadata editor, Dataverse
An indicator catalog	SDMX structure, methodology, units	see the chapter on indicator catalogs
Publishing without a portal	a README file, `datapackage.json` with a dictionary, schema.org markup on the page	this book's templates, validation in CI

A custom field schema is worth creating when the law, the upper-level portal's profile or the sector requires it. It is not worth creating for the sake of a complete questionnaire: each field is work for the publisher on every dataset and work for the administrator at every platform upgrade.

In brief, this is what a justification looks like using the decision record template:

- **Context.** A regional portal on CKAN; the national profile requires two fields the core lacks: the high-value dataset category and the legal basis.
- **Options.** Write values into free-form extra fields (`extras`); install ckanext-scheming; move to a cloud service with templates.
- **Criteria.** Validation of values on input, export to DCAT-AP, cost of upgrades.
- **Decision.** ckanext-scheming with a YAML schema and ckanext-dcat with the country profile.
- **Why.** Free-form fields are not validated and end up in the export under arbitrary names. Moving to a cloud service for two fields costs more than one extension maintained by the CKAN community.
- **When to revisit.** If the country profile starts being shipped as a ready-made extension, or if the number of mandatory custom fields grows beyond a dozen.

How it is done in platforms

The marks are core, extension, no, not checked, as in the summary table. Checked against platform documentation on 12 October 2026.

Platform	Custom description fields	Table data dictionary	What is important to know
CKAN 2.12	extension ckanext-scheming: a schema in JSON or YAML	core: a dictionary for tables in the DataStore (label, description, column type); since 2.11 also Table Designer, a table with a schema and value validation	multilingual descriptions are the ckanext-fluent extension; validating files against Table Schema is the ckanext-validation extension, see below
DKAN	core: replacing the JSON Schema file	not checked	DCAT-US schema by default; changing it on a live catalog is risky
Huwise (OpenDataSoft)	core: custom field templates, administrative templates	core: label, description and one of 12 types for each field; the "Schema" tab shows the fill rate of each column	the basic template cannot be changed
Socrata	core: custom fieldsets with requiredness and SoQL validation	core: column name, description and type, served at `/api/views/ID.json`	only "Common Core" fields go into `data.json`
ArcGIS Online and Hub	no: one style from ready-made standards per organisation	core: field descriptions, value types and lists of allowed values on a hosted layer	the layer description and the item description in the catalog are not synchronised
Dataverse	core: sector blocks and custom blocks in TSV	core: for uploaded tables (SPSS, Stata, R, Excel, CSV), variables are described in DDI	new fields must be added to the Solr index schema
InvenioRDM	core: custom fields in configuration	no	based on the DataCite schema
DSpace	core: schema registry and `submission-forms.xml`	no	editing forms requires a restart

Platform	Custom description fields	Table data dictionary	What is important to know
GeoNetwork 4.4	core: ISO 19139, ISO 19115-3, Dublin Core; country profile as a schema plugin	core: ISO 19110 feature catalogue	the plugin must match your exact GeoNetwork version
NADA	not checked	core: DDI variables with frequencies	field templates live in the separate World Bank metadata editor
JKAN and a static site	no: fields are set by the template	no: the dictionary is put next to the data as a file	validation is written by the site author

A flexible schema almost always means maintenance: someone has to decide which field is mandatory and keep the schema from sprawling. The deeper custom fields are built into the platform, the more expensive each upgrade.

Extensions and their risks

Custom fields and data dictionaries are where the cost of extensions is most visible. The method for evaluating an extension is in the chapter on extension ecosystems; here is what turned up during the check.

- **ckanext-scheming and ckanext-fluent** are maintained by the CKAN community, with automated tests on CKAN 2.10, 2.11 and 2.12. They are a normal part of a mature CKAN portal, not something exotic. Details are in the chapter on CKAN.
- **ckanext-validation** validates uploaded tables against Table Schema using Frictionless. In the main Frictionless Data repository in October 2026, compatibility is declared only with CKAN 2.9; the compatibility table says "no" for 2.10 and 2.11, and tests run only on 2.9. The living branch is the Queensland Government fork qld-gov-au/ckanext-validation: its tests run on CKAN 2.10, 2.11 and 2.12. This is a typical fate for an extension: the original author stepped back, and one of the users maintains the working version. Before installing, check the repository and branch, not the name.
- **GeoNetwork schema plugins** are tied to the core version: a plugin has its own branch for each GeoNetwork line. One plugin can have copies in different places, and they diverge. The British GEMINI 2.3 profile is maintained by Astun Technology, and its repository has a branch for GeoNetwork 4.4. The shared metadata101 collection of plugins holds a copy that has not been updated since February 2023 and goes only up to 4.2.
- **Custom Dataverse blocks and a custom DKAN schema** are not extensions but core configuration. They too are checked at every upgrade: a Dataverse block needs a matching index schema, and an incompatible edit to the DKAN schema breaks validation of old records.

Publishing without a portal

Without a portal, completeness of description is entirely up to the author: no form will demand a licence. On the other hand, depth is not limited by anything. Three files next to the data are enough:

- `README.md`: what it is, for what period, who is responsible, how it is updated, the licence;
- `datapackage.json`: the list of files and the data dictionary of each table in Table Schema;

- schema.org/Dataset markup on the dataset page, if the dataset has its own website.

Ready-made README, datapackage.json and schema.org markup are in the book's templates. Validation can be left to a program: the command `frictionless validate datapackage.json`, run in the repository's CI on every change, checks files against the dictionary and will not let a table be published where text has appeared in a numeric column. How to set up such a repository is covered in the chapter on Git repositories.

How to check a live portal

A manual check is still the best first step:

1. Open ten random datasets from different organisations, not the most recent ones.
2. For each, note whether licence, publisher, contact, update date, period and territory are filled in.
3. Compare the description with the title: does it just repeat it?
4. See whether a tabular dataset has a data dictionary and whether the column types match the contents.
5. Check whether values look true: the period against the file's contents, the territory against the publisher.

The fifth step matters more than the others. A profile validator and the metadata quality assessment on data.europa.eu (MQA) check that a field exists, not that it is true. A placeholder passes both.

Two checks on a sample

On a large portal the check is done by a program. On 12 October 2026 a thousand datasets were taken through the API on each of two national portals: ten blocks of a hundred datasets, spread evenly across the whole catalog. The figures are a snapshot for that day.

data.gov.gr is Greece's national portal on CKAN 2.11 with a DCAT-AP-based schema, about 22,700 datasets in total. The sample included datasets from 118 organisations.

Field	Looks filled in	Filled in for real
Period	97%	938 datasets out of 1,000 have the period "1900-01-01 to 2099-12-31", the same for a ministry, the state archive and dozens of municipalities. The program put it there, not the publisher
Territory	96%	for 60% the territory name is an empty string, for another 35% it is the whole of Greece
Contact and publisher	24% and 31%	22% and 28%: in some records all contact fields are empty strings
Description	94%	31% have a description shorter than 50 characters, 22% repeat the title word for word
Licence	32% of datasets, if you look for it on the dataset	99% of datasets, if you look for it on the files, as DCAT-AP requires
Update frequency	36%	32% of the sample has the value "not planned"; only 4% have a real frequency

The placeholder period also goes into the DCAT export harvested by the European portal: the dataset description contains `dcat:startDate "1900-01-01"` and `dcat:endDate "2099-12-31"`, next to an empty English title. The harvester and the quality assessment will see the period as filled in for almost every dataset in the country.

data.gouv.fr is France's national portal on udata, about 74,400 datasets in total; 78% of the sample was harvested from other catalogs. udata has its own built-in description quality score in the core: nine equal criteria, from licence and period to file availability. The API returns it for each dataset, and the site shows it next to the record. The median for the sample is 5 out of 9.

- The licence is "not specified" for 19% of datasets. Yet the built-in score credited a licence for 98%: according to the udata source code, the criterion is met if the licence field is filled in at all, and "not specified" is also a value.
- A period is given for 23% of datasets; update frequency is "unknown" for 70%.
- Territory: the level of granularity (commune, department) is given for 63%, the territories themselves for 14%.
- A description longer than 100 characters, as the built-in score requires, is present for 73%.

Both portals are good, live and large. That is exactly why the main point is visible on them: formal completeness and real completeness differ several times over, and neither the built-in score nor the external validator notices the gap.

Commands

For a CKAN portal. The command takes 500 datasets and counts the share of short descriptions, descriptions that repeat the title, licences on the dataset and on files, and the two most common period values. The name of the period field depends on the portal's schema: on data.gov.gr it is `temporal_coverage`; on another portal look it up in the `package_show` response.

```
PORTAL=https://data.gov.gr
curl -s "$PORTAL/api/3/action/package_search?rows=500&start=5000" \
  | jq '.result.results as $r | ($r | length) as $n | {
      datasets: $n,
      short_description: ([$r[] | select((.notes // "") | length < 50)] | length / $n),
      description_is_title: ([$r[] | select((.notes // "" | gsub("^\\s+|\\s+$"; "")) ==
.title)] | length / $n),
      license_on_dataset: ([$r[] | select((.license_id // "") != "")] | length / $n),
      license_on_files: ([$r[] | select(any(.resources[]; (.license // "") != ""))] | length
/ $n),
      top_periods: ([$r[] | .temporal_coverage | tostring] | group_by(.) | map({value: .[0],
n: length}) | sort_by(-.n) | .[0:2])
    }'
```

The most useful part here is the most common values. If one value appears in most datasets from different organisations, it is a placeholder. The same technique works for territory, update frequency and contact.

For a udata portal. The command compares the licence with the built-in score and counts period and frequency on one page of results:

```
curl -s "https://www.data.gouv.fr/api/1/datasets/?page_size=100&page=200" \
```

```
  | jq '.data as $d | ($d | length) as $n | {
      datasets: $n,
      license_not_specified: ([$d[] | select(.license == "notspecified")] | length / $n),
      score_counts_license: ([$d[] | select(.quality.license)] | length / $n),
      frequency_unknown: ([$d[] | select(.frequency == "unknown")] | length / $n),
      with_period: ([$d[] | select(.temporal_coverage != null)] | length / $n)
    }'
```

Run the commands several times with different `start` or `page` values so that the sample does not come from a single organisation. For a Socrata portal, the data dictionary and dates are served at `/api/views/DATASET-ID.json`: each column has a name, type and description, and the data update date is recorded separately from the description edit date. How to turn such shares into metrics and what to do with them next is in the chapter on quality metrics.

What follows from this

- Build your own list of mandatory fields from the user's needs: licence, publisher, contact, data update date, period. Do not stop at the standard's mandatory fields.
- Remove default values from fields the publisher may not know. An empty field is better than a placeholder.
- Require a data dictionary for every table and, where possible, validate the file against it.
- Create custom fields for the law, a profile or a sector, not "just in case". Each field costs work on every dataset and at every platform upgrade.
- Check schema extensions by repository and by the branch for your core version, not by name.
- Measure real completeness: the most common values, titles repeated in descriptions, plausibility of the period. Trust built-in scores and validators only on whether a field exists.

Sources

- DCAT-AP 3.0.1: mandatory, recommended and optional properties of datasets and distributions; checked 12 October 2026
- DataCite Metadata Schema: mandatory and recommended properties, codes for unknown values
- Dataset markup in Google's documentation: mandatory properties, description length, table dictionaries in CSVW
- CKAN 2.12: ckanext-scheming, DataStore data dictionary, Table Designer, 2.11 changelog
- ckanext-validation: the Frictionless Data repository, with a compatibility table and tests only on CKAN 2.9; the qld-gov-au fork, with tests on CKAN 2.10–2.12; checked 12 October 2026
- DKAN: replacing the dataset schema
- Dataverse: supported metadata blocks, metadata customisation, tabular data ingest and variable description
- InvenioRDM: metadata reference, custom fields
- DSpace: submission forms
- GeoNetwork: core schemas in release 4.4.12 of 8 July 2026, metadata101 plugins, GEMINI 2.3 by Astun Technology

- Huwise: metadata templates, dataset schema, the "Schema" tab
- Socrata: custom metadata, fields for data.json
- ArcGIS: metadata styles, describing layer fields
- World Bank: metadata editor and templates
- Table Schema, CSV on the Web, the schema.data.gouv.fr schema catalog
- udata: how the dataset quality score is calculated
- Samples: the data.gov.gr API and the data.gouv.fr API, 1,000 datasets each, 12 October 2026; ioChem-BD, dataset 000112758
- I. Begtin, Telegram (in Russian): models of working with metadata, 10 May 2021; Rosvodresursy's "Water Data", 21 August 2021; ioChem-BD, 10 January 2025; catalog properties are passed to datasets, 15 January 2025; quality assessment starts with metadata completeness, 1 August 2025
- Anatomy of a Catalog, Metadata Export, Quality Metrics, Metadata Standards

APIs and Bulk Download

A website is enough for a person. A program needs a stable way to ask the catalog: what datasets exist, what is in the description, where the file is, what has changed since yesterday. That is the application programming interface, the API.

The main practical conclusion of the chapter. Every catalog needs two things: a way to take all descriptions at once and direct links to files, both without registration. A search API and an API to table contents are added according to the size of the catalog and the users' tasks. API limits are fair only if a full dump sits next to them: otherwise everyone who needs the whole database downloads it piece by piece, which is expensive both for them and for the server.

The chapter covers four tasks usually called by the single word "API", the options for solving them, access conditions and limits, interfaces the site owner does not know about, and how platforms do it. At the end there is a check of ten live catalogs on different platforms on 12 October 2026, with commands you can use to repeat it.

What it is

An **API** (application programming interface) is an address at which a program asks the catalog a question and gets an answer, usually in JSON or XML. What matters is not the word "API" itself but which question the interface answers. There are four such questions.

1. **Find and read a description.** Search by text and filters, the record of one dataset. Examples are CKAN's Action API and Dataverse's Search API.
2. **Take all descriptions at once.** One file or protocol that serves the whole catalog: a `data.json` file, a DCAT export, the OAI-PMH and CSW protocols. Harvesters need this: an upper-level portal, a dataset search engine, an archive.
3. **Get the data itself.** A link to a file, a query to a table, a map service, a statistical API.
4. **Find out what has changed.** An RSS or Atom feed, a "modified since date" selection, change files. So as not to download the whole catalog again every day.

Three more terms will be needed below.

Bulk download: the ability to download the whole database or the whole dataset as files, without per-record requests. The EU regulation on high-value datasets defines it as downloading the whole dataset in one or several packages.

Pagination: a response split into pages, 100 datasets or 25 records at a time. A program goes from page to page until it reaches the last one. Almost all search APIs and many DCAT exports work this way.

Change feed: a list of recently added and modified records in RSS or Atom format. Subscriber programs and harvesters read it.

Do not confuse a catalog's API with an API catalog. Here the API is the door into a catalog of datasets. An API catalog is a separate type of site where the interfaces themselves are what is described; it is covered in the chapter on other types of catalogs.

Why it is needed

Without a programmatic entrance, a catalog is visible only to those who open the site in a browser. Here is who is left out:

- **the upper-level portal and aggregators.** The national portal harvests regional ones, data.europa.eu harvests national ones, dataset search engines harvest everyone. They need all descriptions and the day's changes. How such chains work is covered in the chapter on federation and harvesting;
- **search engines.** A sitemap and markup on pages are enough for them; markup is covered in the chapter on metadata export;
- **developers and analysts.** A bus timetable app or a daily report will not download a file by hand;
- **archives.** A web archive saves pages but saves data behind an API poorly. When a catalog is shut down, what can be saved is what can be downloaded in full. More in the chapter on archiving;
- **models and AI agents.** MCP servers for data portals run on top of the same API. Without an API, an assistant is left reading the site's pages. More in the chapter on trends;
- **the owner.** Many catalog websites are built on top of their own API, and the interface for people is just one of its clients.

For some data, an API and bulk download are required by law. EU Regulation 2023/138 on high-value datasets (geospatial, earth observation and environment, meteorological, statistics, companies, mobility) requires public bodies to:

- make datasets available through an API and, where the annex to the regulation says so, also as bulk download. The annex requires both for all six categories, with specific exceptions: for example, numerical weather prediction model data only through an API;
- publish the API terms of use and quality of service indicators: performance, capacity, availability;
- provide API documentation in an open format readable by both people and programs, which in practice means an OpenAPI description;
- designate a contact point for questions about the API.

These four requirements are a good list for those not covered by the regulation, too.

Implementation options

All descriptions at once

Method	How it works	Strength	Weak spot	Where it is found
One file	the whole catalog in one file: `data.json` per DCAT-US, a DCAT file, a CSV table	taken with one command, understandable without documentation	grows with the catalog; becomes heavy at hundreds of thousands of datasets	Socrata, ArcGIS Hub, Huwise, DKAN, static sites
Paginated DCAT export	the same DCAT, split into pages with links to the next and last	does not load the server, works at any size	all pages must be traversed; if the catalog changes during the traversal, datasets may be lost or repeated	CKAN with ckanext-dcat, udata
Harvesting protocol	OAI-PMH for research repositories, CSW and OGC API Records for geoportals	a standard, with selection by modification date	slow: hundreds of thousands of records mean thousands of requests; OAI-PMH has no notion of a file	Dataverse, InvenioRDM, DSpace, GeoNetwork
Regular catalog dump as files	once a day the server itself assembles the whole catalog into files and publishes them as an ordinary dataset	the cheapest method for both server and harvester	data as at the time of the dump, not as of now	data.gouv.fr: descriptions of datasets, files, services, organisations in CSV
Database dump	a copy of the whole database: SQL, RDF, a JSON archive	everything, including relations absent from descriptions	requires skill to restore; the format depends on the platform	the French Senate, Europeana
Sitemap	a list of page addresses	all search engines understand it	addresses only, without descriptions	almost all platforms; in CKAN, as an extension

A full metadata dump is needed almost always. It is the minimum that lets others harvest the catalog. A sitemap does not replace it: it tells a search engine which pages to crawl, but tells a catalog harvester nothing about licences and files.

A search API does not replace a dump either. A search API almost always has a limit per page and on the depth of results. CKAN by default returns no more than 1,000 datasets per request; Zenodo without a key returns 25 records per page and no further than the 10,000th record. Walking the whole catalog through search is sometimes possible and sometimes not, and in any case it is fragile.

Search and the dataset record

There is no single standard for a dataset search API. Each platform has its own: CKAN's Action API, udata's API, Huwise's Explore API, Socrata's SODA and Discovery API, Dataverse's Search API, the REST APIs of InvenioRDM and DSpace. The closest thing to a common standard is OGC API Records: ArcGIS Hub sites serve it, and so does GeoNetwork with a separate service. That is why upper-level portals harvest descriptions via DCAT, not through each platform's API.

A search API is needed when there are hundreds of datasets or more, when the catalog is part of a federation, or when someone builds their own site on top of it. For twenty datasets on a static site, one description file and a sitemap are enough.

The data itself

- **A direct link to a file.** The simplest and most durable method. A person, a harvester and an archive all understand it.
- **An API to a table.** Rows with filtering, sorting and pagination. In CKAN this is the DataStore: by default up to 32,000 rows per request, with SQL queries disabled until the administrator enables them. In DKAN, Huwise and Socrata, a table API is in the core; Huwise also serves a dataset as a whole file, including in Parquet.
- **An API to every file without a loader.** The catalog itself parses a published tabular file and serves an API to it within a few minutes. This is how data.gouv.fr's tabular API (still in beta) works. The limits are 100 MB for CSV, 50 MB for XLS and Parquet, 12.5 MB for XLSX. Each file has a column profile, pagination with filters and a Swagger description of queries. The service's code is open under the MIT licence. Hugging Face similarly converts datasets to Parquet and offers an SQL console for them right in the browser. More in the chapter on leading catalogs.
- **Map services.** WMS serves an image of a layer, WFS and OGC API Features serve the features themselves, ArcGIS REST serves both. More in the chapter on geoportals.
- **A statistical API.** SDMX, PxWebApi, JSON-stat: a query for a series or cube by dimension codes. More in the chapter on indicator catalogs.
- **Data in the cloud.** An S3 bucket address, a STAC catalog, a Parquet file read in parts directly over the network.
- **Server-side subsetting and processing.** The catalog serves not a file but the needed slice or result. ERDDAP accepts a subset by place and time and a response format in the URL. An openEO server runs a chain of operations on imagery and returns the result. The Climate Data Store shows ready-made request code for a subset assembled in a form.
- **A stream.** Real-time data: vehicle positions in GTFS-RT, sensor readings via the SensorThings API.

An API to table contents is needed when data is large or updated often, and downloading the whole file for one row is inconvenient. For archiving an annual report a file is enough. An automatic API inherits the source file's errors: encoding, extra rows above the header, text in a numeric column. Without a column dictionary it serves them faster, but not more accurately.

What has changed

- **An RSS or Atom feed.** Present in the core of CKAN, udata, Huwise, ArcGIS Hub and DSpace. Its length varies: CKAN by default shows the last 20 records, data.gouv.fr 50, while in Huwise and ArcGIS Hub the whole catalog is in the feed. A short feed is fine for subscribing but not for full harvesting: if a harvester misses a day, some changes fall off its end.
- **A "modified since date" selection.** The `from` parameter in OAI-PMH, `modified_since` in the ckanext-dcat DCAT export, a filter on the `metadata_modified` field in CKAN search, `pastDays` in PxWebApi 2. This is the harvester's main tool: a full dump once a month and a selection of changes every day.

- **An activity log.** In CKAN, the activity plugin shipped with the core: who changed what and when.
- **Change files.** For large databases, the difference from the previous version is published next to the full dump. Russia's FIAS address register and OpenSanctions do this.

Change notifications are needed by harvesters of large catalogs. A small portal only needs an honest modification date in its export. Honest means the date the data changed, not the date the record was edited: the difference is covered in the chapter "Anatomy of a Catalog".

Bulk download is not a luxury

If data can be obtained only through API requests, everyone who needs the whole database downloads it piece by piece. This is slow for the user and expensive for the server. The numbers show how unnecessary it is. The European Central Bank serves all its series as bulk download: in 2024 this was 108 CSV files totalling 93 GB, which took 664 MB after conversion to Parquet. The World Bank publishes the core World Development Indicators database as an archive, but in January 2025 there was no bulk download for the other indicators on data.worldbank.org. 12% of the indicators, downloaded through the API, came to 3,382 JSON files totalling 22 GB, and in Parquet the same data took 44 MB. The argument "the data is too big, so API only" usually means nobody has tried compressing it.

API restrictions are reasonable only if a full dump sits next to them. In 2024 Zenodo limited the results of a single search query to ten thousand records, while on 12 October 2026 it held 726,858 datasets. Zenodo has no file with all descriptions. The only way to take everything is OAI-PMH, request by request, fifty or so records at a time. It works, but for a catalog of this size it is slow, and most users do not know about this route.

Good examples of the opposite:

- **data.gouv.fr** publishes the whole catalog every night as CSV files, as an ordinary dataset. On 12 October 2026 dataset descriptions took 276 MB and file descriptions 1.1 GB; data services, organisations, reuses and harvest sources are published separately;
- **the French Senate** publishes its databases as PostgreSQL dumps in ZIP archives: for example, all amendments to bills since 2001;
- **Russia's FIAS address register** in 2023 published full dumps of up to 36 GB compressed, plus separate change files;
- **US agricultural statistics (NASS)** serves the whole Quick Stats database as several compressed files and offers an API for targeted queries alongside. On 12 October 2026 the latest files were from 10 October;
- **OpenSanctions** publishes pre-built dumps and change files in several formats. Technically, it is a model. Legally, it is not: the data may be used free of charge only for non-commercial purposes.

What makes a good dump:

- **a permanent address for the latest version** and an archive of past ones by date;
- **size and date next to the link**, so the user knows what they are downloading;

- **a compressed format**: gzipped CSV or Parquet, not uncompressed XML. How to choose a format and compression is covered in the chapter on formats;
- **descriptions together with the data.** In 2024 the ECB served series values in bulk, but indicator descriptions had to be extracted from the files themselves. Usually it is the other way round, and that is just as bad;
- **change files** for databases that weigh gigabytes in total;
- **a licence on the dump itself**, not only on the dataset page.

Access conditions and limits

Metadata must be readable without registration, without a key and without a CAPTCHA. Otherwise automated harvesting is either impossible or breaks every time the protection is updated. For data, a key is acceptable if it is issued immediately, free of charge and without manual approval: this way the owner sees who creates load and can contact them.

Limits are fine if they are published and predictable. A good example is Statistics Sweden's PxWebApi 2: the `/api/v2/config` address itself tells a program that one request returns no more than 150,000 cells, and no more than 30 requests are allowed per 10 seconds. A bad example is a limit you find out about from an error on page 401 of the results.

What closes a catalog to programs even when an API formally exists:

- **a bot check** at the entrance: Cloudflare, AWS WAF. A "Just a moment…" page instead of a response breaks any harvester. Examples are in the chapters on research repositories and on microdata catalogs;
- **geoblocking**: the site does not respond to requests from other countries;
- **mandatory registration** to protect against AI bots;
- **temporary links** to files that stop working after half an hour.

These cases are covered in more detail in the anti-pattern "The door is closed to programs".

Another requirement people forget is stability. An interface others rely on must not be changed silently. The old version is kept alongside the new one, closure is announced in advance, addresses are not broken. When data.gov moved from CKAN to its own software in April 2026, dataset page addresses were preserved, but on 12 October 2026 the CKAN Action API returned a 404 error. Everything built on the old API stopped working in one day.

Implicit interfaces

Many sites serve data to programs even if the owner has not said so. There are three reasons. The API was planned but not documented. The site itself is built as a page that fetches data from an API with a script, and that API can be called directly. Or the platform has an API by default and the owner does not know about it: in WordPress this is `/wp-json/`, in SharePoint `/_api/`, in DSpace research repositories the REST API and OAI-PMH, on ArcGIS servers a JSON response to any address with the `?f=json` parameter, in Blacklight catalogs the same address with an `Accept: application/json` header.

Three techniques help find such an interface: check the default addresses for the site's platform; open the browser's developer tools and look at the "Network" tab to see what requests the page makes by itself; and search for documentation with the phrase "REST API" on the site's domain.

For a user such an interface is a find: sometimes it is the only way to get data from a geoportal or a register. For the owner it is a reason to document what already exists. Relying on an undocumented interface for ongoing work is risky: it changes without warning when the site is updated. If the data is needed for the long term, keep your own copy of what you download from such an API; there are ready-made programs for this, such as apibackuper.

Data without a direct link

A catalog used to the model "a dataset is files behind links" also encounters other access methods.

- **Cloud catalogs.** The Microsoft Planetary Computer catalog serves some data as `abfs://…` links into internal Azure storage, the Amazon open data registry as S3 bucket addresses, the Google Earth Engine catalog as a code snippet for its own platform. The data is open, but it is most convenient to use inside the owner's cloud.
- **"Code first" in machine learning.** Kaggle, Hugging Face, the TensorFlow and PyTorch catalogs show a line of code that loads the dataset with a library instead of a link to a file. This is convenient for a specialist; for a general dataset search engine it is a dataset without files.
- **On request only.** Some statistical portals serve a table only in response to a POST request with chosen parameters, without a permanent address.

If you publish this way, provide at least one ordinary link to a file or dump alongside: a person, a harvester and an archive will all understand it.

When it is needed and when not

Catalog	Mandatory	Worth adding	Can be skipped
Twenty datasets on a static site or in a Git repository	a file with all descriptions (`data.json` or DCAT), a sitemap, direct links to files	schema.org markup on pages	search API, table APIs
A municipal portal harvested by the national one	a DCAT export in the country profile without registration, direct links to files	a change feed or selection by date	a custom table API, if files are small
A national portal, including one with EU high-value datasets	a search API, a full metadata dump, a selection of changes; for high-value datasets, an API and bulk download, terms of use, documentation, a contact	a daily catalog dump as files	—
A research repository	OAI-PMH, a REST API, direct links to files	a full metadata dump as a file, if there are hundreds of thousands of records	an API to table contents
A geoportal	CSW or OGC API Records, WMS and WFS or OGC API Features services, downloading a whole layer	DCAT via CSW for an open data portal	—
An indicator catalog	an API to series (SDMX, PxWebApi, JSON-stat), a list of all tables with update dates	bulk download of the whole database	—

Catalog	Mandatory	Worth adding	Can be skipped
A large register database: addresses, companies, sanctions	a full dump and change files	an API for single-record queries	—
A dataset for machine learning	a direct link to files, preferably in Parquet, and a Croissant description	a line of library code as a supplement, not a replacement	—

In brief, this is what a justification looks like using the decision record template:

- **Context.** A statistical office on PxWeb, about 3,000 tables. Analysts and a dataset search engine ask to "give us everything at once", and the API server is overloaded on release days.
- **Options.** Keep only PxWebApi; stand up a separate SDMX service; add a nightly dump of all tables as files.
- **Criteria.** How many requests it takes to download the whole database; server load; international reporting requirements; maintenance cost.
- **Decision.** PxWebApi 2 for targeted queries, and a nightly dump of all tables in CSV and Parquet with an inventory: table code, title, update date, size.
- **Why.** Through the API, the whole database takes at least one request per table, and large tables hit the cell limit and need several requests. One set of files per day is cheaper for the server than thousands of requests from each user. The dump is written as a script on top of the same API: the PxWeb core does not have one.
- **When to revisit.** If international partners require SDMX, or if the dump is hardly downloaded.

How it is done in platforms

The marks are core, extension, no, not checked, as in the summary table. Checked against documentation and live installations on 12 October 2026.

Platform	Searching descriptions	All descriptions at once	Data	What has changed
CKAN 2.12	core: Action API, up to 1,000 datasets per request	extension ckanext-dcat: paginated DCAT, 100 datasets per page by default; a sitemap is also an extension	core: files and DataStore; SQL queries disabled by default; CSV loading is the XLoader extension	core: Atom feed, a modification date filter in search, an activity log (the activity plugin shipped with the core); in ckanext-dcat, the `modified_since` parameter
udata 17	core	core: paginated DCAT; a nightly CSV dump of the catalog is a core task enabled in configuration	core: files	core: Atom feed, sorting by update date
Huwise (OpenDataSoft)	core: Explore API	core: the catalog as one file in DCAT, `data.json` and CSV	core: a records API, dataset export in CSV, JSON, Parquet and other formats	core: RSS feed, a modification date filter
Socrata	core: Discovery API	core: `data.json` as one file	core: SODA	not checked

Platform	Searching descriptions	All descriptions at once	Data	What has changed
ArcGIS Hub	core: search via OGC API Records	core: DCAT-US 1.1 and DCAT-AP 2.1.1 as one file; DCAT-US 3.0 and DCAT-AP 3.0.0 enabled separately	core: ArcGIS layers and files	core: RSS 2.0 feed
DKAN	core: metadata API	core: `data.json` per DCAT-US	core: a table store with SQL queries	not checked
Dataverse	core: Search API	core: OAI-PMH; a dataset description in DDI, DataCite, schema.org and other formats	core: Data Access API	core: OAI-PMH with selection by date
InvenioRDM	core: REST API	core: OAI-PMH, including in DataCite and DCAT	core: files	core: OAI-PMH with selection by date
DSpace 7 and later	core: REST API	core: OAI-PMH, sitemap	core: files	core: Atom and RSS feeds for any search query via OpenSearch, OAI-PMH with selection by date
GeoNetwork 4.4	core: its own search API	core: CSW (including in DCAT-AP), OAI-PMH, sitemap; OGC API Records is a separate project microservice	core: links to services and files	core: OAI-PMH with selection by date
PxWeb (PxWebApi 2)	core	core: a list of all tables with update dates	core: JSON-stat 2, CSV, PX, XLSX; no SDMX	core: tables changed in the last N days
NADA	core: its own API	core: its own API and a DDI description for each study; no OAI-PMH or DCAT	core: files by access level	not checked
JKAN, a static site	no	a `data.json` file, if it was built	links to files	the repository's change feed, if the site is built from Git: on GitHub it is `commits/branch.atom`

Three things in the table are not obvious.

In CKAN, the full metadata dump is not core but the ckanext-dcat extension. A mature portal almost always has it, but without it a harvester has only the Action API. The CKAN core has no sitemap either: data.gov.gr returns a 404 error for it.

GeoNetwork 4 removed some earlier interfaces: OpenSearch has been unavailable since version 4.0, and the separate `rdf.search` DCAT service was replaced by DCAT output through CSW. OGC API Records is installed alongside as a separate service. In the European Environment Agency catalog on GeoNetwork 4.4 that address did not respond on 12 October 2026, while CSW and OAI-PMH worked.

Second-generation PxWeb has what the first lacked: a list of all tables with update dates and a selection of those changed in N days. There is still no bulk download of the data itself in the core.

What the registry shows

According to the dataportals-registry export (release 1.27.0 of 15 October 2026), at least one programmatic entrance is recorded for 28,471 of 43,614 catalogs, or 65%. A standard way to take all descriptions (DCAT, OAI-PMH, CSW or OGC API Records) is recorded for 8,894, or 20%. The rest serve what their platform can: an ArcGIS interface, the CKAN API without the extension, a cloud service's own API.

The main difference is not between catalog types but between platforms. Catalogs on ready-made platforms have a programmatic entrance recorded in 76% of cases; custom builds in 10% (697 of 7,034), and a standard metadata dump in 58 of 7,034, under 1%. Partly this is a property of the registry itself: it finds interfaces at known platform addresses, and for a custom build the addresses are not known in advance. But a harvester faces the same thing. If you write a catalog yourself, you will have to build a standard dump yourself, otherwise nobody will harvest the catalog.

The spread by type is also large. A programmatic entrance exists for 70% of geoportals, 69% of open data portals and 63% of research repositories, but only for 37% of indicator catalogs and 23% of ML catalogs. The registry hardly records RSS and Atom feeds for CKAN portals, although they are in the core, so we do not calculate the share of feeds from the registry. The queries are in the appendix on the registry.

How to check a live portal

Developer documentation on the site itself is the first sign. Then five steps:

1. **Identify the platform.** It determines which addresses to look at: CKAN responds at `/api/3/action/status_show`, Dataverse at `/api/info/version`, DSpace 7 and later at `/server/api`.
2. **Find all descriptions at once** and get to the last page of the dump.
3. **Compare the numbers.** The dump should contain as many datasets as the site's search. If search says there are 10,000 datasets and the dump contains 200, a harvester will see a different catalog from the one a person sees.
4. **Check the selection of changes**: something should turn up for the last few days if the portal is alive.
5. **Repeat without a browser.** A request from the command line, without logging in, preferably from another network. If the response is a bot check or a 403 error, programs will not see this catalog.

Ten catalogs on 12 October 2026

Catalog	Platform	All descriptions at once	What has changed	What stands out
data.gov.gr	CKAN 2.11.3 with ckanext-dcat	paginated DCAT: 455 pages of 50 datasets, 22,736 in total, the same as in search	Atom feed; 1,328 datasets changed since 1 October, both by search and by the DCAT export	no sitemap

Catalog	Platform	All descriptions at once	What has changed	What stands out
data.gouv.fr	udata 17.12.1	paginated DCAT: 744 pages of 100, 74,375 in total; a nightly CSV dump of the catalog	Atom feed, sorting by update	the nightly dump replaces 744 DCAT requests with one download; file descriptions are a separate 1.1 GB CSV
opendata.paris.fr	Huwise	one 14 MB DCAT file, `data.json` and CSV for 490 datasets	RSS feed; 96 datasets changed since 1 October	every dataset can also be exported to Parquet
data.cityofnewyork.us	Socrata	`data.json` as one file, 4.8 MB, 2,711 records	not checked	the shared search across all Socrata portals works without a key
opendata.dc.gov	ArcGIS Hub	DCAT-US 1.1 and DCAT-AP 2.1.1 as one file, 1,871 records; DCAT-US 3.0 and DCAT-AP 3.0.0 feeds on the same site also respond	RSS 2.0 feed	the dump also includes documents, maps, applications: search shows 1,476 datasets out of 1,864 items
Harvard Dataverse	Dataverse 6.10.1	OAI-PMH; 305,251 datasets in search	OAI-PMH with selection by date	a dataset page answers a robot with an AWS WAF check, while the API and OAI-PMH work
Zenodo	InvenioRDM	OAI-PMH only, including in DCAT and DataCite; 726,858 datasets	OAI-PMH with selection by date	search: without a key, 25 records at a time, no further than the 10,000th
DSpace@MIT	DSpace 8.2	OAI-PMH, sitemap	Atom feed via OpenSearch	REST API at `/server/api`
EEA catalog	GeoNetwork 4.4	CSW: 3,429 records; OAI-PMH; sitemap	OAI-PMH with selection by date	OGC API Records not deployed
Statistics Sweden	PxWebApi 2.3.2	a list of all 2,814 tables	53 tables changed in 7 days	limits published at `/api/v2/config`

The overall conclusion of the check: all ten catalogs have a way to take all descriptions, but each does it differently, and there is no universal harvester. None of the portals checked responds at `/.well-known/api-catalog`, the address that RFC 9727 proposes for describing all of a site's APIs.

Commands

For a CKAN portal. The commands check the version, the full metadata dump, whether the numbers match, and the selection of changes. Substitute the portal address and the date.

```
PORTAL=https://data.gov.gr

# CKAN version
curl -s "$PORTAL/api/3/action/status_show" | jq '.result.ckan_version'

# All descriptions: how many datasets and where the last page of the DCAT export is
curl -s "$PORTAL/catalog.ttl" | grep -E "hydra:(totalItems|lastPage)"

# Search should show the same number of datasets
curl -s "$PORTAL/api/3/action/package_search?rows=0" | jq '.result.count'

# What has changed since 1 October: in search and in the DCAT export
curl -s -G "$PORTAL/api/3/action/package_search" --data-urlencode "rows=0" \
```

```
  --data-urlencode "fq=metadata_modified:[2026-10-01T00:00:00Z TO *]" | jq '.result.count'
curl -s "$PORTAL/catalog.ttl?modified_since=2026-10-01" | grep "hydra:totalItems"

# Change feed
curl -sI "$PORTAL/feeds/dataset.atom" | head -1
```

For other platforms:

```
# udata: the DCAT export and the nightly files with the whole catalog
curl -s "https://www.data.gouv.fr/api/1/site/catalog.rdf" | grep -o "hydra:totalItems[^<]*"
curl -s "https://www.data.gouv.fr/api/1/datasets/catalogue-des-donnees-de-data-gouv-fr/" \
  | jq -r '.resources[] | "\(.title)\t\(.filesize)"'

# Huwise: the whole catalog as one file and datasets modified after a date
curl -sI "https://opendata.paris.fr/api/explore/v2.1/catalog/exports/dcat" | head -1
curl -s -G "https://opendata.paris.fr/api/explore/v2.1/catalog/datasets" \
  --data-urlencode "limit=0" --data-urlencode 'where=modified >= "2026-10-01"' | jq
'.total_count'

# Socrata and ArcGIS Hub: the file with all descriptions
curl -s "https://data.cityofnewyork.us/data.json" | jq '.dataset | length'
curl -s "https://opendata.dc.gov/api/feed/dcat-us/1.1.json" | jq '.dataset | length'

# Research repository: OAI-PMH and records modified after a date
curl -s "https://dataverse.harvard.edu/oai?verb=Identify" | grep -o "<repositoryName>[^<]*"
curl -s "https://zenodo.org/oai2d?verb=ListIdentifiers&metadataPrefix=oai_dc&from=2026-10-
11" \
  | grep -o "<header" | wc -l

# Geoportal: how many records CSW serves
curl -s "https://sdi.eea.europa.eu/catalogue/srv/eng/csw?
service=CSW&version=2.0.2&request=GetRecords&typeNames=csw:Record&resultType=hits" \
  | grep -o 'numberOfRecordsMatched="[0-9]*"'

# PxWebApi 2: all tables changed in a week, and the limits
SCB=https://statistikdatabasen.scb.se/api/v2
curl -s "$SCB/tables?lang=en&pageSize=1" | jq '.page.totalElements'
curl -s "$SCB/tables?lang=en&pageSize=1&pastDays=7" | jq '.page.totalElements'
curl -s "$SCB/config" | jq -c '{maxDataCells, maxCallsPerTimeWindow, timeWindow}'

# Search depth limit: page 401 of 25 records on Zenodo returns error 400
curl -s -o /dev/null -w "%{http_code}\n" "https://zenodo.org/api/records?
type=dataset&size=25&page=401"
```

Implicit platform interfaces. Substitute the site's domain:

```
curl -s "https://portal.example/wp-json/" | head          # WordPress
curl -s "https://portal.example/_api/web" | head          # SharePoint
curl -s "https://portal.example/server/api" | head        # DSpace 7 and later
curl -s "https://gis.example/arcgis/rest/services?f=json" # ArcGIS server
curl -s -H "Accept: application/json" "https://portal.example/catalog?q=" | head  #
Blacklight and others
```

A good result: a response without logging in, containing dataset descriptions and not just the home page, and a dump that reaches its last page. A bad one: a CAPTCHA, mandatory registration, a 403 error for programs, or a search that cannot serve the next page. The summary list is in the checklists.

What follows from this

- Start with a full metadata dump and direct links to files. This is the minimum for any catalog, from twenty datasets on a static site to a national portal.
- Serve metadata without registration, a key or a CAPTCHA. Check this from the command line and from another network, not from your own browser.
- Give harvesters a "modified since date" selection. It is cheaper for everyone than a daily full dump.
- For large databases, publish a full dump as files plus change files. API limits without a dump close the data; they do not protect the server.
- Publish limits, terms of use and API documentation, and designate a contact. For high-value datasets in the EU this is an obligation; for everything else, good practice.
- Do not change an interface silently. When moving platforms, keep the old addresses or announce their closure in advance.
- When choosing a platform, check that the metadata dump and the change selection are in the core, not in an extension. If you are writing the catalog yourself, put a standard dump in the specification: without it nobody will harvest the catalog.

Sources

- EU Regulation 2023/138 on high-value datasets, text on EUR-Lex: Article 2 (definition of bulk download), Article 3 (API, bulk download, terms of use, quality of service, documentation, contact point)
- CKAN 2.12: API documentation, configuration: `ckan.search.rows_max`, `ckan.datastore.search.rows_max`, `ckan.datastore.sqlsearch.enabled`, `ckan.feeds.limit`; ckanext-dcat: RDF endpoints, pagination and `modified_since`
- udata: source code, the catalog CSV export task; the data.gouv.fr catalog export
- Dataverse API Guide, WordPress REST API, DSpace 7+ REST API, ArcGIS REST API
- GeoNetwork 4.4: interfaces, OAI-PMH, OpenSearch removed in 4.0, DCAT via CSW and formatters, microservices with OGC API Records
- Statistics Sweden's PxWebApi 2: list of tables and configuration with limits, version 2.3.2
- Bulk downloads: ECB, World Bank WDI, NASS Quick Stats, OpenSanctions and its licensing terms, the French Senate, Europeana
- apibackuper: saving data from a paginated API
- Checks on 12 October 2026: dumps, feeds and date filters on data.gov.gr, data.gouv.fr, opendata.paris.fr, data.cityofnewyork.us, opendata.dc.gov; OAI-PMH and search on Harvard Dataverse, Zenodo, DSpace@MIT; CSW and OAI-PMH of the EEA catalog; Statistics Sweden's PxWebApi; a 404 response from the data.gov Action API; the `/.well-known/api-catalog` address on data.gouv.fr and opendata.dc.gov
- I. Begtin, Telegram (in Russian): ECB bulk download, 20 August 2024; World Bank indicators, 24 January 2025; Zenodo API restrictions, 6 September 2024; French Senate SQL dumps, 15 November 2022; FIAS, 29 May 2023; NASS, 1 August 2024; OpenSanctions, 23 October 2024;

hidden platform interfaces, 28 November 2021; three reasons for undocumented APIs, 30 August 2022; how to find them, 26 August 2024; cloud catalogs without direct links, 27 March 2024; "code first" in ML catalogs, 21 March 2024

- Prevalence of interfaces: the dataportals-registry export, release 1.27.0 of 15 October 2026. The queries are given in the appendix.
- data.gouv.fr, API tabulaire (beta): limits of 100 MB for CSV, 50 MB for XLS and Parquet, 12.5 MB for XLSX, a profile and Swagger for each file; code at datagouv/api-tabular, MIT licence. Hugging Face, Dataset viewer and SQL Console. ERDDAP, openEO in the Copernicus Data Space Ecosystem, Climate Data Store: API. Checked 15 October 2026

Built-in Storage

A catalog can store the files themselves, or only describe them and link to wherever the file lives. This choice determines who is responsible for the file still opening in five years, and whether you can be sure it is the same file.

The practical conclusion of the chapter is this. A research repository, and any catalog whose files are cited for years, needs its own storage, and a file in it is not changed after publication: a correction is a new version. A portal that harvests other people's descriptions, and a geoportal that shows layers from map servers, can manage with links, but then the links must be checked regularly, and important datasets need a copy. A database table with an API is set up only for tables that people filter and query, not for every file.

At the end of the chapter there is a measurement on four large portals on 12 October 2026: what share of their files is uploaded, what share points elsewhere, and what happens to the links.

What it is

The word "storage" hides three different things.

File storage holds a whole file: a CSV, an archive, an image, a PDF with the methodology. The user downloads it from the catalog's address.

Table storage is a separate database into which the rows of a table are loaded. From it the portal shows the table in the browser, filters it and serves some of its rows through an API, without making the user download the whole file. In CKAN this is the DataStore; cloud services have their own record database.

A link: there is no storage at all, and the catalog keeps only the description and the address of a file or service on someone else's site.

In a DCAT description the difference shows in the distribution fields. A direct link to a file is recorded in `dcat:downloadURL`; a page or service through which the data can be obtained goes in `dcat:accessURL`. A link to an agency page is no longer a file but a promise that the file can be found somewhere there. More on the parts of a description is in the chapter "Anatomy of a Catalog".

"Built-in" does not mean "on your own server". Almost all large portals keep files in cloud object storage: Greece's data.gov.gr in Microsoft Azure, the Humanitarian Data Exchange (HDX) and Switzerland's opendata.swiss in Amazon S3. Object storage is a cloud warehouse for files, where each file sits under its own key and is served over HTTP. The user does not see this: they click "Download" on the portal page. On HDX the portal address redirects to a temporary signed S3 link, which on 12 October 2026 lived for three minutes. Link to the portal address, not to whatever ended up in the browser's address bar after the download.

Why it is needed

So that a file does not disappear along with someone else's site. If a file is in the catalog, it is available as long as the catalog lives. If the catalog only links to an agency's site, the link breaks every time that site is rebuilt. In the full data.gov.uk dump of 2023, about 10% of links to files did not open, and by 2026, by the service's own estimate, more than a quarter of links were broken. The portal collected links to the sites of local councils, and they kept changing platform or domain. The same question was examined in 2026 by the US Congressional Research Service: data.gov is by law a single entry point leading to data on agency sites, and the report directly asks whether it should become a repository so that access to data does not disappear along with an agency's site.

So that the file is the same file. A paper, a report or a machine learning model refers to specific data. If a year later a different file sits behind the same link, the conclusion can no longer be checked. Your own storage lets you calculate a checksum (a short "fingerprint" of the contents that changes if even one byte changes) and forbid swapping the file after publication. With someone else's site, neither can be guaranteed.

To work with the contents. A table in a database can be shown in the browser, filtered, served in parts through an API, converted to another format. This is also possible over an external link, but only if the catalog downloads the file itself, which means setting up storage anyway.

To check a file on upload. Encoding, delimiter, headers and conformity with the data dictionary are checked at the moment the file enters the catalog. If the file lives on someone else's site, it has to be checked again every time, and the catalog cannot fix it.

To make preservation easier. An archive is taken from one site, not from a hundred. How web archives lose data that sits behind links on other servers is covered in the chapter "Archiving and Rescuing Catalogs".

The flip side is obligations. Your own storage means disk space, backups, payment for outgoing traffic, deleting a file on demand (personal data, an error, copyright), and an answer to the question of who looks after all this in five years. The DSpace documentation, for example, warns separately that when storing in S3 you may pay for traffic between storage and server if the server is outside the Amazon cloud.

Implementation options

Links only

The catalog is a pointer. Many geoportals work this way: the layer lives on a map server, and the catalog keeps the service's address. Harvesters work the same way: data.europa.eu, dataset search engines, national portals that harvest agency descriptions. The Czech data.gov.cz does not store a single file: on 12 October 2026 all download addresses for about 31 thousand datasets led to publishers' sites. In the measurement at the end of the chapter, the Irish and Swiss national portals have not a single uploaded file in the sample.

This is a normal model if the sources are stable and maintain their own data. There is one weak spot: the catalog cannot keep a copy if the source deletes the file, and it is the last to find out.

A link with checking

The catalog keeps a link but regularly opens it and records the result. data.gouv.fr works this way: a separate crawler, hydra, checks every resource, and the resource description in the API shows the date of the last check, the server response code, availability and a checksum. In CKAN the ckanext-archiver extension does the same: it downloads files from links, shows on the resource page that a link is broken, and compiles a report on broken links for publishers.

Checking does not save the file, but it turns "links break somehow" into a list you can work with: which sources stopped responding and since what day.

A link with a copy

The next step is to keep a copy of what you link to. ckanext-archiver offers the user the saved copy if the original link breaks. The data.gouv.fr crawler goes further: it loads CSV and Excel tables, including external ones, into PostgreSQL and converts them to Parquet, and a separate tabular API works for these tables. In the measurement, the external links of Ireland's data.gov.ie also have copies of tables in the DataStore: the portal downloads the Central Statistics Office's files and loads their rows into its own database.

A copy solves the access problem but raises the question of which version is authoritative. The user should see when the copy was taken and whether it still matches the source.

Files held by the catalog

The publisher uploads a file and the catalog stores it. Technically there are three options, and the choice is made before launch: moving already uploaded files is more painful than choosing on day one.

- **The server's disk.** The simplest and cheapest for a small catalog. Backups, space and migration are the administrator's job.
- **Object storage**: S3, Azure Blob, Google Cloud Storage or S3-compatible systems on your own servers such as MinIO and Ceph. Space is almost unlimited and the provider makes copies, but you pay for storage and downloads. The user gets the file either through the catalog server or via a redirect to a temporary storage link, as on HDX.
- **Several stores at once.** Dataverse lets you choose a store for each collection or dataset; DSpace lets you set up several numbered stores and move files between them; CKAN since version 2.12 lets you keep resource files, logos and extension attachments in different places. This is needed when different data has different requirements: large imagery in cheap cloud storage, sensitive files on your own server.

For large files it matters that they do not pass through the catalog server. Dataverse can upload directly to S3, bypassing the server; InvenioRDM supports multipart upload that can be resumed after a dropped connection. InvenioRDM can also fetch a file by link: the server downloads it itself and then stores it as its own.

Trusted external storage

An intermediate option for very large and sensitive data: the file stays with the owner, but the catalog knows its exact address and access rules. In Dataverse this is a "trusted remote store": all files must sit at addresses starting with a given base address, while the dataset's auxiliary files (descriptions, thumbnails) are stored by Dataverse itself. In InvenioRDM such a file is marked as remote, and the documentation honestly warns that the platform does not guarantee the availability and integrity of such a file.

A table in a database

A CSV file is not only stored but also loaded into a database. Then you can view the first rows in the browser, request the ones you need through an API and not download everything. In CKAN this is the DataStore, a separate PostgreSQL database where each table is tied to a resource. Search returns 100 rows by default and no more than 32 thousand per request; SQL queries are disabled until the administrator enables them. The whole table is downloaded at `/datastore/dump/RESOURCE-ID` in CSV, TSV, JSON or XML.

In the cloud services Huwise (formerly OpenDataSoft) and Socrata, a database table is the main mode: a dataset is a table of records, and a file is an export from it. Dataverse parses uploaded SPSS, Stata, R, Excel and CSV tables and describes their variables. Indicator catalogs such as PxWeb and .Stat Suite have no files at all, only a database and an API to it.

A database table is convenient for large and frequently updated tables and pointless for a map in PDF. Its cost is one more database to update and a loader that breaks on "dirty" files: an unexpected encoding, extra rows above the header, mixed types in one column.

The loader can be taken out of the publisher's hands. data.gouv.fr, as mentioned above, loads every tabular file into a database itself, and Hugging Face similarly converts datasets to Parquet. The publisher does nothing, but a "dirty" file still produces a "dirty" API. More in the chapter on leading catalogs.

A mix

Some datasets are stored, some link elsewhere. Most open data portals work this way: a register is uploaded to the catalog as a file, while sensors and maps stay on the agency's server. A mix is honest if the user sees which method applies to which resource, and the catalog checks the external links.

Immutability, versions and checksums

Your own storage makes it possible not to change a file after publication, but does not oblige you to. Platforms behave differently.

- **CKAN in classic mode** overwrites a resource file on update: the same address, new contents, and the previous contents are not kept. For an open data portal where the "current year budget" resource is updated every month, this is convenient. For data cited in research, it is

not. CKAN 2.12 introduced a new way of handling files: each file has its own identifier and checksum, and the contents of an existing file cannot be changed, only deleted and created anew.

- **Zenodo** allows files to be corrected within 30 days of publication. After that only a new version of the record with its own DOI, while the files of the old version remain available unless the author restricts them. Files older than 30 days can be changed only in exceptional cases: copyright infringement, disclosure of personal data.
- **Dataverse** creates a new version of the dataset when a file in a published dataset is replaced; the previous one remains. A checksum is calculated for each file, MD5 by default, with SHA-1, SHA-256 or SHA-512 as alternatives.
- **DSpace** stores a checksum for each file and can regularly verify it against the contents.

It is useful to expose the checksum too. DCAT 3 has the `spdx:checksum` property on a distribution for this, and data.gouv.fr shows the checksum in the resource description. Then a user, a harvester and an archive can check that they downloaded the same file and not one quietly corrected. How versions and identifiers work is covered in the chapter "Persistent Identifiers, DOIs and Versions".

Sizes and limits

Storage almost always has a file size limit, and people find out about it when a publisher arrives with a large archive.

Where	Limit	What to do beyond it
CKAN, default settings	10 MiB per uploaded file (`ckan.max_resource_size`)	raise the limit if there is enough space and bandwidth; put large files in object storage and link to them
Zenodo	50 GB and 100 files per record	a one-off increase to 200 GB on request; the rules forbid splitting a dataset to get around the limit
Harvard Dataverse	2.5 GB per file through the browser	larger by arrangement; on your own Dataverse the limit is set per store
GitHub	100 MiB per file; a repository is best kept under 1 GB	Git LFS, release attachments, separate storage
Hugging Face	for public datasets, space "best effort"; for private ones, 100 GB	see the chapter on ML catalogs

The CKAN default limit is especially treacherous: 10 MiB is a small table. A portal where the limit has not been raised quietly turns into a portal of links: publishers put everything larger on their own site. Limits and prices of research repositories are covered in detail in the chapter on research repositories.

For data in terabytes, the catalog usually stores the description, while the files themselves sit in object storage open for public reading, in formats that can be read in parts: Parquet for tables, Cloud Optimized GeoTIFF for imagery, Zarr for multidimensional arrays. The Registry of Open Data on AWS and Source Cooperative work this way; they are covered in the chapter on cloud storage and hubs.

When it is needed and when not

Situation	What to choose	Why
A research repository, data behind papers	own files, no overwriting, versions, checksums	a paper cites a specific file through a DOI; swapping breaks verifiability
A portal of an organisation whose data sits "on a shared drive"	own files	the publisher has no reliable place; there is nothing to link to
A national or regional portal harvesting agencies and municipalities	links with regular checking; copies of important datasets	duplicating everything is expensive and creates two diverging copies, but you need to be the first to learn a file has gone
A geoportal over map servers	links to services; files only for whole-layer downloads	the layer lives on the server; the catalog does not replace it
An indicator catalog	an indicator database and API, no files	the user takes a slice, not a file
Large tables filtered by programs and people	a database table with an API plus a whole-table download	partial access without downloading everything; the download is for those who need it all
A quarterly report in one file	just the file	a database table adds nothing
Terabytes of imagery, models, dumps	object storage, the description in the catalog	such files will not pass through the catalog server
Sensitive data	the file stays with the owner; the catalog holds the description and access procedure	see the chapter on access and roles

In brief, this is what a justification looks like using the decision record template:

- **Context.** A regional portal on CKAN harvests descriptions from 40 municipalities. Half the resources are links to their sites; some sites moved last year, and complaints about "broken links" arrive monthly.
- **Options.** Leave things as they are; move all files to the portal; keep the links but check them and store copies.
- **Criteria.** Share of broken links, cost of space and traffic, who updates the data, the risk of two diverging copies.
- **Decision.** The links stay; ckanext-archiver checks them every week and stores copies; a monthly report on broken links goes to publishers. Municipalities without their own site are offered direct upload to the portal.
- **Why.** Moving all files makes the portal responsible for the freshness of other people's data, while municipalities keep updating their sites, and the copies diverge. Checking with copies is cheaper and gives a list of problems by source.
- **When to revisit.** If the share of broken links does not fall within six months, or if the portal becomes the only legally mandated place of publication.

How it is done in platforms

The marks are core, extension, no, not checked, as in the summary table. Checked against documentation and repositories on 12 October 2026.

Platform	Own files	Cloud storage	Links elsewhere	Database table and API	File after publication
CKAN 2.12	core	extension: in 2.12, ckanext-file-keeper-cloud; earlier, ckanext-s3filestore and others	core	core (DataStore); CSV loading is the XLoader extension	overwritten; the new file handling in 2.12 does not change contents
Dataverse	core	core: S3 and compatible, Swift, Globus; several stores	core: trusted remote store	core: parsing tabular files, variable descriptions	replacement creates a new dataset version
InvenioRDM, Zenodo	core	core since v13: S3 and compatible	core: remote file, without guarantees	no	Zenodo: edits for 30 days, then a new version
DSpace	core	core: S3; several stores	no	no	a checksum is stored
Huwise (OpenDataSoft)	core: dataset attachments	stored by the provider	not checked	core: a dataset is a table of records	not checked
Socrata	core	stored by the provider	core: an "external data source link" asset	core	not checked
ArcGIS Hub	core	stored by the provider	core: a layer can live on an ArcGIS server	core (ArcGIS services)	not checked
GeoNetwork 4.4	core: record attachments in the server's data directory	S3, JCloud and CMIS modules shipped	core, the main mode	no	not checked
udata (data.gouv.fr)	core	not checked	core	separate open services hydra and api-tabular	not checked
PxWeb, .Stat Suite	no	—	no	core (PxWeb has its own API, .Stat has SDMX)	—
A static site, Git	files next to pages or by link	as you decide	as you decide	no, unless you build it	Git history keeps all versions

For cloud services, "stored by the provider" means the portal owner does not choose where the bytes live and cannot take them out other than by export. This must be reflected in the contract: the format and timing of an export on exit is as much a requirement as the price. More in the chapter "Build, Open Source or SaaS".

CKAN extensions and their risks

In CKAN, storage is the area with the most extensions and the fastest ageing. The method for evaluating an extension is in the chapter on extension ecosystems; here is what turned up during the check.

- **Cloud storage.** CKAN 2.12 introduced configurable storage, but only disk storage ships with it. Cloud adapters were moved to the CKAN community's ckanext-file-keeper-cloud; its tests run only on 2.12, and there are no releases on GitHub yet. For versions before 2.12 there are previous-generation extensions: Keitaro's ckanext-s3filestore is tested on 2.10 and 2.11 but not 2.12; ckanext-cloudstorage has not changed since February 2024 and has no automated tests; Datopian's ckanext-blob-storage is tested on 2.8 and 2.9, with the last release in January 2023. Moving to 2.12 for a portal with files in the cloud means changing the storage extension, not just upgrading.
- **A public bucket.** The CKAN 2.12 documentation warns: if resource files and organisation images sit in one store that is open for reading, files of private datasets are available to everyone without a permission check. Resources and public images get separate stores.
- **Loading tables into the DataStore.** The officially supported loader is XLoader, tested on 2.10–2.12. An alternative with type inference and computed description fields is datHere's DataPusher+; its tests run on CKAN 2.11, and release 3.0.0 came out in January 2026. The old DataPusher is no longer supported, and the CKAN documentation recommends moving off it.
- **Link checking and copies.** ckanext-archiver and ckanext-qa, which runs on top of it and rates format openness, are maintained by the CKAN community, with tests on 2.10–2.12.

On live portals these setups are visible at `/api/3/action/status_show`. On 12 October 2026, data.gov.gr on CKAN 2.11.3 kept files in Azure via cloudstorage and its own module, checked links with archiver and qa, and loaded tables with XLoader. opendata.swiss on 2.11.6 and HDX on 2.11.4 stored files in S3 via s3filestore and loaded tables with their own version of XLoader and with DataPusher+. Ireland's data.gov.ie ran CKAN 2.9.16 with the old DataPusher, a core version and loader that are no longer supported.

Publishing without a portal

Without a portal you choose the storage yourself, and there are more options than it seems.

- **Files next to the pages** of a static site, for small datasets: everything sits in one place and moves together with the site.
- **A Git repository**: the history of all versions for free, but a limit of 100 MiB per file. Large files go into release attachments or Git LFS. More in the chapter on Git repositories.
- **Zenodo or another general-purpose repository**, when you need an immutable copy with a DOI. A GitHub repository release can be archived in Zenodo automatically.
- **A public object storage bucket**, for large data; it needs a `README.md` file with a description and licence alongside. More in the chapter on cloud storage and hubs.

The rule is the same as for a portal: a file's address must not change, and a corrected file gets a new name or a new version. Checksums can be placed alongside in a `SHA256SUMS` file or recorded in `datapackage.json`.

How to check a live portal

A manual check takes ten minutes.

1. Open five to ten datasets from different organisations and click "Download". Look at the file's address.
2. If the domain matches the portal or is the portal's cloud storage, the file is held by the catalog. If the domain is different, it is a link to the source.
3. Open a few external links. See what opens: a file, a page listing files, the site's home page or an error.
4. If the dataset page has a table with a filter and a separate API address for its rows, table storage is enabled.
5. Look for a checksum or the date the link was last checked. If there is neither, the catalog does not know whether it is the same file or whether it is alive.

Measurement on four portals

On a large portal the same is done by a program. On 12 October 2026 a thousand datasets were taken through the CKAN API on each of four portals: ten blocks of a hundred datasets, spread evenly across the whole catalog. Then one random external resource was taken from each dataset, 300 per portal in total, and each link was opened. Failed requests were retried one at a time with a wait of up to a minute.

Portal	Datasets	Resources in the sample uploaded to the portal	Datasets with a DataStore table	External links: opened	Error or resource missing	Closed to programs	Page instead of file
data.gov.gr (Greece)	22,736	35%	3%	94%	5%	1%	1%
HDX (UN)	28,312	55%	7%	68%	6%	24%	2%
data.gov.ie (Ireland)	22,770	0%	41%	95%	3%	under 1%	1%
opendata.swiss (Switzerland)	16,521	0%	2%	94%	2%	0%	4%

"Error or resource missing" combines 404 and 410 responses, server errors, unreachable domains, timeouts and certificate errors. "Closed to programs" means 401, 403 and 429 responses: the file may exist, but a program does not get it. "Page instead of file" means the resource declares a file format (CSV, XLSX, PDF), but a web page opens. A sample of 300 links gives an accuracy of a few percentage points; it is a guide, not a census. All uploaded files that fell into the control sample (a hundred each on the Greek portal and HDX) opened.

What follows from the measurement.

Errors are concentrated in a few sources. On the Greek portal, 12 of 300 links failed because of a certificate error, and all of them lead to one map server of the municipality of Heraklion. On HDX, 73 of 300 links returned a 403 refusal, all to the UNHCR microdata library, which is closed to programs. So it is useful to aggregate link checking by domain: one conversation with one source removes most of the problems.

"Closed" and "deleted" are different problems. A file behind a bot check exists, and a person will download it. But a harvester, a web archive and a user's program will not get it, and for them it is just as lost. On the same day, Canada's open.canada.ca answered automated requests with

refusals and delays even at its own service API address, so it was not included in the table. More in the anti-pattern "The door is closed to programs".

A link that opens does not mean the data is there. Eight datasets in the sample on the Swiss portal declare CSV format, and the link in all of them leads to the same City of Bern page listing files. A link with PDF format to the statistiquedesrevenus.ch site redirects to a section of the Agroscope site: the old domain was redirected, and a section of the site opens instead of the document. A program that checks only the response code will count both links as working.

A well-assembled portal of links can be reliable. The Irish and Swiss portals have not a single file of their own in the sample, and obvious losses are under 3%. Their main sources, statistical offices and city portals, maintain their own data and do not change addresses. The data.gov.uk figure of a quarter of broken links is the result of years of collecting links to the sites of hundreds of local councils without checking, not an inevitable property of the link model.

Commands

For a CKAN portal. The first command takes 500 datasets and counts the share of uploaded resources, resources with a DataStore table, and the most common external domains:

```
PORTAL=https://data.gov.gr
curl -s "$PORTAL/api/3/action/package_search?rows=500&start=5000" \
  | jq '[.result.results[].resources[]] as $r | ($r | length) as $n | {
      resources: $n,
      uploaded: ([$r[] | select(.url_type == "upload")] | length / $n),
      in_datastore: ([$r[] | select(.datastore_active == true)] | length / $n),
      top_external_hosts: ([$r[] | select(.url_type != "upload") | .url |
capture("^https?://(?<h>[^/]+)").h]
        | group_by(.) | map({host: .[0], n: length}) | sort_by(-.n) | .[0:5])
    }'
```

The second opens external links from the same sample and prints the response code, content type and final address. A 200 code with type `text/html` for a resource declared as CSV is a reason to open the link and look:

```
curl -s "$PORTAL/api/3/action/package_search?rows=50&start=5000" \
  | jq -r '.result.results[].resources[] | select(.url_type != "upload") | .url' \
  | head -30 \
  | while read -r url; do
      curl -s -o /dev/null -L -m 30 -A "Mozilla/5.0" \
        -w "%{http_code}\t%{content_type}\t%{url_effective}\n" "$url"
    done
```

Not all servers answer a `HEAD` request, so the command makes an ordinary request and discards the response body. Run the check slowly, one link at a time: dozens of parallel requests to one site look like an attack, and you will get a refusal an ordinary user would not.

To find out which storage and link-checking extensions are installed on the portal:

```
curl -s "$PORTAL/api/3/action/status_show" \
  | jq '{ckan_version: .result.ckan_version,
         storage: [.result.extensions[] |
select(test("xloader|datapusher|archiver|qa|cloud|s3|blob|filestore|file_keeper"))]}'
```

For data.gouv.fr, check results are already recorded in each resource's description: the fields `check:status`, `check:available`, `check:date` and the checksum `analysis:checksum`. The share of files uploaded to the portal is visible from the `filetype` field: `file` is held by the portal, `remote` is a link.

```
curl -s "https://www.data.gouv.fr/api/1/datasets/?page_size=100&page=200" \
  | jq '[.data[].resources[]] as $r | ($r | length) as $n | {
      resources: $n,
      uploaded: ([$r[] | select(.filetype == "file")] | length / $n),
      checked: ([$r[] | select(.extras["check:status"] != null)] | length / $n),
      not_available: ([$r[] | select(.extras["check:available"] == false)] | length / $n)
    }'
```

How to turn shares of broken links into a quality metric and track it is covered in the chapter on quality metrics.

What follows from this

- Decide before launch where the bytes live: disk, cloud or links. Moving already uploaded files costs more than choosing on day one.
- If your files are cited for years, store them yourself, do not overwrite them after publication, and calculate checksums. A correction is a new version.
- If the catalog links elsewhere, check the links on a schedule, aggregate errors by source and keep copies of important datasets. Check not only the response code but also that a file opened and not a page.
- Raise the upload size limit to a realistic one: CKAN's default 10 MiB quietly turns a portal into a portal of links.
- Set up a database table with an API for tables people filter and query, and always keep a whole-table download alongside.
- In CKAN, choose storage extensions for your core version: when moving to 2.12, cloud storage changes extension.
- Give users a permanent file address on the portal, not a temporary cloud storage link.

Sources

- CKAN 2.12: file storage and configurable stores: the 10 MiB limit, cloud adapters in a separate extension, the public store warning, immutable files in the new handling; DataStore: PostgreSQL, result limits, dumps, XLoader, DataPusher+ and the end of DataPusher support
- CKAN extensions: ckanext-file-keeper-cloud, ckanext-s3filestore, ckanext-cloudstorage, ckanext-blob-storage, XLoader, DataPusher+, ckanext-archiver, ckanext-qa: automated test files and change dates, checked 12 October 2026
- Dataverse: file storage: S3, Swift, Globus, trusted remote store, direct upload, temporary links, checksums
- InvenioRDM: S3 storage (since v13) and file transfer types: local, fetch, multipart, remote
- Zenodo: managing files: 50 GB and 100 files, edits within 30 days
- DSpace: storage layer: disk and S3, several stores, checksums, traffic costs

- GeoNetwork: storage modules in release 4.4.12
- Socrata: linking to an external data source
- data.gouv.fr: the hydra crawler and the tabular API; check fields in the datasets API, checked 12 October 2026
- Measurement on 12 October 2026: the CKAN APIs of data.gov.gr, HDX, data.gov.ie, opendata.swiss, 1,000 datasets and 300 external links per portal; the `status_show` address for versions and extensions; HDX's redirect to a temporary S3 link
- I. Begtin, Good and bad data publication practices (in Russian), November 2024: the share of broken links on data.gov.uk in the 2023 dump
- Congressional Research Service, Data.gov: Implementation and Perspectives on Its Functions, R48954, 21 May 2026: registry or repository, preserving access to data
- GDS, What's changing on data.gov.uk and why, 25 March 2026: at least 25% of links lead to an error
- Anatomy of a Catalog, APIs and Bulk Download, publishing without a portal: static sites: data.gov.cz as a portal without files
- Leading catalogs: data.gouv.fr's tabular API and Hugging Face's conversion of datasets to Parquet

Persistent Identifiers, DOIs and Versions

Data gets cited: in a paper, in a report, in another catalog, in someone else's program. A link is useful only if a year later it opens the same dataset. Hence the chapter's three practical conclusions.

Every catalog needs a permanent page address, and it costs almost nothing if planned before launch: the address should contain a key that does not change when the description is edited or when the catalog moves to another platform. A DOI is needed where data is cited as a source. It is not a button but an agreement with a registration agency and a commitment to maintain, for years, the page it points to. A version is a promise that the link will open the data the author used, not today's dump.

The main thing the checks at the end of the chapter show: an identifier is no more reliable than the site it points to. In DataCite, more than five hundred "findable" dataset DOIs lead to `localhost`, the address of the computer on which they were set up. In four out of ten DSpace installations checked, the persistent identifier printed on the record page does not resolve.

What it is

Address and identifier. An address says where a page is: `portal.example/dataset/budget-2024`. An identifier says what the object is, regardless of where the object is now. A persistent identifier has three parts: a name, a service that redirects from the name to the current address (called a resolver), and an owner who commits to updating that address at every move. Without the third part the first two do not work: the service will faithfully redirect wherever it was told to, even if nothing has been there for a long time.

There are three mechanisms, and they are often confused.

A permanent page address. A dataset's address does not change because of a typo fix in the description, and does not contain today's date or a database row number. When the site changes, the old address redirects to the new one. Everyone needs this, and the catalog itself provides it.

An external persistent identifier. DOI, Handle, ARK and the like: the name is registered with an external service, which redirects to the dataset's current page. A link of the form `https://doi.org/10.xxxx/yyyy` survives a change of domain, as long as the owner remembers to update the address in the service.

A version. A fixed state of a dataset. A scientific citation points to the version used in the analysis. An open data portal more often maintains one dataset that is updated: what matters is the update date and an archive of past dumps, not a new DOI every Monday.

Kinds of identifiers

Identifier	Who issues it and what it costs	What is mandatory	Where it is found
DOI	registration agencies: DataCite for data, Crossref for publications, and others. The member organisation pays, usually through a consortium	a description following the agency's schema: DataCite has six mandatory fields	research repositories, journals, Zenodo, Figshare, some geocatalogs
Handle	Handle.Net Registry; a prefix for an annual fee, your own Handle server	nothing	DSpace by default, Dataverse as an option. Technically a DOI is also a Handle with the prefix `10.`
ARK	ARK Alliance; an organisation number (NAAN) is free for libraries, archives and similar "memory organisations"	nothing	national libraries and archives, for example the National Library of France
PURL, w3id.org	free redirect services: purl.org is run by the Internet Archive, w3id.org by a W3C community; rules are added through a GitHub pull request	nothing	linked data vocabularies and ontologies, permanent addresses for small projects
CSTR	China's national system	a description of the object	Chinese research repositories, often together with a DOI
Installation permalink	the platform itself, without an external service	nothing	Dataverse (PermaLinks), the record identifier in any catalog

Next to the dataset identifier work identifiers for the people and things around it: ORCID for the author, ROR for the organisation, IGSN for a physical sample, RAiD for a research project. Since the 2020s IGSN and RAiD have been issued as DOIs through DataCite: its repositories include 40 sample catalogs and 17 project registries. Through these links aggregators build a graph: who created a dataset, in which organisation, under which project and in which paper it was used.

Why it is needed

Links die fast. According to a 2024 Pew Research Center study, 38% of pages that existed in 2013 no longer opened by October 2023. 21% of pages on US government websites contain at least one broken link, most often on city websites. The same goes for datasets: in the full data.gov.uk dump of 2023, about 10% of links to files did not open; more in the chapter on storage.

Links are broken by people, not technology. So Tim Berners-Lee wrote in "Cool URIs don't change" (1998), and since 1999 W3C has kept all its documents at their original addresses under its own persistence policy. The opposite example is Russia's register of domestic software in 2021: after a move, old links stopped opening, and the new addresses contained the technical database record number (330494) instead of the program's register number (10269). At the next database change such an address will break again. Short links are not permanent either: their fate is decided by the service owner, as the goo.gl story in the chapter on archiving shows.

So that data can be cited and the citation counted. A DOI makes a dataset a full-fledged source: it goes into the reference list, citations are counted by it, and aggregators use it to link the dataset to a paper and an author. Without an identifier, a link to data remains a website address that nobody counts. How this becomes a ready citation string is covered in the chapter on citation export.

To match records. Registers, reference lists and classifications serve other data as a point of reference. If a record has no stable identifier, you cannot reliably tell that a row in last year's file and a row in the new one are the same organisation. This is covered in the section on registers below.

How a DOI works

A DOI consists of a prefix and a suffix: `10.5281/zenodo.16546069`. The prefix `10.5281` is assigned to an organisation; it makes up the suffix itself. The doi.org service responds to a request with a redirect to the address the owner recorded at registration. This address is called the landing page: it has the dataset description, a citation and a path to the files. The DOI itself stores no files.

Agencies. For data, DOIs are registered by DataCite, an association registered in Germany. On 12 October 2026 DataCite held 138.3 million DOIs, 76.4 million of them with the type "dataset". What lies behind that number is covered in the chapter on the world of catalogs in numbers. Crossref, a US agency, registers DOIs for articles, books and preprints. There are other agencies too, for example the Publications Office of the EU. Starting with InvenioRDM v14, and in Dataverse, DOIs can also be registered through Crossref, but for datasets the usual choice is DataCite.

A description is mandatory. Registration with DataCite requires six fields: identifier, creator, title, publisher, year, resource type. Without a creator and a year, a DOI cannot produce a citation. What else is worth filling in is covered in the chapter on metadata completeness.

A DOI cannot be deleted. A DataCite record has three states. A draft is not yet registered in the global system and can be deleted. A "registered" DOI works, but its description is hidden from search. A "findable" one works and is visible to everyone. From the last two states there is no way back. If a dataset has to be withdrawn, DataCite advises leaving a tombstone page in its place: the citation, the DOI and an explanation of why the data is no longer available. For those who cannot make such a page, DataCite has a shared placeholder page; on 12 October 2026, 5,145 findable DOIs led to it.

What it costs. DataCite does not charge per DOI as long as an organisation registers fewer than 100 thousand a year. Payment is for membership: a direct member pays 2,000 euros a year plus an infrastructure fee of 200 to 700 euros depending on the country. An organisation inside a consortium does not pay the membership fee, only the infrastructure fee. That is why universities usually get DOIs through a national or library consortium, and small projects through someone else's repository such as Zenodo, where a DOI is free for the author.

What a DOI does not guarantee

That the data will be preserved. A DOI leads to a page, and the owner keeps the page. The study "Open is not forever" (2020) counted 176 open access journals that vanished between 2000 and 2019, with and without DOIs. An identifier helps find a trace but does not replace an archive. How to preserve the data itself is covered in the chapter on archiving.

That the address is correct. The doi.org service redirects wherever the owner recorded and does not check whether anything is there. On 12 October 2026, 765 findable DOIs in DataCite led to `localhost`, 548 of them datasets. Such an address opens only on the computer of whoever

recorded it. One cause is visible in the GeoNetwork documentation: as a template for the landing page address it suggests `http://localhost:8080/geonetwork/srv/resources/records/{{uuid}}`. Portugal's Institute for Sea and Atmosphere, IPMA, registered two DOIs with such an address in 2025, one of them titled "Copy of template MFSD Template…", that is, a test copy of a template.

That there is a dataset behind the DOI. DOIs are also issued for individual records of a large database: a species observation, a crystal structure, a single experiment. So the number of DOIs with the type "dataset" cannot be read as the number of datasets; see the chapter on the world of catalogs in numbers.

That the agency will always work with you. DataCite and Crossref comply with their countries' sanctions. Crossref does not accept applications from organisations that are part of the Russian government or are under US, UK or EU sanctions; other applications from Russia and Belarus undergo a separate check, and after 2022 every Russian member was checked against sanctions lists and some were removed. DOIs already issued continue to resolve, but an organisation may be unable to get new ones. China has built a parallel system: the national CSTR identifier and the Science Data Bank repository, which at the same time issues DOIs through DataCite (28,406 records as of October 2026) and provides descriptions to European and American aggregators.

Versions

What counts as a new version. The data changed: a new version. A typo in the description was fixed or a keyword added: a revision, which does not create a new version. The line is whether a reader would get a different result by repeating the analysis.

In a research repository a published version is not rewritten. A correction is a new version with its own DOI, or with the same DOI and a new number, according to the repository's rules. Platforms do this differently, and the difference matters for citation.

Model	How it works	Where
A DOI for each version and a concept DOI	each version has its own DOI, plus a concept (parent) DOI that always leads to the latest version	InvenioRDM and Zenodo, Figshare (a concept DOI and DOIs with the suffix `.v1`, `.v2`)
One DOI and a version number	one DOI for the whole dataset; the version (1.0, 1.1, 2.0) is given in the citation	Dataverse
A DOI with a version number	the DOI of a new version is the original DOI with a number after a dot	DSpace with item versioning enabled
One updated dataset and dated snapshots	the dataset address is permanent, the file is replaced; past dumps sit in an archive with a date	open data portals, registers, reference lists
A dataset series	each release is a separate dataset with its own address, and the series groups them	the `dcat:DatasetSeries` class in DCAT 3, annual statistics

In InvenioRDM and Figshare a paper can cite the DOI of a specific version, while the concept DOI remains for those who always need the latest. In Dataverse a paper cites the DOI and the version number together. Since version 13 InvenioRDM can make a DOI optional, but by default it will not

publish a version without a DOI if the previous one had one: otherwise the concept DOI could lead to a version without an identifier.

One link, different contents. On an open data portal a link to a file usually leads to the current file. In CKAN the address of an uploaded file is built from the resource identifier, and the file name in it has no effect: replace the file and the old link downloads the new one. For a portal this is convenient; for a scientific citation it is not: a year later the link will give different data. Two things help. A checksum shows that the file has not changed: DCAT 3 has a distribution property for it, and Dataverse additionally calculates a UNF for tables, a checksum of the contents independent of the file format. And dated snapshots: if the file may already have been cited, the previous variant stays in an archive.

How metadata describes a version. DCAT 3 (2024) introduced version properties: the number (`dcat:version`), what changed (`adms:versionNotes`), previous and current versions (`dcat:previousVersion`, `dcat:hasCurrentVersion`). In DataCite, versions are linked with the relations `IsNewVersionOf`, `IsVersionOf`, `HasVersion`. CKAN has a "version" field, and ckanext-dcat exports it to DCAT, but it is just a label: CKAN does not store past states of a dataset as separate objects with their own addresses.

On an open data portal there is usually one dataset, and files are replaced. Then you need:

- the update date and frequency in the description;
- an unchanged dataset address when the file is replaced;
- an archive of previous files with dates, if they may already have been cited.

Registers and reference lists

A register of licences, a list of organisations, a classification of economic activities: datasets that other data and programs refer to. For them, not only the dataset but every record needs a permanent address. They are maintained in one of two ways.

- **A versioned reference list.** Published in full; each version is a separate dated file. Classifications and address databases are maintained this way: the user takes the version for the date needed and can compare two versions.
- **A record-keeping system with record history.** Records change one at a time, and each keeps its edit history. This suits registers with daily changes better: licences, permits, statuses.

In both cases, a register that is convenient to use has the same things:

- a unique identifier for each record, assigned under the register's rules, not a row number in a file or a technical database number;
- a permanent link to the whole register and to each record, built from that identifier;
- records are not deleted but get the status "removed" with a date;
- past versions of a record or of the whole register are visible;
- the basis for the record is given (a decision, an order, a document) with a link;
- the register is downloadable in full, as regular dated snapshots, and is available through an API.

A published register without identifiers and history is a snapshot that cannot be reliably matched with other data: a record may have been renamed, removed or entered with an error, and the file does not show it. Regular snapshots are also cheaper for the system itself: exporting a whole register on demand from a large database easily brings the server down, while a ready file once a day does not.

Identifiers of objects within data

The same approach works for data that is not called a register. In Overture Maps, every building, road, place and administrative unit in the monthly release has a persistent GERS identifier. Alongside are kept an identifier registry with dates of first and last appearance, a per-object changelog and crosswalk tables to source identifiers. Another provider can publish only its own indicators (traffic, population) without geometry, and they are joined by identifier.

The lesson is there too. In release 2025-06-25.0 Overture replaced the previous identifiers, which encoded an H3 grid cell and a theme, with random UUIDs, and the old numbers can no longer be found. An identifier with meaning built in breaks when the meaning changes. In OpenSanctions, after deduplication an entity has one canonical identifier, and if it matched a Wikidata record, the Wikidata number. More in the chapter on leading catalogs.

Implementation options

A permanent address provided by the platform

The main question is what key sits in the dataset address. A short name derived from the title is easy to read, but there is a temptation to change it together with the title. A number or code assigned once is harder to read, but it does not change.

Platform	What is in the address	What happens on renaming
CKAN	the dataset's short name (`/dataset/budget-2024`); the dataset also has an unchanging UUID	the address with the old name returns "not found". The UUID address always redirects to the current name, so use it for citations
Socrata	a code of two groups of four characters such as `erm2-nwe9` , plus the title	the title in the address is decorative: any path with the code redirects to the current address
Huwise (OpenDataSoft)	a readable dataset identifier; there is also an internal one that never changes	the readable identifier can be changed only before publication
Dataverse	a persistent identifier (DOI, Handle or installation permalink)	independent of the title
DSpace	Handle	independent of the title
GeoNetwork	record UUID	independent of the title
InvenioRDM	record identifier	independent of the title; each version has its own

Checked on 12 October 2026: on data.gov.gr the UUID address of a dataset redirects to the name address, and on New York's Socrata portal a path with an arbitrary title and the dataset code leads to the current page.

Your own permanent address without a DOI

If a DOI is not needed but the address must survive a change of platform, there are inexpensive routes.

- **Redirects on your own domain.** Dataset addresses are built from a domain the organisation does not intend to change, and at each move a table of "old address → new" redirects is set up on the web server. The cheapest method and the only one everyone needs.
- **Handle.** A prefix is obtained from Handle.Net Registry for an annual fee, and you run your own Handle server. DSpace ships with one, but with the placeholder prefix `123456789`, which the global system does not know. Until the prefix is registered and the server is running, a Handle on the pages is only for show.
- **ARK.** An organisation number is issued free of charge; ARKs can be resolved on your own server or through the shared resolver n2t.net. Suitable for libraries, archives, museums.
- **w3id.org and PURL.** Free redirect services. On w3id.org, a rule for your namespace is added through a pull request to the GitHub repository. Suitable for small projects and static sites that need an address not tied to hosting.

DOI

In the platform core. Dataverse, InvenioRDM, DSpace, Figshare and GeoNetwork register DOIs themselves. You need an agreement with an agency, a prefix and a test account. InvenioRDM and DSpace are connected to the DataCite test system by default so as not to mint real DOIs by mistake. A mistake made in the production system is hard to fix: DOIs are not deleted.

Through an extension. CKAN has no DOIs in its core. The ckanext-doi extension from the Natural History Museum in London maps dataset fields to the DataCite schema and registers DOIs. For the risks see the section on extensions.

Manually or through a consortium. An organisation with a prefix can register DOIs in the DataCite Fabrica web interface or through its API, giving the page addresses of any platform, including a cloud one. This is done when a DOI is needed for a dozen datasets, not for every one. Consortia offer this as a service: for example, Australia's ARDC issues DOIs for data of public research organisations, but warns plainly that maintaining the addresses is up to them.

In someone else's repository. A copy or release of the data goes to Zenodo or Figshare and gets a DOI there. This is done with GitHub repositories: a release is archived in Zenodo automatically. More in the chapter on Git repositories. The one drawback: the DOI leads to the copy in Zenodo, not to your portal, and the copy has to be updated separately.

When it is needed and when not

Situation	What is needed	What usually covers it
A research repository; a funder or journal requires a data citation	a DOI for each published version, authors' ORCIDs, a tombstone page on withdrawal	Dataverse, InvenioRDM, DSpace with DataCite, Figshare, Zenodo
An open data portal of a ministry or city	a permanent address with an unchanging key, redirects on moves, an update date; no DOI needed	any platform, if you choose the key in the address

Situation	What is needed	What usually covers it
A transport timetable, operational data that changes daily	a permanent dataset and feed address; a DOI is superfluous	any platform
A register or reference list	an identifier for each record, dated snapshots, history	your own record-keeping system and dumps; a portal for snapshots
A geoportal with ISO records	the record UUID in the address; a DOI if layers are cited in research	GeoNetwork with DOI enabled
Individual datasets cited in the organisation's reports	DOIs for those datasets only	Fabrica under a consortium agreement, or a copy in Zenodo
Publishing without a portal	an address not tied to hosting, a DOI per release	a Git repository and Zenodo, w3id.org

Issue a DOI when the dataset will be cited as a source and the organisation is ready to maintain landing pages for years. Do not issue a DOI "for respectability": a dataset that changes daily with a DOI misleads, as the link promises a fixed state and opens today's.

In brief, this is what a justification looks like using the decision record template:

- **Context.** A university runs a portal on CKAN 2.12 for departmental data. The foundation that funds most projects has started requiring DOIs for data behind reports.
- **Options.** Install ckanext-doi; set up Dataverse or InvenioRDM alongside for cited data; send such datasets to Zenodo; register DOIs manually through a consortium.
- **Criteria.** Immutability of the published version, support for the extension on the current CKAN version, maintenance cost, where the link will point: to your own portal or to someone else's repository.
- **Decision.** Set up Dataverse alongside CKAN for data behind publications; keep CKAN for operational and teaching datasets; get DOIs through the library consortium.
- **Why.** CKAN does not store versions: a DOI on a replaceable file does not give the immutability the foundation requires it for. ckanext-doi has not been tested on CKAN 2.11 and 2.12. Zenodo is simpler, but links lead away from the university portal, and copies have to be updated manually.
- **When to revisit.** If ckanext-doi starts being tested on the current CKAN version and DOIs are needed only for a small number of unversioned datasets, or if there is so little data behind publications that Zenodo is enough.

How it is done in platforms

The marks are core, extension, no, not checked, as in the summary table. Checked against platform documentation on 12 October 2026.

Platform	DOI	Versions	What is important to know
Dataverse 6.12.1	core: DataCite, Crossref (since 6.12.1), EZID, Handle and installation permalinks; several providers on one installation	core: major and minor versions (1.0, 1.1, 2.0), one DOI per dataset	files can be given their own identifiers, either dependent on the dataset DOI or independent

Platform	DOI	Versions	What is important to know
InvenioRDM v14, Zenodo	core: DataCite, Crossref since v14; a DOI issued elsewhere can be accepted	core: each version has its own DOI plus a parent DOI (since v12)	since v12 it does not register DOIs for restricted records; since v13 a DOI can be made optional
Figshare	service core: DataCite	core: a concept DOI leads to the latest version, versions have the suffix `.v1`, `.v2`	a cloud service; self-hosting is not possible
DSpace 10	core, enabled in configuration: DataCite or EZID, in addition to Handle	core: item versions; a version DOI is the original with a number after a dot	the item address is still built from the Handle; DOIs are sent to DataCite by a separate scheduled task; a filter decides which items get a DOI
GeoNetwork 4.4	core, enabled in settings: DataCite or the EU Publications Office API; only for a public record	no: with approval workflow enabled there is a working copy of a record and a status history, but not versions with their own identifiers	the DOI is assigned to the layer description, not to the data: a layer on a map server may change
CKAN 2.12	extension ckanext-doi: CKAN 2.9 and 2.10 declared	no: the activity stream and the "version" field are not versions with their own addresses	use the dataset's UUID address for citations
NADA	no: only a field for a DOI issued separately	not checked	citation via DataCite works if there is a DOI
Huwise, Socrata, ArcGIS Hub	no	no	a permanent address uses the dataset's or item's unchanging code; a DOI can be minted manually and entered in a custom field
Hugging Face, Kaggle	platform core: DOI through DataCite at the press of a button	Hugging Face: Git repository revisions; a DOI is issued for the current revision, a new one is minted for a new revision and the old one is marked outdated	after a DOI is issued, a dataset on Hugging Face cannot be deleted or renamed without support; more in the chapter on ML catalogs
A static site, a Git repository	through Zenodo: a DOI per release	release tags in the repository	a link to a file on a branch shows the current state; a link to a release tag shows a fixed one

Extensions and their risks

ckanext-doi is the only route to DOIs inside CKAN, and as of October 2026 a risky one. The latest release, 4.0.4, came out on 22 September 2025, with no changes since. The description declares CKAN 2.9 and 2.10. Automated tests run in the museum's own prebuilt images, and those images were last updated on 20 September 2023, before CKAN 2.11 came out. So the extension has not been tested on 2.11 and 2.12 at all. A portal that needs DOIs on current CKAN either takes testing upon itself or issues DOIs outside CKAN. The state of other CKAN extensions is in the chapter on CKAN.

DOI modules in the core also need attention during upgrades. In DSpace, registration is split into two steps: an item is flagged in the database, and sending to DataCite is done by a separate scheduled command. If the scheduled task was not moved to the new server during an upgrade,

DSpace considers the DOIs issued, but they are not in DataCite, and the links do not resolve. In GeoNetwork the landing page address template is set manually, and the `localhost` error described above is linked to it.

What the registry shows

According to the dataportals-registry export (release 1.27.0 of 15 October 2026), the "DOI" mark is set for 2,008 of 43,614 catalogs. Of these, 1,404 are research repositories and 584 are geoportals; the rest are isolated cases: 7 indicator catalogs, 3 open data portals and 3 microdata catalogs.

The number must be read with care. The mark in the registry is mostly set by platform: if the software can issue DOIs, all its installations without a manual "no" get the mark. That is why 555 of the geoportals with DOI are GeoNetwork installations. In DataCite itself on 12 October 2026, 7,209 DOIs led to addresses with the word `geonetwork` in the path, and they belong to about 45 installations; more than three thousand belong to the French hydrographic service SHOM and the Austrian surveying agency BEV. Even allowing for installations with a different address, this is an order of magnitude less than 555. The registry shows who can issue DOIs, not who does. The queries are in the appendix "Using the Registry".

Publishing without a portal

Without a portal, address persistence is entirely up to the author, but the methods are simpler.

- **A release and a DOI.** A GitHub repository is connected to Zenodo, and every release gets its own DOI, while the whole series of releases gets a concept DOI. A `CITATION.cff` file at the root tells people how to cite. How to set this up is covered in the chapter on Git repositories.
- **A link to a version, not a branch.** A link to a file on the `main` branch always shows the current state. For a citation in a paper, use the address of a release tag or the release DOI.
- **An address not tied to hosting.** If a dataset lives on a static site, an address like `https://w3id.org/your-project/` will survive a move from GitHub Pages to other hosting: only the redirect rule changes.

How to check a live portal

- Change a dataset's description on a test instance, including the title. The page address must not change, and if it does, the old one must redirect.
- Find a permanent link on the page. If there is none, see whether the address contains an unchanging key: a UUID, a code, a number.
- If a DOI is promised, it is visible on the page, and `https://doi.org/...` opens this same page, not the home page and not `localhost`.
- If the page shows a Handle, check it through `hdl.handle.net`. The prefix `123456789` is the placeholder shipped with DSpace.
- Open a previous version of the dataset and check that its number appears in the suggested citation.
- After a migration, the old address has a redirect, not a "not found" page.

Commands

Where a DOI leads and what DataCite knows about it:

```
curl -sI https://doi.org/10.5281/zenodo.16546069 | grep -i '^location'
curl -s https://api.datacite.org/dois/10.5281/zenodo.16546069 \
  | jq '.data.attributes | {url, state, publisher, version, related: [.relatedIdentifiers[]?
| select(.relationType | test("Version"))]}'
```

The first response should be a redirect to the dataset page. The second shows the address the owner recorded, the state and the links to other versions.

How many DOIs a repository has, and whether any of them have addresses that will open nowhere but the owner's computer. The repository identifier in DataCite (`client-id`) is visible in the response to the previous command, in the `relationships` section:

```
CLIENT=cern.zenodo
curl -s "https://api.datacite.org/dois?client-id=$CLIENT&page%5Bsize%5D=0" | jq
'.meta.total'
curl -s -G https://api.datacite.org/dois --data-urlencode "client-id=$CLIENT" \
  --data-urlencode 'query=url:*localhost*' --data-urlencode 'page[size]=0' | jq
'.meta.total'
```

Whether a Handle printed on a DSpace record page resolves:

```
curl -s https://hdl.handle.net/api/handles/1234/5678 | jq '.responseCode'
```

Code `1` means the Handle exists, `100` not found, `2` an error on the Handle server side; an empty response means the server did not answer.

Whether an address on CKAN is stable: the UUID address should redirect to the name address.

```
PORTAL=https://data.gov.gr
ID=$(curl -s "$PORTAL/api/3/action/package_search?rows=1" | jq -r '.result.results[0].id')
curl -s -o /dev/null -w '%{http_code} %{redirect_url}\n' "$PORTAL/dataset/$ID"
```

Three checks on a sample

The DataCite checks were done on 12 October 2026; for the registry, release 1.27.0 of 15 October was used.

Do dataset DOIs lead to live pages? A random sample of DOIs says nothing about practice: half of it comes from one Japanese institute that registers every experiment. So 700 random DataCite repositories were taken; 314 of them had datasets, and one random dataset DOI was checked from each.

Result	Repositories
the page opened	233
bot protection (403, browser check), cannot be checked by a program	55
dead link: "not found", a non-existent domain, a private address	16
server error or no response twice in a row	10

Among the dead links are DOIs that led nowhere from the start: to the private address `10.4.0.146`, to an IP address with port 8080, to the `nhsdigital-dev.collibra.com` test instance of an enterprise catalog. The rest are traces of moves: "not found" for old B2SHARE records and for 2018 NOAA geoportal pages, non-existent domains for four university repositories. One in twenty repositories with datasets serves at least some DOIs that do not resolve, and for another one in six a program cannot check.

Do Handles in DSpace resolve? 120 random active DSpace installations with OAI-PMH were taken from the registry, and 86 responded. Using the identifier of the first record, it was checked whether its Handle resolves through hdl.handle.net. For 51 installations, yes. 12 have the shipped placeholder `123456789`, and 7 have an alphabetic local prefix such as `item` or `edata` that means nothing outside the site. Another 16 have a numeric prefix, but the Handle is not found or the Handle server does not respond. In total, for 35 installations out of 86 the persistent identifier on the record page is only for show.

How many geocatalogs actually issue DOIs? The comparison of the registry and DataCite is described above: 555 GeoNetwork installations marked "DOI" versus about 45 installations to which DOIs actually lead.

What follows from this

- Plan a permanent address before launch: an unchanging key in the dataset address, and for citations an address that does not depend on the title.
- Prepare the "old address → new" table and redirects before a move, not after. A DOI saves only the datasets that were given one in advance.
- Issue DOIs when data is cited as a source and you are ready to maintain landing pages for years. A dataset that changes daily does not need a DOI.
- If you issue DOIs, issue them for immutable versions, and when withdrawing data leave a tombstone page: a DOI is not deleted.
- Check the addresses in your DOIs and Handles with a program: not a test instance, not `localhost`, not a shipped placeholder.
- Check a DOI extension by its tests for your core version. ckanext-doi has not been tested on CKAN 2.11 and 2.12.
- Remember that a DOI is an agreement with an agency in another jurisdiction. Keep your own permanent address as well: it will remain even if you cannot get new DOIs.

Sources

- DataCite: DOI states, tombstone pages, landing pages, versioning, fee model, REST API; queries on 12 October 2026: 138.3 million DOIs, 76.4 million datasets, 765 DOIs pointing to `localhost`, 5,145 to the shared placeholder page, 7,209 to GeoNetwork addresses, a sample of 314 repositories
- Crossref: sanctions compliance, removal of members
- Handle.Net Registry: prefix registration, ARK Alliance, w3id.org, PURL at the Internet Archive, CSTR

- Dataverse 6.12: persistent identifier providers, dataset versions, data citation
- DOIs in InvenioRDM: Crossref since v14, parent DOI since v12, optional DOI since v13, restricted records
- Versioning and DOIs in Figshare
- DSpace 10: DOI, the default Handle prefix in dspace.cfg, Handle.Net Registry support
- GeoNetwork 4.4: DOI configuration, record life cycle and working copy
- CKAN 2.12: dataset view and resource download in dataset.py, resource.py; ckanext-doi and its test images; checked 12 October 2026
- Huwise: dataset identifiers in the Automation API
- DCAT 3: version properties, dataset series, distribution checksum
- Pew Research Center, When Online Content Disappears, May 2024
- T. Berners-Lee, Cool URIs don't change, 1998; W3C URI persistence policy
- M. Laakso, L. Matthias, N. Jahn, Open is not forever: a study of vanished open access journals, 2020
- ARDC: Australia's national persistent identifier strategy, DataCite DOI service
- The dataportals-registry export, release 1.27.0 of 15 October 2026: the DOI mark on 2,008 catalogs, a sample of 120 DSpace installations with OAI-PMH
- I. Begtin, Telegram (in Russian): the domestic software register and link rot, 15 April 2021; Crossref, sanctions and CSTR, 23 February 2023; Australia's identifier strategy, 24 March 2023; vanished open access journals, 19 September 2020
- I. Begtin, State registers (in Russian), April 2021: two models of maintaining registers and requirements for them
- Citation Export, Metadata Completeness, Catalog Lifecycle, Archiving
- Overture Maps, GERS: identifier registry, changelog, crosswalk tables, replacement of identifiers with UUIDs in release 2025-06-25.0; OpenSanctions, the statement model: a canonical identifier after deduplication

Citation Export

People cite data when it takes a second. If a citation has to be assembled by hand from the title, year and address, most will cite the paper or nothing at all.

The main conclusion of the chapter: a "Cite" button is cheap; what it builds the citation from is expensive. A generator will instantly assemble a string from description fields, but if they lack an author, a year and a version, the result is a neatly formatted dud. So first check the fields, and check that a cited dataset will be the same a year later. The button comes afterwards. If a dataset has a DOI, DataCite already serves its citation in a dozen formats, and the catalog's job is not to spoil it with its own version. The check in this chapter shows that this does happen: for the same dataset, the catalog and DataCite produce different citations.

What it is

Citation export is everything that helps cite a dataset without manual work. It has four layers, and platforms cover them differently.

A string on the page. A "How to cite" block with a ready citation that is copied into the text of a paper or report.

A file for a reference manager. A reference manager is a program in which a researcher keeps their sources and from which they insert citations into a text: Zotero, EndNote, Mendeley, JabRef. It is given the citation as a file in one of several formats: BibTeX (for LaTeX), RIS and EndNote XML (for most managers), CSL-JSON (for programs that format a citation according to a CSL style). A CSL style is a description of formatting rules: APA, Chicago, GOST, the requirements of a particular journal.

Hints for programs on the page itself. Meta tags in the HTML head (`citation_title`, `citation_author` and others, invented by Google Scholar, or Dublin Core fields). With them Zotero saves a dataset with one button in the browser, and Google Scholar understands what it is looking at. This also includes the `Link` response header with a `rel="cite-as"` link: it tells a program which address to treat as the address for citation. This is part of FAIR Signposting, briefly described in the glossary.

A citation by identifier, without the catalog. If a dataset has a DOI registered with DataCite, its citation can be obtained from DataCite itself, without visiting the catalog's site. This technique is called content negotiation: a program requests the `https://doi.org/…` address and asks in the `Accept` header not for a page but, for example, for BibTeX. The commands are in the section on checking.

Two things are often confused with citation export.

- The `citation` property in schema.org markup is not a citation of the dataset itself but the papers the author asks to be cited along with it. Google says plainly: to cite the dataset, use the name, identifier, creator and publisher, and `citation` is for related papers.

- The CC BY licence requirement to credit the author is a legal condition of use, not a scholarly citation. The FORCE11 data citation principles distinguish the two. But a ready string helps fulfil both: GBIF, for example, under its user agreement obliges users to cite a download by its DOI, and the Ifremer catalog states in a dataset's terms that citation is mandatory.

Why it is needed

So that the data author gets credit. The FORCE11 Joint Declaration (2014) set out eight principles of data citation. The first: data is a legitimate research output, and citing it carries as much weight as citing a paper. For a catalog, three others matter most: a unique identifier, persistence of the citation, and specificity, meaning the citation must point to the version or subset of data on which the conclusion was based.

So that the citation can be counted. The Make Data Count project together with DataCite collects the Data Citation Corpus. Its fourth release of 27 July 2025 has 10.7 million records, 9.7 million of them unique dataset–publication pairs. Almost half the records, 5.2 million, were added by Europe PMC, a collection of life sciences papers. Its software finds datasets not in reference lists but in the text of papers, where a dataset is mentioned by an accession number in a database, for example a genetic bank. Another part of the corpus came from similar software annotation of papers by the Chan Zuckerberg Initiative. So data is cited more often than reference lists show, but counting such citations is hard and imprecise. A tidy citation with a DOI in a reference list is what gets counted without guesswork.

How few counted citations there are can be seen in DataCite itself. On 12 October 2026, three thousand random DOIs with the type "dataset" were taken from its API. More than half of them (1,667) were minted by Japan's National Institute for Fusion Science (NIFS), so it is better to count without it. Of the remaining 1,333 DOIs, 78, about 6%, have at least one citation known to DataCite.

So that the catalog owner sees the benefit. Citations in papers and reports are the most convincing argument for management and funders that the catalog is needed. A download counter says a file was taken; a citation says something was made from it.

So that the user does not get it wrong. Without a ready string everyone assembles a citation themselves: some write the portal name instead of the publisher, some the download date instead of the publication year, almost nobody writes the version. With such a citation the dataset cannot be found, or if it is found, it is the wrong one.

What a citation consists of

DataCite recommends this order of elements:

```
Creator (Year). Title. Version. Publisher. Resource type. Identifier
```

A minimal citation omits version and type: creator, year, title, publisher, identifier. This is what a citation of a real Harvard Dataverse dataset looks like in APA style:

```
Durbin, P. (2017). Open Source at Harvard (Version 5.0) [Dataset]. Harvard Dataverse.
https://doi.org/10.7910/DVN/TJCLKP
```

Each element breaks in its own way.

Creator. People or an organisation. For government data, an organisation as creator is normal: "Rosstat (2026)" is better than the surname of the employee who exported the file. The mistake is mixing roles: putting the department that runs the catalog among the creators, or the publisher alongside people.

Year. The year the dataset or its version was published, not the year the user downloaded it, and not the year the record was created. If the software takes the year from the record creation date, after datasets are moved to a new portal all of them will get the year of the move.

Title. The dataset title, not the file name and not the portal name.

Version. Without it the citation points to the dataset in general, not to what was used. In the same DataCite sample, the version is filled in for only 191 of 1,333 DOIs, or 14%. But even a filled-in version does not always reach the citation: the BibTeX that DataCite serves for the Harvard Dataverse dataset contains no version, although the dataset has five versions, while the APA citation from the same DataCite does contain it.

Publisher. The organisation that published the dataset: a repository, an agency, an institute.

Resource type. The "[Dataset]" label distinguishes a data citation from a paper citation. Classic BibTeX has no type for data, and everyone improvises: DataCite serves `@misc`, Dataverse the non-standard `@data`, InvenioRDM `@dataset`, which biblatex, the modern replacement for BibTeX, understands.

Identifier. A DOI in the form of an address `https://doi.org/…`. If there is no DOI, use the permanent address of the dataset page and the access date. The Russian standard GOST R 7.0.108-2022 on citing online documents requires exactly that: the address and the access date are mandatory, and DOIs are not mentioned in the standard at all. Databases are named in it as one kind of electronic document.

If the description fields are empty, a citation generator will not help. Which fields are needed and how to check that they are filled in for real is covered in the chapter on metadata completeness; how DOIs and versions work is in the chapter on identifiers.

Citing changing data

A research repository fixes a version, and a citation of it stays precise. An open data portal and an indicator catalog more often maintain one dataset whose files are replaced every week. The citation "Ministry of Transport (2026). Suburban train timetable" points to different data a month later. There are five ways to deal with this, from simple to complex.

1. **An access date and an archive of dumps.** The citation contains a date, and the portal keeps past files, so the state on that date can be restored. Without an archive, the access date only honestly admits that the citation cannot be checked.

2. **A snapshot of the page.** Our World in Data suggests citing not the live chart page but its archived snapshot with a date in the address (`archive.ourworldindata.org/20260910-…`). It also asks you to cite both the original source and itself as the processor: "Gapminder (2015); UN IGME (2025) – processed by Our World in Data". A two-level citation is needed by any aggregator and any portal that republishes other people's data. A program gets the source string by an address convention: append `.metadata.json` to the chart address, and the response has a ready field for the citation, while `.csv` serves the data itself.
3. **Versions with their own identifier.** Each dump of a dataset becomes a version. Zenodo and Figshare do this, and a portal can too if there are not many releases. CELLxGENE Census adds a time commitment. Long-term support releases are promised every six months, each is kept for at least five years, and there is a citation generator for any slice. A retention period named in advance is what the promise "versions are kept" usually lacks.
4. **Citing a query.** GBIF gives its own DOI to every occurrence download: it records the selection criteria, the list of source datasets and the date. On 15 October 2026 the iNaturalist observation dataset alone was part of 2,623,241 such downloads. If the user later heavily filtered a download or collected data through the API, they register a "derived dataset" and also get a DOI. The general scheme is described in the Research Data Alliance recommendations on citing dynamic data: data is stored with versions and timestamps, the query itself is saved with a timestamp and given an identifier, through which the query can be re-run on the same state of the data. A new edition of the recommendations was endorsed in July 2025: it has 12 items instead of the 14 in the 2015 edition, and the substance has not changed.
5. **Citing part of a dataset.** Dataverse shows a separate citation for each file of a dataset, GBIF for an individual occurrence. This is needed when a conclusion rests on one file out of a hundred.

For an open data portal, a reasonable minimum is the first method: an access date in the suggested string and an archive of past dumps. How to keep past files is covered in the chapter on data storage.

Implementation options

A string template on the page. The cheapest method: a string assembled from fields is inserted into the dataset page template. The ckanext-doi extension for CKAN works this way: the "Cite this as" block is assembled from creator, year, title and publisher, and a DOI link is added. The cost is one theme edit. The risk is one style for everyone and mistakes in field choice: ckanext-doi itself takes the year from the CKAN record creation date.

A generator with CSL styles. The platform stores CSL styles and on request formats a citation in the required style and format. There are many styles: in October 2026 the CSL repository had 2,863 independent styles and another 8,002 "dependent" ones, for journals that use someone else's style under their own name. Dataverse, InvenioRDM and Zenodo, GeoNetwork, and the ckanext-citeproc extension for CKAN work this way. The cost is maintaining the set of styles and the mapping of fields to CSL variables.

A citation from DataCite. The catalog generates nothing itself but takes the citation from DataCite by DOI. NADA does this, and the BibTeX button in ckanext-doi leads straight to the DataCite API. The advantage is that the citation matches the one any other service will get. The drawbacks: it works only for datasets with a DOI, DataCite knows nothing about versions not recorded in its metadata, and its generator has traps:

- DataCite silently replaces an unknown style name with APA. A request for the style `gost-r-7-0-100-2018` returns an ordinary APA citation without any warning. InvenioRDM honestly responds to the same request with an error;
- the CSL repository has three GOST styles, all following the old GOST R 7.0.5-2008, and none outputs a DOI for a dataset. For the Harvard Dataverse dataset they give "Durbin P. Open Source at Harvard // 2017." and "Durbin P. Open Source at Harvard. Harvard Dataverse, 2017." If a catalog needs a GOST citation, it is better to write the data template yourself.

Meta tags for reference managers. `citation_*` and Dublin Core meta tags in the page head cost almost nothing and give one-click saving to Zotero and a correct record in Google Scholar. The DSpace core has them even where there is no citation block at all. Zotero also has separate translator programs for large platforms: Dataverse, InvenioRDM, Figshare, Dryad, DSpace. But a translator is tied to a list of addresses: the InvenioRDM translator knows 14 sites, from zenodo.org to the University of Tübingen repository. It will not recognise a new installation, and Zotero will be left reading the meta tags. Meta tags have a limit: the Google Scholar vocabulary has neither a version nor a "dataset" type, so Zotero often saves a dataset from them as a web page or an article. The old COinS format can convey the type, but almost nobody uses it. Zotero itself has had a "Dataset" type since version 6.0.24.

A file in the repository. For data on GitHub, the role of the button is played by a `CITATION.cff` file at the root of the repository. From it GitHub shows a "Cite this repository" link in two formats, APA and BibTeX. For the citation to be of data and not software, the file needs the line `type: dataset`. Without it GitHub will format the dataset as software: "[Computer software]" in APA and `@software` in BibTeX.

Option	What the user gets	Cost	Main risk
String template	one string in one style	a theme edit	wrongly chosen fields, especially the year
CSL generator	a string in any style and files for managers	a platform or extension, a set of styles	poor mapping of fields to CSL variables
Citation from DataCite	a string and files matching other services	nothing, if there is already a DOI	no DOI, no citation; silent style substitution
Meta tags	one-click saving to Zotero	a few lines in the template	no version and no "dataset" type
`CITATION.cff`	a button on GitHub, metadata for Zenodo	one file	without `type: dataset` you get a software citation

The options are not mutually exclusive. A good research repository does all of them at once: a string, files, meta tags and a DOI through which DataCite serves the same citation.

When it is needed and when not

Situation	What is needed	What usually covers it
A research repository	a string with version and DOI, BibTeX and RIS, meta tags, a citation for a dataset's file	the core of Dataverse, InvenioRDM, Figshare
An open data portal at a university or ministry whose data researchers use	a string with a permanent address and access date, an archive of past dumps	a template edit; for datasets cited in papers, a copy in a research repository with a DOI
An indicator and statistics catalog	a citation of a series or query with a date, not of a table page	a citation of a saved query, as in GBIF; a snapshot, as in Our World in Data
A geoportal with datasets cited by scientists	a string from ISO 19115 fields and a DOI	the GeoNetwork 4.4 core
A machine learning hub	a DOI for a dataset revision and a string in the dataset card	a DOI on request on Hugging Face; a citation from DataCite
Publishing in a Git repository	`CITATION.cff` with `type: dataset`, a release DOI	GitHub and Zenodo
An internal showcase, a layer of an internal map	a permanent address	no citation generator needed

It is mandatory for a research repository: citation is part of its job, and journals and funders require data citations. It is desirable wherever data is used in research and analysis. It is not needed if the data is not cited: an operational showcase, an internal map layer. Even there a permanent address is useful, but a BibTeX generator is not.

In brief, this is what a justification looks like using the decision record template:

- **Context.** A university open data portal on CKAN 2.11. Researchers ask for a "cite button". The portal has no DOIs and no agreement with DataCite.
- **Options.** A string in the dataset page template; the ckanext-citeproc extension with CSL styles; ckanext-doi with an agreement and a DataCite prefix; send datasets cited in papers to Dataverse or Zenodo.
- **Criteria.** Citation precision (version, permanent address), the cost of CKAN upgrades, whether a DOI is needed.
- **Decision.** A string in the template with a permanent address and access date, and an archive of past dumps. Datasets behind papers go to a research repository with a DOI.
- **Why.** Ten CSL styles do not make a citation without a version and DOI more precise. A template string does not depend on someone else's extension, while ckanext-doi requires an agreement the portal was not created for.
- **When to revisit.** If the university gets a DOI prefix through a library consortium, or if ckanext-citeproc releases a stable version tested on CKAN 2.12.

How it is done in platforms

The marks are core, extension, no, not checked, as in the summary table. Checked against documentation, source code and live installations on 12 October 2026.

Platform	String on the page	Files and styles	Hints for programs	What is important to know
Dataverse 6.x	core: with version and a content checksum (UNF)	core: EndNote XML, RIS, BibTeX, CSL-JSON for a dataset and for an individual file; through the API, for any version	core: Dublin Core meta tags, JSON-LD, the `cite-as` header	one DOI per dataset; the version is in the citation string
InvenioRDM v14, Zenodo	core	core: six styles in the menu (APA, Harvard, MLA, Vancouver, Chicago, IEEE), any CSL style through the API; BibTeX, CSL-JSON	core: `citation_*` meta tags, the `cite-as` header	each version has its own DOI plus a shared parent DOI. No RIS or EndNote in the API
Figshare	core: a "Cite" button	core: RefWorks, BibTeX, EndNote, DataCite, NLM, Dublin Core, RIS	not checked: the page answers robots with a check	each version has a DOI with the suffix `.v1`, `.v2`
DSpace 10.1	no	no	core: `citation_*` meta tags for Google Scholar	the "Cite" button is being ported from DSpace-CRIS; the porting pull requests were open in October 2026
GeoNetwork 4.4	core: a block in the record, enabled in the interface settings	core: HTML, text, RIS, BibTeX through the API	not checked	the composition of the string is set in a core XSL file; author and publisher roles are configured by parameters
NADA	core, through DataCite, if the study has a DOI	core, through DataCite	not checked	no DOI, no citation
CKAN 2.11–2.12	extension: ckanext-doi (only with a DOI), ckanext-citeproc	extension: BibTeX through DataCite (ckanext-doi), CSL styles (ckanext-citeproc)	no: there are no meta tags in the core; schema.org markup comes from ckanext-dcat	the extensions are young or tied to DOIs; see below
Huwise (OpenDataSoft)	no	no	no	there are text fields for the source and a `bibliographicCitation` field in the DCAT template, filled in by hand
Socrata, ArcGIS Hub	no	no	no	no citation generator
Hugging Face	a string in the dataset card, if the author wrote one	a citation from DataCite, if a DOI was minted	not checked	DOIs are minted on request for the current revision; for a new revision a new one is minted and the old one is marked outdated
GitHub	a "Cite this repository" button from `CITATION.cff`	APA and BibTeX	—	not a catalog platform; needs `type: dataset`

Dataverse explicitly lists citation formats as a standard feature, and its citation is the most precise: it has the version number and a content checksum with which you can check that the table has not changed. InvenioRDM's citation is simpler, but the DOI itself fixes the version: each version has its own.

Extensions and their risks

CKAN has no citation in its core. There are three candidates, each with its own caveat. The method for evaluating an extension is in the chapter on extension ecosystems.

- **ckanext-doi** from the Natural History Museum in London. It issues DOIs through DataCite and shows a "Cite this as" block with a BibTeX button served by DataCite. Without a DOI there is no citation. The documentation names CKAN 2.9 and 2.10; the extension has not been

tested on 2.11 and 2.12; more in the chapter on CKAN.

- **ckanext-citeproc** was published by the open.canada.ca portal team in March 2025 under the MIT licence. It formats a citation for a dataset and a resource using CSL styles that the administrator places in a folder on the server: the library ships with only one style, Harvard. Compatibility is declared with CKAN 2.10 and 2.11, and automated tests run on CKAN's main branch. On open.canada.ca itself (CKAN 2.10.8) the extension was not enabled in October 2026.
- **ckanext-citation** from the TIB group in Hannover appeared in June 2026 under the AGPL-3.0 licence. It serves BibTeX, RIS, EndNote, APA, the ACM format and CSL-JSON. CKAN 2.9 and 2.10 are declared. The tests check only citation formatting; the extension is not automatically tested on a live CKAN. Under the same name there is an older extension from the Taiwanese repository depositar: it has not been updated since 2022 and knows only the title, author and address.

For DSpace 6 there were third-party add-ons: Atmire's CSL module and the University of Waikato's RIS export. They have not been ported to DSpace 7 and later. In GeoNetwork the composition of the string is edited in an XSL file inside a core schema: such an edit has to be repeated after every upgrade.

How it breaks: one dataset, three citations

A check on 12 October 2026: for each dataset, the citation the catalog itself produces was compared with the one DataCite serves for the same DOI.

Harvard Dataverse (Dataverse 6.10.1), the "Open Source at Harvard" dataset. The dataset has five versions: the first from July 2017, the fifth from June 2026.

- The string on the page: `Durbin, Philip, 2017, "Open Source at Harvard", https://doi.org/10.7910/DVN/TJCLKP, Harvard Dataverse, V5, UNF:6:…`, with the version and checksum.
- BibTeX from Dataverse itself for the first version: `version = {V1}`, type `@data`.
- BibTeX from DataCite: no version, type `@misc`.
- APA from DataCite: "(Version 5.0)".

A user who takes the BibTeX from DataCite will cite the dataset without a version, that is, in effect the latest one. A year later that will be different data.

The Ifremer Sextant catalog (GeoNetwork 4.4.13), a temperature and salinity climatology of the Mediterranean Sea.

- The string from the catalog: `IFREMER / IDM/SISMER, Simona Simoncelli, Paolo Oliveri . SeaDataCloud Mediterranean Sea - Temperature and Salinity Climatology V1. https://doi.org/10.12770/…`: no year, and the organisation is the first author.
- BibTeX from the catalog: `author = {IFREMER / IDM/SISMER, Simona Simoncelli, Paolo Oliveri}`, `publisher = {}`. BibTeX separates authors with the word `and`, not commas, so a reference manager will read three authors as one with a long name. The publisher is empty.

- APA from DataCite: `Simoncelli, S., & Oliveri, P. (2019). … (Version 1.1) [Dataset]. IFREMER / IDM/SISMER.`, with year, version and publisher.

The fields are in the description, but the catalog's generator takes them from the wrong roles. This is a typical failure: the feature is "in the core", but the result is worse than an external service's.

InvenioRDM (the demo instance) and the GOST style. BibTeX and APA are assembled correctly: creator, year, version, publisher, DOI. The `gost-r-7-0-5-2008` style gives "Gallagher J. и др. Jones and Sons's gallery // 1987–2018.", without a publisher or DOI. The problem is not the platform but the CSL style: it was not written for datasets.

Four more failures are common:

- **No version.** The citation is correct in form and useless for verification. In the DataCite sample, the version is filled in for 14% of datasets.
- **The year from the record date.** The software substitutes the year the record was created, not the year the data was published. After a move to a new platform, all datasets are "published" in the year of the move.
- **Silent style substitution.** For an unknown style DataCite serves APA, and the user is sure they got what they asked for.
- **A software citation instead of data.** `CITATION.cff` without `type: dataset` on GitHub gives "[Computer software]".

Publishing without a portal

Without a portal the author assembles the citation entirely, but then nobody but them can mix up the roles. Three things are enough.

`CITATION.cff` at the root of the repository. The format version is 1.2.0 and has not changed since 2021. `cff-version`, `message`, `title` and `authors` are mandatory; for data add `type: dataset`, the version, the release date and the DOI once it appears. GitHub will show a "Cite this repository" button based on the file. Zenodo, when archiving a release, will take the authors and title from it, but only if the repository has no `.zenodo.json` file: if there is one, Zenodo ignores `CITATION.cff`. The Zenodo record type (dataset, not software) is set by the `upload_type` field in `.zenodo.json`. How to combine both files is covered in the chapter on Git repositories. The file can be checked with `cffconvert --validate`. A ready CITATION.cff template is in the book's templates.

A "How to cite" line in the README and on the dataset page. One line in the style accepted in your field, with the version and a permanent address, and after archiving in Zenodo, with the DOI. Zenodo offers two citations to choose from: the DOI of a specific release and the concept DOI for all versions, which always leads to the latest. A paper needs the first.

Markup on the dataset page. schema.org markup with creator (`creator`), publication date (`datePublished`), version (`version`) and publisher (`publisher`), which Google and aggregators read. The `identifier` field is added once the DOI has been issued: the template does not include it, so as not to publish an empty number. A few more meta tags are useful for Zotero:

```
<meta name="citation_title" content="Dataset title">
```

```
<meta name="citation_author" content="Surname, Given name">
<meta name="citation_date" content="2026/01/15">
<meta name="citation_publisher" content="Publishing organisation">
```

Once a DOI is issued, `citation_doi` with the real number is added to these fields. A version cannot be passed in meta tags, so it must be in the title or in the "How to cite" line. How to set up the repository and releases is covered in the chapter on Git repositories.

How to check a live portal

A manual check is five questions about five random datasets:

1. The dataset page has a "How to cite" block with creator, year, title, publisher and identifier.
2. The year in the citation is the year the dataset or version was published, not the year the record was created and not the current year.
3. For two versions of one dataset, the citations differ in version number or identifier. If there are no versions, the citation has an access date.
4. BibTeX or RIS can be downloaded, with the same identifier and version as on the page, and authors separated properly (in BibTeX, by the word `and`).
5. If the dataset has a DOI, the DataCite citation matches the one on the page in creator, year and version.

Commands

A citation by DOI from DataCite, in a style, in BibTeX and in RIS. It works for any DOI minted through DataCite, even if the catalog has no button at all:

```
DOI=10.7910/DVN/TJCLKP
curl -sLH "Accept: text/x-bibliography; style=apa" "https://doi.org/$DOI"; echo
curl -sLH "Accept: application/x-bibtex" "https://doi.org/$DOI"; echo
curl -sLH "Accept: application/x-research-info-systems" "https://doi.org/$DOI"
```

If the style name is misspelled, a response still comes, in APA style. Check that you got what you asked for. The list of style names is in the CSL repository.

Dataverse: a citation for a specific version in BibTeX, RIS, EndNote or CSL-JSON:

```
SERVER=https://dataverse.harvard.edu
PID=doi:10.7910/DVN/TJCLKP
curl -s "$SERVER/api/datasets/:persistentId/versions/1.0/citation/BibTeX?persistentId=$PID"
curl -s "$SERVER/api/datasets/:persistentId/versions/:latest-published/citation?
persistentId=$PID"
```

InvenioRDM: a string in a chosen style and BibTeX. The example is CaltechDATA; Zenodo itself did not answer programs without a browser in October 2026:

```
SERVER=https://data.caltech.edu
ID=$(curl -s "$SERVER/api/records?size=1&q=metadata.resource_type.id:dataset" | jq -r
'.hits.hits[0].id')
curl -s -H "Accept: text/x-bibliography" "$SERVER/api/records/$ID?style=apa&locale=en-US";
echo
```

```
curl -s -H "Accept: application/x-bibtex" "$SERVER/api/records/$ID"
```

GeoNetwork 4.4: the list of formats and a RIS citation for a record:

```
API=https://sextant.ifremer.fr/geonetwork/srv/api
UUID=ad07a55f-5de7-4abc-ba89-8899b16c4b59
curl -s "$API/records/$UUID/formatters/citation?format=?"; echo
curl -s "$API/records/$UUID/formatters/citation?format=ris"
```

Hints for programs on the page, meta tags and the `cite-as` header:

```
PAGE=https://dspace.mit.edu/handle/1721.1/93094
curl -sL "$PAGE" | grep -o -E '<meta name="(citation_|DC\.)[a-zA-Z_.]*"' | sort | uniq -c
curl -sI https://data.caltech.edu/records/$ID | grep -i '^link' | tr ',' '\n' | grep cite-
as
```

Empty output from the first command means Zotero will save the dataset as an ordinary web page. Empty output from the second means a program will not know which address to treat as the address for citation.

What follows from this

- Fields first, then the button: creator, publication year, title, version, publisher, identifier. Without them any generator produces a dud.
- If datasets have DOIs, compare your citation with DataCite's. A mismatch in creator, year or version is an error to fix in the description, not in the template.
- For changing data, add an access date to the citation and keep past dumps. For scholarly citations, fix versions or give a download an identifier, as GBIF does.
- `citation_*` and Dublin Core meta tags cost a few template lines and save the user manual entry.
- CSL styles will not assemble a GOST citation for a dataset correctly: write the template yourself.
- A Git repository with data needs `CITATION.cff` with the line `type: dataset`.

Sources

- Joint Declaration of Data Citation Principles, FORCE11, 2014: eight principles, the distinction between legal and scholarly attribution
- DataCite: content negotiation and formats; the recommended order of citation elements per the CESSDA guide; a sample of 3,000 random DOIs with the type "dataset" through the DataCite API, 12 October 2026
- Make Data Count: Data Citation Corpus documentation, release 4.0 data, 27 July 2025
- Research Data Alliance: Precise, Actionable Reference to Dynamic Data, 2025; the 2015 edition
- GBIF citation guidelines: download DOIs and derived datasets; Our World in Data: how to cite
- Dataset markup in Google's documentation: the `citation` property is not for citing the dataset itself

- Dataverse: features, citation API, data citation
- InvenioRDM: record serializers, default styles
- Figshare: citing and exporting citations
- DSpace: meta tags for Google Scholar; porting citation export from DSpace-CRIS: issue 11783, pull requests 11451 and dspace-angular 4779, open as of 12 October 2026
- GeoNetwork 4.4: citing a record
- CKAN: ckanext-doi, ckanext-citeproc, ckanext-citation (TIB), ckanext-citation (depositar)
- Huwise: compatibility fields, Hugging Face: DOI
- GitHub: CITATION files; Citation File Format; Zenodo: CITATION.cff and .zenodo.json
- Zotero: exposing metadata, the Dataset type, translators
- The CSL styles repository, count as of 12 October 2026; biblatex: the dataset type
- GOST R 7.0.108-2022 (in Russian): bibliographic references to electronic documents on networks
- Live checks on 12 October 2026: Harvard Dataverse, the Open Source at Harvard dataset, Ifremer Sextant, the Mediterranean climatology, the InvenioRDM demo instance, CaltechDATA, DSpace@MIT
- Identifiers and DOIs, Metadata Completeness, Git Repositories
- Our World in Data, Chart API: the `.csv`, `.metadata.json`, `.zip` suffixes; CZ CELLxGENE, Census: long-term support releases every six months, kept for at least five years; GBIF, downloads including the iNaturalist dataset: 2,623,241 as of 15 October 2026

Access, Roles and Workflows

A catalog with one publisher lives without roles: there is a person who uploads files. An organisation's catalog becomes shared almost immediately. Then you need to agree who creates a dataset, who reviews it, who sees the file and who can create an account at all.

The practical conclusion of the chapter is this. Most government portals need two roles (administrator and organisation editor) and a draft that is not visible in search. Review before publication is needed when the publishers are people you do not trust to publish data straight away: external authors of a research repository, hundreds of municipalities, users of a hub. Restricted access is needed for personal, licensed and sensitive data, and the dataset description stays open even then. Open registration is needed only where outside people publish, and then with an invitation or a check, not for everyone. Reading descriptions and downloading open files should not require an account at all.

At the end of the chapter there is a measurement of 1,125 CKAN portals on 12 October 2026: how many have open registration, how many of them are already filled with spam accounts, and which login and access extensions are actually installed.

What it is

The words "access and roles" hide five different questions. The re3data registry of research repositories separates at least three of them: who can enter the repository, who can download data and who can upload it. The answers do not have to match: in Zenodo any registered user can upload, while a restricted file can be downloaded only with the author's permission.

Who publishes. The publishing organisation, an editor, a curator, an administrator. A **role** is a set of permissions given to a person within an organisation or collection: "editor of organisation A" can edit A's datasets and cannot touch B's. Permissions are given to roles, not to individuals, so that a change of staff does not mean going through a hundred checkboxes.

What happens before publication. A **draft** is a dataset that has been created but is not visible in search or in the dump. **Moderation**, or review before publication, is a step at which another person (a curator, a moderator, an organisation administrator) looks at the dataset and releases it or returns it to the author. An **activity log** records who changed what in the description and when.

Who sees the data. An open file, a file after accepting terms, after registration, after an approved request, after an embargo date, only in a secure environment, or not at all, description only. You can restrict a whole record, only the files, or an individual file, and in cloud services even individual columns and rows of a table.

How people log in. Two terms here are often confused. **Authentication** is checking who you are: a password, a second factor, a government account. **Authorisation** is checking what you are allowed to do: a role, permissions on a dataset, an approved request. **Single sign-on** (SSO) moves authentication from the catalog to an external organisational system: an employee logs in with the same account as for email. Two standards are used for this. **SAML 2.0** is a 2005 OASIS standard

on which university federations and enterprise user directories run. **OpenID Connect** is an identity layer on top of the OAuth 2.0 protocol; it is supported by Keycloak, cloud identity providers and many government login systems.

Who can create an account. Registration can be closed (accounts are created by an administrator or by email invitation), open with a check (an administrator approves the application, or only addresses from certain domains are allowed), open with a CAPTCHA, or open to all.

Why it is needed

So that agencies do not edit each other's data, and it is clear who is responsible. When twenty organisations publish on a portal, each must see and change only its own datasets. A role also answers the user's question "who is responsible for this": a dataset shows an organisation and a contact, not just the site name. The publisher and organisation in the description are covered in the chapter "Anatomy of a Catalog".

So that an error is caught before publication. A published dataset is picked up within hours by harvesters, search engines and web archives. A file with non-anonymised personal data published by mistake is then almost impossible to remove from all copies. A draft and a second look before publication are cheaper than dealing with a leak.

To show data that cannot be opened. Survey microdata, medical records and licensed databases cannot be made available to everyone, but people need to know they exist. In the EU this is already an obligation. Under the Data Governance Act (Regulation 2022/868, applicable since 24 September 2023), each country maintains a single information point with a list of protected public-sector data: format, size and conditions for reuse. The data itself is provided on request, including only in a secure processing environment. The countries' lists are collected by the European Register for Protected Data (ERPD) on data.europa.eu. The European Health Data Space Regulation (2025/327) requires health data access bodies to maintain an open machine-readable catalog of datasets with access conditions; this part applies from 26 March 2029.

So that robots do not take over the portal. Open registration on a public site is an invitation to spam. Robots create accounts and, in their name, create organisations, groups, datasets and data requests advertising casinos and fraudulent links. This is why registration on DataHub.io had to be closed in 2013; in 2021 demo.ckan.org was clogged with spam; and in 2025 the portal of Tunisia's meteorological service had more than 1.3 million accounts and more than 45 thousand spam texts in its data requests section. The CKAN developers therefore turned open registration off by default: in 2.9.8, and in the 2.10 line only from 2.10.4. In 2.10.0–2.10.3 it is open by default again. The measurement at the end of the chapter shows that the problem is not historical.

To know the publisher. On government portals publishers often log in through the government identity system, and this is not a formality: this way the portal knows on behalf of which agency a dataset was published. On data.gov.ru, under the 2024 guidelines, a provider (a public authority or local government) logs in through ESIA, the Russian government identity system; an administrator verifies its profile, only a verified provider can publish, and datasets go through moderation. On Norway's data.norge.no, descriptions are registered after logging in through the

national ID-porten system and only with a role in the Altinn authorisation register. On data.gouv.fr, organisation staff log in through ProConnect, the government login for professionals built on OpenID Connect, alongside ordinary email registration.

The flip side is cost. Every role, review step and access type is a setting that has to be maintained, explained to new staff and checked when the platform is upgraded. Moderation that nobody has time for turns into a queue of a hundred datasets that do not come out for months.

Implementation options

Who publishes: five permission models

No roles. One person or one team with shared permissions. Static sites, Git repositories and small portals live this way. This is enough while one team publishes.

Organisations with roles. A dataset belongs to an organisation, and people in the organisation have roles. CKAN has three: a member sees the organisation's private datasets; an editor creates, edits and deletes datasets and switches them between public and private; an administrator also manages members. Above all of them is the sysadmin. udata, on which data.gouv.fr runs, has administrator and editor roles, and since version 15.4.0 (March 2026) also a partial editor. In Socrata, roles are given for the whole portal: administrator, publisher, editor, designer, viewer. This is the most common model on open data portals.

Collections with inherited permissions. Permissions are given on a collection and pass to everything inside. Dataverse (collections and the datasets in them), DSpace (communities, collections, items and files) and GeoNetwork work this way; in GeoNetwork each group has its own set of permissions on a record: view, download, show on the interactive map, edit, notifications. The model suits a university: a faculty is a collection, a department is a collection inside it.

Communities with curators. The author publishes on their own, and a community decides whether to include the record in its collection. InvenioRDM and Zenodo work this way: a community has an owner, managers, curators and readers, and a record enters the community after a curator's review.

Permissions on an individual dataset. Permissions are given not by organisation but per dataset: in Huwise (formerly OpenDataSoft), "view", "view and edit", "view, edit and publish" for a user or group; in ArcGIS Online, the access levels "owner only", groups, organisation and everyone. In CKAN a similar role is played by dataset collaborators: since version 2.9 an organisation administrator can give an individual member or editor permissions on one dataset. By default this feature is turned off.

The finer the permissions, the more expensive the maintenance. A matrix of twenty roles almost never pays off: a year later nobody remembers why it is set up that way, and permission errors are found through complaints.

What happens before publication

Immediate publication. Save means publish. CKAN and DKAN 2.x work this way by default: a new dataset in DKAN immediately gets the "published" state unless the administrator has changed the setting. Fine if your own staff publish and an error can be fixed quickly.

A draft. The dataset is saved but not visible in search. In CKAN the multi-step creation form saves a dataset as a draft, and 2.12 added "Publish" and "Unpublish" buttons. In udata a draft is a private dataset; when working through the API, the private flag must be passed with every request, otherwise the dataset becomes public. In Huwise edits are saved separately and released by publishing again. A draft costs nothing and is needed by everyone except the simplest catalogs.

A separate right to publish. One person can prepare a dataset, another releases it. In Socrata an editor uploads data, but only an administrator, a publisher or the dataset owner publishes. In Huwise the rights "create datasets" and "edit all datasets" do not include the right to publish. This is not yet a review, but it is already a second person before a dataset goes out.

Curator review. The author submits a dataset for review, and a curator releases it or returns it with a comment. In Dataverse a contributor submits a dataset for review, and the collection curator publishes it or returns it to the author. In DSpace a collection can have a multi-step workflow: review, edit, final approval. In GeoNetwork 4.4 the workflow "draft → submitted → approved → retired" is enabled for the whole catalog or for individual groups, and editing an approved record creates a working copy that outsiders do not see until it is approved again. A GeoNetwork subtlety: by default, even with the workflow enabled, a record can be published without approval (the `allowPublishNonApprovedMd` parameter), and this has to be switched off separately. In Socrata, approval before publication is enabled in settings, automatic or manual. CKAN has no such step in the core, only extensions, covered below.

An activity log. Who changed what and when. In CKAN since version 2.10 the log has been moved to a separate `activity` module, which ships with CKAN but must be enabled. It stores a snapshot of a dataset on every edit and shows the difference between versions; by default outsiders do not see the details. In research repositories versions play the role of the log: editing a published dataset in Dataverse creates a new version, and the previous one remains. Versions are covered in the chapter "Persistent Identifiers, DOIs and Versions".

Part of the review is better left to a program rather than a person: mandatory fields, a licence from a list, the file opens, the encoding is right. Then the curator looks at the content, not at typos. See the chapter "Metadata Completeness and Depth".

Who sees the data: access levels

Level	What the user does	Where it is found
Open	downloads without conditions	any catalog
Open after accepting terms	ticks "I agree" or fills in a short form	"direct access" in NADA, the guestbook in Dataverse, gated datasets with automatic approval on Hugging Face
After registration	logs in	public use files in NADA
On request	submits a request and waits for the owner's decision	Dataverse, InvenioRDM and Zenodo, DSpace (request a copy), Huwise, Hugging Face with manual approval, "licensed data" in NADA

Level	What the user does	Where it is found
Embargo	waits for the date after which the file opens	Dataverse, InvenioRDM and Zenodo, DSpace
Secure environment	works with data on someone else's server or in a special room, takes away only checked results	"data enclave" in NADA, trusted research environments, a Data Governance Act condition
Description only	learns that the data exists and whom to ask	"data not available" in NADA, the EU register of protected data

How these levels work in microdata catalogs is covered in detail in the chapter on microdata catalogs.

Two clarifications. Registration is not yet restricted access. The EU Publications Office access-right vocabulary considers a dataset behind registration or an API key public, too, if anyone can get them. For open data in this book's sense, registration is already a barrier, but that is a separate discussion: see the glossary entry "Publicly available and open data". And "on request" without a response deadline and without conditions written in advance is not an access level but a refusal stretched out in time.

Description open, file closed

Closing the description together with the file is a bad trade. A hidden dataset cannot be found or cited, and there is no one to send an access request to. The FAIR principles explicitly separate these things: a dataset with restricted files can be findable and accessible through a clear procedure if its description is open and it has a persistent identifier.

Platforms behave very differently here, and this is worth checking before choosing.

- **Can do it in the core:** Dataverse, InvenioRDM and Zenodo, DSpace, NADA: the description is open, the files are restricted, and the user sees how to get them. In GeoNetwork, download is a separate permission that can be withheld from guests; it protects files uploaded into GeoNetwork itself, but not external links. Huwise shows the dataset in the catalog to everyone, and the contents only to selected users and groups, with a "Request access" button; the button works only if marketplace mode is enabled on the portal.
- **Cannot:** in CKAN a private dataset is hidden entirely, together with its description. Only the ckanext-restricted extension lets you open the description and restrict an individual file. In Socrata a private dataset does not appear in the public catalog at all; you can only hide individual description fields of a public dataset.

For a restricted dataset to be found not only on the site, the access conditions must get into the metadata dump.

- **DCAT.** The `dct:accessRights` property with a value from the EU Publications Office vocabulary: PUBLIC, RESTRICTED, NON_PUBLIC (the vocabulary also has the refinements SENSITIVE and CONFIDENTIAL within NON_PUBLIC). In DCAT-AP 3.0.1 the property is optional, but the European Register for Protected Data requires it for every dataset and forbids the value "public", and in HealthDCAT-AP it is mandatory together with the health data access body.

- **DataCite and OpenAIRE.** A rights field with a value from the COAR vocabulary: open access, embargoed, restricted access, metadata only. This is how OpenAIRE and research data harvesters see the access mode.
- **schema.org.** The `isAccessibleForFree` property and the free-text `conditionsOfAccess`. Google Dataset Search reads the first and does not require the data to be open: it needs an open page with a description, without a login and without a robots block.

Export formats are covered in the chapter "Metadata Export", and the licence alongside access conditions in the chapter "Licences and Access Conditions".

How common restricted access is at all is shown by two large repositories. On 12 October 2026, of 726,858 datasets in Zenodo, 680,981 (94%) were open, 41,715 (5.7%) had restricted files, and 4,162 (0.6%) were under embargo. In Harvard Dataverse, of 5.55 million files, 76,579 (1.4%) were restricted and another 1,513 were under embargo. Restricted files are the exception even where they are easy to set up.

How people log in

A password on the portal. Works everywhere and is fine for a small editorial team. The weak spot is passwords guessed by brute force. A second factor protects: a one-time code from an app. In CKAN it comes from the ckanext-security extension; in Huwise and ArcGIS Online it can be made mandatory in the settings. Thailand put two-factor login into its standard portal build: in the measurement, the `thaigdc2fa` extension is installed on 160 of 250 Thai CKAN servers.

Single sign-on through the organisation's system. The publisher logs in with an agency or university account via SAML or OpenID Connect. Needed when there are many publishers who come and go, and nobody is available to create and disable separate passwords for them. A departing employee loses access to the portal along with email. Dataverse, InvenioRDM, DSpace, GeoNetwork, Huwise, Socrata and ArcGIS Online have single sign-on in the core or in the service settings; CKAN only through extensions.

A government login system. ESIA in Russia, ID-porten in Norway, ProConnect in France. The portal gets a verified identity and organisation for the publisher but becomes tied to someone else's system: if it fails, no publisher can log in.

A research account. ORCID is a researcher identifier that research repositories also accept as a login method. Convenient for a repository to which authors come from different organisations.

A key for programs. Publishing through the API uses not a password but a key (token) issued for a program and revoked separately. CKAN has had tokens since version 2.9, and the shipped `expire_api_token` module limits their lifetime.

A login is needed by the publisher, not the reader. If a portal requires an account to view descriptions or download open files, this is no longer a security question but a door closed to programs.

Who can create an account

Option	How it works	When it fits
Closed registration	accounts are created by an administrator or by an organisation administrator's email invitation	only staff of known organisations publish; the usual case for a government portal
Open with a check	an administrator approves the application, or registration is allowed only from the right domains	there are outside publishers, but not many
Open with a CAPTCHA	anyone can register; the CAPTCHA filters out simple robots	a research repository, a hub, a site with discussions, together with checks on what is published
Open to all	anyone registers and can publish immediately	almost never for a public catalog

A spammer does not need to be able to publish datasets. Any place where text and a link appear on a public page is enough: a user profile, a new organisation or group, a data request, a comment. So what matters is not only registration but also what a new account can do. In CKAN any logged-in user can by default create groups and organisations (the `ckan.auth.user_create_groups` and `ckan.auth.user_create_organizations` parameters), and in the data requests extension create requests and comments without review. In ArcGIS Hub Premium, community accounts created through open registration get the publisher role by default. In Socrata, by contrast, a self-registered account gets no role on the portal.

A CAPTCHA helps but does not solve the problem. CKAN supports Google reCAPTCHA v2: at registration, and since version 2.11 also at login and password reset. Automatic content checking works better: udata flags suspicious datasets, organisations, profiles, reuses and discussions by a word list and sends them to a moderator.

How common it is

According to registry export 1.27.0, restricted access to at least part of the data is recorded for 1,131 of 43,614 catalogs, or 2.6%. 537 of them have no open data at all; in 594 open and restricted data live side by side. The queries are in the appendix.

More interesting is which platforms these catalogs run on.

Platform	Catalogs with restricted access	Total in the registry, all types	Share
DHIS2 (a health information system)	72	74	97%
Gen3 (a biomedical data platform)	29	30	97%
Redivis (a research data platform)	12	13	92%
NADA	114	159	72%
Pure (a research information system)	133	369	36%
InvenioRDM	4	76	5%
Dataverse	5	257	1.9%
DSpace	10	830	1.2%
CKAN	12	1,951	0.6%
GeoNetwork	1	764	0.1%

This counts all catalogs on a platform, whatever their type. The platform profiles show smaller numbers because they count only the main type: for example, CKAN has 1,951 catalogs in total and 1,906 open data portals, DSpace 830 and 829 research repositories, NADA 159 and 158 microdata catalogs.

Dataverse, DSpace and InvenioRDM can all restrict a file while keeping the description open in the core, but catalogs where restricted access is a noticeable part of the content are rare on them. Catalogs with restricted data run on platforms where restriction is the norm: health systems, biomedical platforms, microdata catalogs. The conclusion for choosing: if restricted access is the main task rather than a rare exception, look at platforms built for it, not at an open data portal with an extension. The mark in the registry refers to the catalog as a whole: Harvard Dataverse is open in the registry, although one and a half percent of its files are restricted.

When it is needed and when not

Situation	What is needed	What is not needed
One team publishes its own data	a draft, closed registration	roles and moderation
An agency and its subordinate bodies; a regional portal with municipalities	organisations with administrator and editor roles, a draft, an activity log, closed registration with invitations	multi-step moderation: each organisation is responsible for its own data
A national portal harvesting descriptions from other catalogs	permissions only for its own editorial team; publishers work in their own catalogs	accounts for hundreds of publishers on the national portal
A university repository where staff upload	faculty collections, library curator review, university single sign-on, embargo	registration open to all
A general research repository or hub for outside authors	open registration with a CAPTCHA, community curators, spam checks on content, restricted files on request	manual approval of every account
A microdata catalog	several levels of file access, requests with a decision log, a description always open	closing the description together with the file
Protected public-sector data under the Data Governance Act	an open description with `dct:accessRights`, conditions and request procedure, a response deadline	storing the data itself in the catalog
A portal with discussions, data requests, reuses	moderation of user texts, restricted permissions for new accounts	open registration without content checks

Roles are needed as soon as more than one team publishes. Moderation is needed if the circle of publishers is wider than the circle of those you trust to publish data straight away. Restricted access is needed for personal, licensed and sensitive data. It is not needed "to see who downloads": statistics are for that, not closing the file.

In brief, this is what a justification looks like using the decision record template:

- **Context.** A ministry portal on CKAN was upgraded from 2.8 to 2.11. Fourteen subordinate organisations publish, about 60 staff. Registration has been open since the early years; there are tens of thousands of accounts, and the user list and data requests section contain advertising. The ministry asks to "turn on moderation for all datasets".
- **Options.** Leave things as they are; close registration and invite publishers; install a review-before-publication extension; move login to the agency's single sign-on.

- **Criteria.** How many people publish and who is responsible for them; who will do moderation and how much time they have for it; compatibility of extensions with the core version; what the portal user will see.
- **Decision.** Close registration, close the user list (`ckan.auth.public_user_details = false`), delete spam accounts, forbid ordinary users from creating organisations and groups. Organisation administrators invite publishers. Roles are administrator and organisation editor; the draft and activity log are enabled. Do not introduce moderation of all datasets; moderate only the data requests section.
- **Why.** Staff of known organisations publish, and the organisation itself is responsible for the datasets. Moderating 14 organisations falls on one person in the ministry and will become a queue. The review-before-publication extension is tested only on 2.10 and 2.11, and when moving to 2.12 the portal would get an unnecessary obstacle.
- **When to revisit.** If outside organisations or citizens start publishing datasets, if errors in published datasets keep recurring, or if the number of publishers grows so much that nobody is available to create and disable them manually; then move to single sign-on.

How it is done in platforms

The marks are core, extension, no, not checked, as in the summary table. Checked against documentation and repositories on 12 October 2026.

Platform	Roles	Review before publication	Description open, file closed	Access requests, embargo	Single sign-on	Registration
CKAN 2.12	core: organisations, three roles, dataset collaborators	draft in the core; curator review is an extension	no: a private dataset is hidden entirely; the ckanext-restricted extension	extension (ckanext-restricted); no embargo	extension: SAML, OpenID Connect, LDAP	closed by default since 2.9.8 and 2.10.4; email invitation in the core
Dataverse	core: roles on collection and dataset, groups	core: submission for curator review	core	core: access requests, terms, guestbook, embargo	core: Shibboleth (SAML), OpenID Connect, ORCID	configurable
InvenioRDM, Zenodo	core: communities with owner, managers, curators	core: review on inclusion in a community	core: the whole record or only the files restricted	core: access requests, embargo	core: ORCID, OpenID Connect, SAML	open on Zenodo
DSpace 10.1	core: groups, permissions on community, collection, item and file	core: configurable multi-step workflow	core	core: embargo, Request a Copy	core: Shibboleth, LDAP, ORCID, OpenID Connect	configurable
NADA	core	not checked	core: seven types of file access	core: requests for licensed data	not checked	open: needed for requests

Platform	Roles	Review before publication	Description open, file closed	Access requests, embargo	Single sign-on	Registration
GeoNetwork 4.4	core: five roles in groups, group permissions on a record	core: draft, review, approval; by default approval can be bypassed	core: a separate permission to download uploaded files	no	core: LDAP, CAS, OpenID Connect, Keycloak, Shibboleth, EU Login	closed by default; can be enabled with a CAPTCHA and a domain list
udata (data.gouv.fr)	core: administrator, editor, partial editor	draft in the core; no curator review	not checked	not checked	ProConnect on OpenID Connect at data.gouv.fr	open; automatic spam flagging
DKAN 2.x	core: Drupal permissions	core: states "draft", "published", "hidden", "archived"; by default a dataset is published immediately	not checked	not checked	Drupal modules, not checked	not checked
Huwise (OpenDataSoft)	service core: dataset permissions for users and groups	a separate right to publish; no curator review	core: the dataset is visible in the catalog, the contents to selected users	core: access requests on portals with marketplace mode	core: SAML, OpenID Connect	closed, open, open with approval; CAPTCHA
Socrata (Tyler Data & Insights)	service core: five portal roles, three dataset roles	core: approval automatic or manual	no: a private dataset is not visible in the catalog	no	core: OpenID Connect, SAML through support	open, with no role on the portal
ArcGIS Online, Hub	service core: roles and item access levels	no	partly: a public item over a secured service requires a login	joining a group on request	core: SAML, OpenID Connect	open for the community on Hub Premium, publisher role by default
Hugging Face	core: organisation roles, resource groups on paid plans	discussions and pull requests	core: gated dataset	core: automatic or manual approval, a custom form	on Team and Enterprise plans: SAML, OpenID Connect	open
A static site, Git	repository permissions	a pull request with required review	no	no	GitHub or GitLab login	—

CKAN extensions and their risks

The CKAN core has no single sign-on, no review before publication, no per-file access and no spam protection beyond a CAPTCHA. All of this is covered by extensions, and this part of the ecosystem has the weakest maintenance. The method for evaluating an extension is in the chapter on extension ecosystems; here is what turned up during the check.

Task	Extension	Maintainer	Tests on CKAN versions	What is important to know
SAML login	ckanext-saml2auth	Keitaro	2.10, 2.11	last release December 2022; changes go to the main branch
SAML login	ckanext-saml	Link Digital (DataShades)	no automated tests	the changelog mentions 2.12 support (June 2026)
OpenID Connect login	ckanext-oidc-pkce	Link Digital	2.9, 2.10, 2.11	by its description built for Okta and not tested with other providers
LDAP login	ckanext-ldap	Natural History Museum (London)	test images not updated since September 2023	release 4.1.3 on 28 September 2026; 2.9 and 2.10 declared
Login protection: brute-force lockout, second factor, password rules	ckanext-security	data.govt.nz (New Zealand)	no automated tests	version 4.2.0 of 5 October 2026 is for 2.11; needs a separate Redis; the second factor is on by default for everyone
Open description, restricted file, access request	ckanext-restricted	EnviDat (Switzerland)	no working tests	compatibility declared up to 2.11; requires ckanext-scheming and two more extensions
Review before publication	ckanext-approvalworkflow	Keitaro	2.10, 2.11	release 2.0.0 in March 2026; enabled for the whole portal or for individual organisations; needs email
Configurable workflow with steps and approvals	ckanext-workflow	Link Digital	no automated tests	the new version was rewritten in summer 2026 and has not been released
Data requests	ckanext-datarequests, the Queensland Government branch	Queensland Government	2.10, 2.11, 2.12	the original CoNWeT Lab repository was abandoned in 2019; no CAPTCHA or moderation queue, but requests can be limited to organisation members
CAPTCHA on any form	ckanext-recaptcha	Link Digital	2.9	the only extension found against spam beyond the core CAPTCHA

Three practical conclusions follow from the table. A login extension is the riskiest at upgrade time: if it does not work on the new core version, no publisher can log in to the portal, so login is tested on a test instance before the move. Review before publication in CKAN is someone else's extension, and before requiring it, it is worth asking whether a draft and roles would do. CKAN handles a restricted file with an open description worse than research platforms: if this is the main task, choose a platform built for it.

Publishing without a portal

Without a portal there are fewer roles and reviews, but they exist.

- **A Git repository.** Changes to data and descriptions arrive as a pull request (GitHub) or merge request (GitLab), and review can be made mandatory. In GitHub, branch protection requires approval before merging, and a `CODEOWNERS` file assigns required reviewers for folders: for example, the statistics department reviews everything in `statistics/`. On GitHub's free plan

this works in public repositories. On GitLab's free plan approval is optional and does not block merging; mandatory approval rules and code owners are on paid plans. Git history also serves as the activity log. More in the chapter on Git repositories.

- **A private repository** hides both data and description. If the data is restricted but needs to be described, the description goes into a public repository or onto a site, and the data where access can be configured.
- **Zenodo and Hugging Face** give "description open, file closed" without a portal of your own. In Zenodo a record's files are restricted and access requests are accepted; in Hugging Face a dataset is made gated with automatic or manual approval and a custom form.
- **A static site** has no built-in "on request": it is for publishing open data. For restricted data the site can honestly describe the dataset, the procedure and the response time for a request, and give an address to write to. In schema.org markup you then set `isAccessibleForFree` and describe the conditions in `conditionsOfAccess`. More in the chapter on static sites.

How to check a live portal

A manual check takes fifteen minutes.

1. Open five datasets from different organisations. Each shows an organisation and a contact, not just the site name.
2. Open the portal without logging in. Descriptions can be read and open files downloaded. If the portal promises restricted data, its descriptions are visible too, while downloading is not, and next to it is written how to get access and how long the answer takes.
3. Find the metadata dump (DCAT, OAI-PMH) and see whether it includes access conditions: `dct:accessRights` in DCAT, the rights field in DataCite.
4. Open the registration page (`/user/register` on CKAN). It is either closed or leads to an invitation or a check.
5. Open the user list, the data requests section, comments. There is no advertising or fraudulent links there.
6. If you have a test editor account, check that only your organisation's datasets are visible and that a new dataset does not appear in search until it is published.

Commands

For a CKAN portal. The core version and extensions related to login, access and moderation:

```
PORTAL=https://data.gov.kg
curl -s "$PORTAL/api/3/action/status_show" \
  | jq '{ckan_version: .result.ckan_version,
         access: [.result.extensions[] |
select(test("saml|oidc|oauth|ldap|keycloak|sso|security|2fa|restricted|approval|workflow|dat
arequest|captcha"))]}'
```

Whether registration is open. When it is closed CKAN returns code 403; when open it shows a form with a password field:

```
curl -s -o /tmp/register.html -w "%{http_code}\n" "$PORTAL/user/register"
```

```
grep -c 'name="password1"' /tmp/register.html
```

Whether the user list is open and whether it contains obvious spam. A 403 response means the list is closed. The search word is common in the names of spam accounts:

```
curl -s "$PORTAL/api/3/action/user_list?q=gacor" | jq '.result | length'
```

For Zenodo and InvenioRDM, how many records are open, restricted and under embargo:

```
for s in open restricted embargoed; do
  curl -s "https://zenodo.org/api/records?q=access.status:$s&type=dataset&size=1" \
    | jq --arg s "$s" '{status: $s, records: .hits.total}'
done
```

For Dataverse, how many files are open, restricted and under embargo:

```
curl -s "https://dataverse.harvard.edu/api/search?q=*&type=file&per_page=0&show_facets=true"
\
  | jq '.data.facets[0].fileAccess.labels'
```

Run checks slowly: one request at a time to a site, not dozens in parallel.

Measurement on 1,125 CKAN portals

On 12 October 2026 all 1,839 active CKAN catalogs from registry release 1.27.0 were checked. The `status_show` service address answered for 1,136 of them, which is 1,125 distinct servers; the rest had closed it, hidden CKAN behind their own site or did not respond. For each server the core version and extensions were recorded, along with whether the registration form is open to an anonymous visitor, whether the user list is open, and how many accounts in it contain the word "gacor". This word comes from Indonesian online casino advertising and does not occur in the names of ordinary users.

Registration is open on 123 servers out of 1,125, or 11%. 29 of them have a CAPTCHA.

CKAN version	Servers	With open registration
older than 2.9	206	58 (28%)
2.9	247	19 (8%)
2.10	411	16 (4%)
2.11	243	28 (12%)
2.12 and later	18	2

In older versions registration is open by default, and almost a third of installations older than 2.9 still run that way. Upgrading by itself does not help: on 2.11 registration is open on 12% of servers, meaning it was enabled deliberately or the setting was carried over during the upgrade. Among them is Kyrgyzstan's national portal data.gov.kg on CKAN 2.11.0.

The user list is open on 547 servers, almost half. In CKAN it is open by default (`ckan.auth.public_user_details`) even when registration is closed, and the documentation advises closing it together with registration.

Spam is where registration is open. Among the 547 servers with an open list, accounts with the word "gacor" were found on 38, 337 accounts in total. Of the 81 servers where both registration and the list are open, 31, or 38%, have spam. Of the 466 servers with closed registration, 7 do: these are traces of the time when registration was open. On the portal of Portugal's Ministry of Justice, dados.justica.gov.pt (CKAN 2.7.2), there are 91 such accounts. On data.gov.kg there are 19, with names like `agenslotgacorslotgacor`, created in 2026. One word gives a lower bound: spammers have many other names.

Login and access extensions are installed on few portals. On 1,125 servers: ckanext-security on 47, SAML login on 27, LDAP on 23, OAuth2 on 21, OpenID Connect with PKCE on 10, Keycloak on 7, ckanext-restricted restricted files on 13, data requests on 19, moderation on 3. Separately, two-factor login in the Thai standard build on 160 servers. The names in the `status_show` response are module names, not repositories, so one name can mean different branches of one extension. Of the 19 portals with data requests, 12 have open registration: exactly the combination on which spam grew on the Tunisian meteorological service portal.

What follows from the measurement. The first thing a CKAN portal owner should do is open their own `/user/register` page and user list without logging in. If registration is open and outside people do not publish, close it, and delete spam accounts together with their organisations, groups and requests. CKAN has the `ckan clean users` command for this, which deletes accounts with invalid avatar files; the rest has to be found by name and content.

What follows from this

- Start with two roles (administrator and organisation editor) and a draft. Introduce moderation only when people you do not trust to publish data straight away are publishing, and when someone is available to do it.
- Keep the dataset description open at any level of file access. Write access conditions, procedure and response time on the dataset page and in the dump: `dct:accessRights` in DCAT, the access right from the COAR vocabulary in DataCite, `isAccessibleForFree` in schema.org.
- If restricted access is the catalog's main task, choose a platform where it is in the core: Dataverse, InvenioRDM, DSpace, NADA. In CKAN a private dataset is hidden together with its description, and a restricted file is someone else's extension.
- Close registration if outside people do not publish, and close the user list. If they do publish, you need an invitation or a check, a CAPTCHA and restricted permissions for new accounts. Check this on your portal today: one in nine CKAN portals is open for registration, and a third of those already have spam.
- Set up single sign-on when there are many publishers and nobody to create and disable them manually. In CKAN it is an extension: test it on a test instance before every core upgrade.
- A reader does not need an account. Descriptions and open files without a login, a key or a CAPTCHA.

Sources

- re3data, repository description schema 4.0 and XSD: access to the repository, to data and to upload as separate questions
- CKAN 2.12: organisations, roles and collaborators, configuration options: `ckan.auth.create_user_via_web`, `ckan.auth.public_user_details`, `ckan.auth.user_create_organizations`, `ckan.auth.allow_dataset_collaborators`, reCAPTCHA; API and tokens; CKAN changelog and PR 7208: registration closed by default "so as not to open the site to spam"; the default value checked in the code of tags 2.9.7, 2.9.8, 2.10.3 and 2.10.4; source code of tag ckan-2.12.0: drafts, the `activity` module, `user_invite`, `expire_api_token`
- Spam on CKAN portals: registration closed on DataHub.io, May 2013; demo.ckan.org filled with spam, 2021; the account clean-up command; I. Begtin, on the price of openness, Telegram (in Russian), 25 November 2025: the Tunisian meteorological service portal
- CKAN extensions: ckanext-saml2auth, ckanext-saml, ckanext-oidc-pkce, ckanext-ldap, ckanext-security, ckanext-restricted, ckanext-approvalworkflow, ckanext-workflow, ckanext-datarequests, ckanext-recaptcha: automated test files, releases and change dates, checked 12 October 2026
- Dataverse: roles and permissions, restricted files, terms of access, guestbook, embargo, review before publication, login methods
- InvenioRDM: restricted records and access requests, communities; Zenodo: access to records
- DSpace (documentation for the 9.x line; the same mechanisms in 10.x): item and file permissions, embargo, authentication and authorisation
- Data access types in NADA
- GeoNetwork 4.4: users and groups, record privileges, record life cycle, self-registration, authentication modes; default values in data-db-default.sql
- udata: organisation roles, changelog (the partial editor in 15.4.0), spam flagging; data.gouv.fr: publishing a dataset and drafts, user accounts and ProConnect
- DKAN: the dkan_publishing workflow, roles and permissions
- Huwise (OpenDataSoft): dataset permissions and access requests, portal security settings, workspace permissions
- Socrata (Tyler Data & Insights): roles, approvals before publication, single sign-on, catalog and private datasets
- ArcGIS: item sharing levels, roles, secured services, Hub Premium community accounts
- Hugging Face: gated datasets, organisation roles, single sign-on
- GitHub: code owners, protected branches; GitLab: merge request approvals
- Access conditions in metadata: DCAT 3, DCAT-AP 3.0.1, the EU Publications Office access-right vocabulary, HealthDCAT-AP, the COAR access rights vocabulary, OpenAIRE guidelines, schema.org conditionsOfAccess, dataset markup for Google
- The Data Governance Act, Regulation (EU) 2022/868: Articles 5 and 8; the European Register for Protected Data and its harvesting requirements; the European Health Data Space, Regulation (EU) 2025/327: Articles 77 and 105

- Publisher login: guidelines on publishing open data, version 4.0 (in Russian), December 2024; publishing descriptions on data.norge.no; ProConnect
- Login standards: SAML 2.0, OpenID Connect Core 1.0, OAuth 2.0, RFC 6749
- Measurement on 12 October 2026: the search APIs of Zenodo and Harvard Dataverse; 1,839 active CKAN catalogs from registry release 1.27.0, the `status_show`, `/user/register` and `user_list` addresses
- The dataportals-registry export, release 1.27.0 of 15 October 2026: queries in the appendix
- Licences and Access Conditions, Microdata Catalogs, CKAN: Core and Extensions

Deployment Speed and Maintenance

A fast launch and a cheap ten years are different answers, and they are recorded separately. Catalogs are shut down when there is nobody to update them, not when the platform is "bad".

The launch is visible in the first week: a page and the first dataset have appeared. Maintenance is visible three years later, when the project has ended but the catalog still has to open, accept edits and install security fixes.

What exactly is assessed

Two quantities.

Time to the first real dataset. Not to a slide and not to the vendor's demo data. The dataset is yours: with a licence, a file and a description that gets into the dump.

The cost of the next year. Who upgrades the version, who checks extensions, who answers the "the link does not open" email, where the backups are, what happens if the subscription or the contractor runs out.

A platform with a short installation and expensive maintenance is fine for a pilot. A platform nobody is available to update is not fine for a catalog that must outlive a grant.

Rough timelines

The timelines are orders of magnitude for a team that already knows how to administer servers. A pilot with ten datasets is shorter. A catalog with customisations and migration of old data is longer.

Method	To the first dataset	Who updates afterwards
Publishing in someone else's catalog	hours	your own descriptions and files; the server is someone else's
A static site or repository	hours or days	files and pages; a server is hardly needed
A cloud service	days	the provider; you pay the subscription
Open source software in a ready-made container	days or weeks	your own team: server, backups, version
Open source software with extensions and a custom schema	weeks or months	the same, plus checking every extension
A custom build	months	everything, including security and other people's formats

The figures are not a contractor's promise. They are for comparing methods with each other, not for putting a deadline in an order.

What eats time after launch

- **Security updates.** A platform that has not been updated for two years can no longer simply "bump the version": along the way the extensions, the schema and the way files are stored have changed.
- **Extensions.** Each is checked on the new core version. One abandoned extension keeps the whole portal on an old version. How to assess this is covered in the chapter on extension ecosystems.
- **Backups and space.** Your own file storage grows unnoticed. A backup nobody has tried to restore does not count as a backup.
- **People.** You need at least one person who answers emails and releases updates. Without this even a successful launch turns into an abandoned site. What to do when a catalog is shut down after all is covered in the chapters on the lifecycle and on archiving.

A fast launch matters more for a pilot and a one-off publication. A low maintenance cost matters more for a catalog that must live ten years. How to record such a choice is covered in the chapter "Justifying a Decision".

What to choose when

Situation	What matters more
Ten datasets must be shown by the end of the quarter, and a team is not promised for a year	someone else's catalog, a static site or the cloud
The catalog will be written into law or a multi-year grant, and there is an administrator post	an open source platform; launch time is secondary
Mandatory features exist only in extensions not tested on the current core	a different method: a cloud service where the feature is built in, or another platform
A contractor delivered the portal and leaves with the acceptance certificate	the contract includes a year of maintenance and a dump of all descriptions and files; otherwise the launch was someone else's

How it is done in platforms

Checked against platform documentation in October 2026. The timeline in a cell is the same order of magnitude as in the methods table, not a promise.

Platform	To a pilot	How versions live	What to check before installing
CKAN	days or weeks: there is an official Docker setup	the latest minor version is fully supported (since August 2026, 2.12); the previous one gets bug and security fixes	every extension on this core version; see CKAN: Core and Extensions
InvenioRDM	the first installation is heavier than a static site	one major version a year; fixes for at least a year and another six months after the next one. v14 came out on 7 August 2026, v13 is supported until 7 February 2027	whether you can keep up with yearly upgrades
Dataverse	there are installation and upgrade instructions	since version 6.8 containers are considered ready for production; several application servers in containers are not supported	the DataCite agreement is separate from installation. The current release is 6.12.1; live sites lag behind

Platform	To a pilot	How versions live	What to check before installing
DSpace	like other research repositories	official upgrade instructions; the current release is 10.1, and the registry still shows 8.x and 9.x	for a library that already has DSpace, upgrading is cheaper than a new installation
GeoNetwork	official instructions, including Docker	country profiles are schema plugins, checked at upgrade	a map server is usually needed alongside: GeoNetwork describes a layer, the server serves it
Huwise, Socrata, ArcGIS Hub	days	the provider does the updates	how to take out all descriptions and files if the subscription ends
A static site	hours or days	nothing to crash in the usual sense	who will build the site when the only person who knew how leaves

Research platforms are heavier to install than a static site, but DOIs, protocols and roles are already in the core. The cloud has a short launch time as long as the subscription and hosting location suit you. Leaving later costs as much as the dump and the address migration. The cost of leaving is analysed in the chapter on build models.

How to assess before buying or installing

- Is there an official container and a short upgrade guide?
- How many recent versions are still supported, and how often do they come out?
- How many extensions are needed for mandatory features, and who releases them?
- Who will be on duty two years from now: your own staff post, a contractor or the service provider?
- Can all descriptions and all files be exported? If not, a short launch is paid for with an expensive exit.

If there is no answer to the question of who will be on duty, a simpler publishing method is usually more honest than a beautiful portal.

How to check a live portal

Not every platform shows its version from outside. Where it does, compare it with the supported version, not with "the newest one in general".

In CKAN the version and the list of plugins are served by a service address:

```
curl -s "https://portal.example/api/3/action/status_show"
```

In the response look at `ckan_version` and `extensions`. A version two generations older than the supported one is a reason to ask who updated the portal and when. An extension in the list that is not in your matrix is a reason to ask why it is installed.

A cloud service often has no version visible from outside. Then check something else: whether the full metadata dump opens and whether a file can be downloaded without logging in. That is exactly what you will take with you.

A footer date of "updated in 2019" on a live site means little by itself: footers get forgotten. The date of the most recently changed dataset and the age of the version say more.

Sources

- Installing CKAN with Docker, upgrading CKAN, CKAN release support policy
- InvenioRDM maintenance policy: v13 and v14 support periods, checked 11 October 2026
- Installing Dataverse, upgrading, containers in production
- Installing GeoNetwork, with Docker
- Extension Ecosystems
- Build, Open Source or SaaS

Other Capabilities

Search, preview, multiple languages, harvesting other catalogs, statistics and feedback end up in the specification because a large portal has them. Each has a threshold. Until it is crossed, a metadata dump, a licence and permanent addresses come first.

In brief:

- Full-text search over descriptions and filters are needed when there are hundreds of datasets and people search by topic. Dozens of datasets with clear titles are fine with a list.
- Preview helps choose a file. A citation in a paper and a program's request go to the file and the API: preview is not a source for them.
- Translating buttons and translating titles are different jobs. Search in a second language finds a dataset only if its title or description is written in that language.
- Harvesting other catalogs is needed by those who show other people's datasets. For a portal with its own datasets it adds records without an owner.
- A page view counter does not see downloads through the API. Such a figure is no good for an annual report: how a catalog's numbers mislead is covered in the chapter on metrics.
- A form without a person who reads it and replies within a stated time is worse than a contact on the dataset page.

How harvesting works at the source and at the harvester is covered in the chapter "Federation and Harvesting". Here: the threshold, where the capability sits in the platform, and a check of a live site.

When each is needed

Capability	Needed when	Can do without when
Search and filters	there are hundreds of datasets or more, and people search by topic, organisation and format	there are dozens of datasets, each with a clear title
Preview	a person needs to understand a file without downloading it	files are small, or only a program reads the data
Multiple languages	several official languages or an international audience, and someone available to maintain translated descriptions	one city and one language
Harvesting other catalogs	a regional or national portal, or an aggregator	the portal publishes only its own data
Usage statistics	decisions on what to update are based on them, and both pages and the API are counted	one "big number" is needed for the annual report
Feedback	a public portal, with a named person who reads messages	an archive kept under grant rules: a contact in the description is enough

Search through the API is a neighbouring capability covered in the chapter on APIs and bulk download. In CKAN the search box on the site and a program's request look into the same index, so a word missing from the site's results will not be found through the API either.

Search and filters

Search by words and filters by organisation, theme and format. On a geoportal, search by map is added.

A list is enough when there are dozens of datasets. A separate search index (a copy of the descriptions organised for fast search) is needed when there are thousands of datasets and people filter rather than scroll. If the index has drifted from the description database, the dataset exists but search does not show it.

Search by title and search by description are different index settings. The word people search for is often in the body of the record, not in the title. An index that the description did not get into will not find such a word.

CKAN's index is Solr, part of a standard installation. An ordinary query searches the combined `text` field: title, description, tags and other record fields go into it. A query with a colon searches one field, for example `title:budget`. Filters by organisation, group, tag and file format are also in the core. In Dataverse 6.12, Solr is a mandatory part of the installation, and its API must not be exposed externally: whoever reaches Solr can change the index behind the catalog's back. In GeoNetwork 4.4 the index is Elasticsearch (the October 2026 documentation recommends version 8.19.13, minimum 8.14); the index's own port is not exposed, and search goes through the catalog. InvenioRDM's installation guide names OpenSearch. udata's search is Elasticsearch, and it is not updated by itself at the moment a record is saved.

On 14 October 2026 on data.gov.gr (CKAN 2.11.3), a query with the word `Guarantees` returned 6 datasets. The Greek title of the first does not contain this word: it is in the English translation of the title. So the translated title got into the index. On the same day, a CSV format filter selected 5,599 datasets from the catalog: the format filter on this portal works.

Check: type a word from the middle of a description, not from the title. Then filter by one organisation and one format. If there is no filter and there are more than a few hundred datasets, people have to scroll through results.

Preview

A table, chart or map on the dataset page helps decide whether this is the right file. A preview cannot be cited and cannot be reliably fetched by a program. The copy used for display goes stale separately from the file: it is checked just like the source file. See the chapter "Anatomy of a Catalog".

In CKAN, view types are plugins shipped with the core: table (`datatables_view`), text, image, video, audio, embedded web page. The table is enabled only if the rows are in the DataStore. The DataStore itself is part of the core, while loading CSV into it is done by the XLoader extension. The CKAN 2.12 documentation no longer has the Recline viewers: the table is shown by `datatables_view`. A layer map comes from the ckanext-geoview extension, release 0.3.2 of 2 October 2026. The browser will not show a file on someone else's site from a CKAN page by itself: a resource proxy is enabled for this, and the proxy has a size limit. The embedded page view (`webpage_view`) is not enabled where anyone can create datasets: it shows someone else's address inside the portal.

On data.gov.gr on 14 October 2026, `datatables_view`, `datastore` and `xloader` are all enabled. For the dataset on state guarantees for December 2023, the XLS file is not loaded into the DataStore. The CSV of the same dataset is loaded, but the field `datastore_contains_all_records_of_source_file` is `false`: the database does not hold the whole file. A table preview depends on the specific file and may show only part of it.

Dataverse has a mechanism for connecting viewers in the core (external tools), while the viewers themselves are a separate community set, dataverse-previewers. The administrator installs them. On the file page, a restricted file's preview is not shown. It is seen by those who already have the right to download the file, including through a separate preview link to a draft. So the phrase "Dataverse has preview" is true for an installation where viewers are enabled, and for a file this person is allowed to download.

In InvenioRDM viewers are shipped with the platform: PDF, CSV, images, Markdown, JSON, XML, ZIP, Jupyter notebooks; since release v12 also audio and video. Some of them, including CSV, read the file from the installation's own storage. In Socrata the dataset page is a table and an exploration canvas: the statistics guide calls them ordinary view pages. GeoNetwork's core has a layer map, not a CSV table.

Check: open a dataset with a table and a dataset with a restricted file. The first has a table on the page or a direct download link, without promising a table that is not there. The second's contents are not visible to someone the file is closed to.

Multiple languages

There are three different things here.

Interface language. Menus, labels, error messages. In CKAN the interface translations are in the core: the list of languages is set by the `ckan.locales_offered` and `ckan.locale_default` settings. In Dataverse and InvenioRDM additional interface languages are enabled in configuration, and the translations come from the community. In GeoNetwork 4.4 the list of interface languages is chosen in the settings; if there is only one language, the switcher is not shown.

Language of the data. A field such as `dct:language` says what language the table or text itself is in. It is not a translation of the record.

Translation of the record. The title and description in each language in which the portal is used. In CKAN such fields come from the ckanext-fluent extension together with ckanext-scheming. Both are maintained by the CKAN community; as of 14 October 2026 the automated tests on the main branch include CKAN 2.10, 2.11 and 2.12. In GeoNetwork, multiple languages within an ISO record are a core capability: the title and abstract are stored separately for each language, and search can be limited to the interface language or run across all of the record's languages. In Dataverse and InvenioRDM separate field translations appear if the schema provides for them. The "dataset language" field itself does not replace a translated title.

If the buttons are in two languages and the records in one, search in the second language comes back empty: a person switched the language and cannot find a dataset that sits under a title in the first.

data.gov.gr maintains records in two languages: the plugin list includes `fluent` and `scheming`. For the dataset on state guarantees for December 2022, the Greek title, the Greek description and the English title all refer to December 2022, while the English description says "Hellenic Republic Guarantees – January 2026". The translated description points to a different month. Search by the English word still works, because the English title got into the index. A translation nobody checks against the original record drifts from it quietly.

Machine translation of harvested records makes them findable in EU languages: data.europa.eu translates titles and descriptions this way after harvesting. Precise wording in two languages is still written by the publisher. How the harvesting chain works is covered in the chapter on federation.

Check: switch the language and find a dataset with a word in that language. Then open the record and compare the translated description with the text in the original language, not only with the translated title.

Harvesting other catalogs

The portal collects descriptions from other catalogs on a schedule: from DCAT, the CKAN API, CSW, OAI-PMH. Records are taken, not files. For a regional or national portal and for any aggregator this is the main job. An agency that publishes only its own data does not need harvesting: other people's records inflate the counter and leave a second copy without an owner.

Where this lives in the platform, checked in October 2026:

- **CKAN.** The harvesting framework is the ckanext-harvest extension. RDF harvesting is done by ckanext-dcat, geocatalog harvesting by ckanext-spatial. Who maintains them and which core version they work with is covered in the chapter on CKAN.
- **udata.** Six harvesters ship with it: DCAT, two variants of CSW, CKAN, DKAN and the French Ministry of Agriculture's format. None is enabled until it is listed in the `HARVESTER_BACKENDS` setting. An administrator also has to validate a new source, otherwise harvesting does not start. A record written to the database does not appear in search by itself: after a run the index is rebuilt separately. On data.gouv.fr on 14 October 2026 the list of sources returned 502 records. That night the source "Communauté de Communes du Pays de Landivisiau" (a CSW harvester, the GeoBretagne catalog) finished without errors and collected 4 datasets.
- **Dataverse.** A harvesting client is in the core and uses OAI-PMH, including against other Dataverse installations. A harvested dataset in search leads to the source repository and cannot be edited locally. The OAI-PMH server, which serves your own datasets outward, is off by default: that is serving, not harvesting.
- **Piveau.** DCAT harvesting is the core's main function.
- **GeoNetwork.** The core harvests CSW, OAI-PMH, OGC services and STAC collections.
- **Huwise.** Harvesters are part of the service: CKAN, Socrata, ArcGIS, ArcGIS Hub, a `data.json` file, data.gouv.fr and other Huwise portals. CSW harvesting is off by default and is enabled through support.

- **Socrata (Tyler Data & Insights).** The Catalog Connector in the admin panel takes a `data.json` file and a public ArcGIS Server, plus there is exchange between Socrata's own domains. Datasets from `data.json` become public after synchronisation: a dataset private at the source will be opened again on the Socrata side. This is checked before connecting another catalog with non-public records.
- **JKAN and a static site.** There is no harvesting in the template: the `data.json` file is assembled by the generator from pages. Other catalogs are collected by a separate script in the same repository, as on opendata.scot.

For ArcGIS Hub, harvesting arbitrary external catalogs was not confirmed in this check. For InvenioRDM, DSpace and NADA, a harvesting client was not settled by the documentation in this check.

On data.gov.gr, a dataset collected by the DCAT-AP harvester has the address of the original (`https://portal.raise-science.eu/datasets/...`) and the name of the harvest source on its page. A reader can tell someone else's publication from the city's own.

Check: the list of sources has a live address, the date of the last successful run and the number of datasets collected. A harvested dataset's page shows a link to the original. A word from the title is found by search: a record written to the database without an index rebuild is not visible on the site.

Usage statistics

Page views, file downloads, API calls. They are used to decide which datasets to update first and to show that the catalog is used. A single "portal views" figure is no good for this: it is easily inflated and does not see those who fetch files by program.

In the CKAN 2.12 documentation, view counting is the `tracking` plugin shipped with the core. It is added to the plugin list. In CKAN 2.10 and earlier the same counting was enabled by a setting inside the core. The plugin records views of dataset and resource pages. The summary the site shows is updated by the `ckan tracking update` command; to get the figures into search, the index is rebuilt afterwards. The `recent` field is views over 14 days, `total` is all counted views. This counter does not see API calls. On data.gov.gr the plugin is enabled: on 14 October 2026 one harvested dataset's summary showed 51 views in total and 31 in the last 14 days, while the neighbouring dataset had zeros for both. Zero means the page has not been opened since counting started seeing it.

A separate route is the ckanext-googleanalytics extension (CKAN community). Automated tests on the main branch run on CKAN 2.10, 2.11 and 2.12. The chapter on CKAN notes that the extension's documentation names versions up to 2.11. Web server logs remain a third method when you need to see requests that the page does not count.

Dataverse has an ordinary download counter without additional installation. Counting under Make Data Count rules is a separate program, Counter Processor, run daily, with exchange with DataCite. Make Data Count collects views, downloads and citations of research data according to the

COUNTER rules for data. Citations from this exchange arrive only for datasets with a DOI. An installation on Handle can count views and downloads locally, but they are not sent to the DataCite hub.

In InvenioRDM since release v12, view and download statistics ship with the platform and follow the same COUNTER and Make Data Count rules. By default the record shows unique views and downloads: a repeat visit from the same source is not added a second time. The current release in the line is v14.

In Socrata statistics are part of the service. The Site Analytics dataset counts page views, downloads and reads through the API, so programmatic access is visible in this system. In udata the counters are in the dataset record. On data.gouv.fr on 14 October 2026, for the dataset "Trajets réalisés en covoiturage" (the register of verified carpooling trips), the API showed 18,334,348 views and 63,155 file downloads. The numbers side by side answer different questions: the page and the file. For a demand report, take the number that matches the way the data is used.

Check: open the statistics of a dataset downloaded by script and of a dataset opened in a browser. If the first shows zeros, the counter sees only pages.

Feedback

A "found an error" button, a discussion on the dataset page, a "request a dataset" form, a showcase of reuses. A reuse is an application, article or visualisation based on a dataset and linked to its page. For a public authority's portal this is a way to see demand. For an archive where data is deposited under grant rules, a contact in the description is enough.

A form without a person who reads it is worse than a contact on the dataset page: messages pile up, and the user assumes the request has been accepted. Before adding a form or a showcase, name who is responsible and within what time.

In udata, discussions and reuses are in the core. On data.gouv.fr on 14 October 2026 the portal summary showed 17,732 discussions and 5,996 reuses. The same carpooling dataset had 23 discussions, 19 of them still open, and 5 reuses. A message counter does not say whether anyone replied.

In CKAN a reuse showcase comes from the ckanext-showcase extension. It is maintained by the CKAN community; tests on the main branch run on 2.10, 2.11 and 2.12, and the latest release as of 14 October 2026 is 1.8.4 of 22 July 2025. On CKAN 2.12, file storage is configured separately for showcase images. "About the portal" pages come from ckanext-pages, also from the CKAN community. Its latest release is v0.5.2 of 7 June 2023. Support for CKAN 2.12 was merged into the main branch on 26 June 2026, with no release since: on 2.12 the extension is installed from the main branch. The tests of that branch include 2.10, 2.11 and 2.12. At a core upgrade this is a separate check even when tests are green: there is nothing to install except a snapshot of the branch.

In Dataverse a dataset page has a contact, and a restricted file has an access request. This is core. The guestbook on download records a visit; it does not replace a message to the owner.

Check: send an error report about one dataset and see whether it reached the data owner. A shared mailbox nobody opens fails the check.

How it is done in platforms

The marks are core, extension, no, not checked, as in the summary table. "Core" here means "ships with the platform". A CKAN plugin that sits in the same distribution and only needs enabling is recorded as core. Checked against documentation and live portals on 14 October 2026.

Platform	Search	Preview	Translated records	Harvesting	Statistics	Feedback
CKAN 2.12	core (Solr)	core (view types); table with DataStore; CSV loading via XLoader; map via ckanext-geoview	extension (ckanext-fluent and scheming); buttons in the core	extension (harvest, dcat, spatial)	core (the tracking plugin, page views); ckanext-googleanalytics is an extension	showcase and pages are extensions; contact is a record field
Dataverse 6.12	core (Solr)	mechanism in the core, viewers installed by the administrator	interface language is a setting; field translation if the schema has it	core (OAI-PMH client)	download counter in the core; Make Data Count is a separate program	contact and access request in the core
InvenioRDM v14	core (OpenSearch)	core	interface language is a setting	not checked	core since v12 (COUNTER, Make Data Count)	not checked
GeoNetwork 4.4	core (Elasticsearch)	layer map in the core	core (languages in the ISO record and in the interface)	core	not checked	not checked
udata	core (Elasticsearch)	not checked	not checked	core (six harvesters, off by default)	the core stores counters	core (discussions and reuses)
Huwise	core	not checked	not checked	core (CSW enabled on request)	not checked	not checked
Socrata	core	core (table and exploration canvas)	not checked	core (Catalog Connector)	core (views, downloads, API reads)	not checked
ArcGIS Hub	core	not checked	not checked	not checked	not checked	not checked
JKAN, a static site	a JSON filter in the browser, no API	no	no: the author writes the text	not in the template; a script is written separately	no	a contact in the page text

A "not checked" cell means the documentation in this check gave no answer. Before choosing a platform, settle it from that platform's guide.

Example: a city copies a large portal

In brief, this is what a justification looks like using the decision record template:

- **Context.** A city publishes about 40 datasets in one official language, with no other catalogs to harvest. The national portal's feature list was copied into the specification: search with filters, table and map preview, three languages, harvesting district portals, statistics for the annual report, a "found an error" form and a reuse showcase.

- **Options.** Require the whole list and cover it with CKAN extensions. Take Socrata or another cloud service where the table and statistics are already part of the service. Put search and a contact in the requirements, and record the rest as rejected with a threshold for revisiting.
- **Criteria.** The number of datasets and the number of languages in which records will actually be maintained. Whether there are other catalogs the city is obliged to harvest. Whether a person is named who reads the form and updates translations.
- **Decision.** The requirements keep search by title and description and a contact on the dataset page. A table preview is desirable only together with table storage, for the CSVs loaded there. Record translation, harvesting, the showcase and statistics for the report are not in this round.
- **Why.** The full list of CKAN extensions does not pass the threshold: there is no second language, districts do not hand their catalogs to the city, nobody is available to read the form, and every extension is checked at each core upgrade. Socrata, where the table and statistics are already in the service, fails the same threshold: these features do not become a task just because they are included in the subscription. Search and a contact cover today's 40 datasets.
- **When to revisit.** There are more than a few hundred datasets and a word from a description cannot be found. A second official language has appeared and a person is named to maintain translated records. The city has been tasked with harvesting district catalogs. A staff member and a response time have been assigned to the form.

Publishing without a portal

On a static site and in a Git repository, almost all six capabilities are left to the page author.

Search means site search provided by the platform or a list with clear titles. The author adds a separate index when there are many pages. There is no preview in the catalog sense. GitHub shows a small CSV or TSV as a table; the platform's limits are covered in the chapter on formats. There is one language, the one the page is written in; a second language means a second page, which also has to be findable by search. There is no harvesting of other catalogs. The platform's statistics count visits to the repository or site and miss those who took a file by direct link. Feedback is an address in the page text and issues in the repository, if someone reads them.

This is enough for a small set of pages. The thresholds from the table above are the same here: hundreds of files, a second language, an obligation to harvest other catalogs and a report on API demand bring you back to a catalog. How to make such a publication findable is covered in the chapter "Making a Publication Findable".

How to check a live portal

The commands below were run on 14 October 2026.

Search by a word that is not in the default-language title but is in the translated title. CKAN's response gives the number of datasets found in `result.count`:

```
curl -s "https://data.gov.gr/api/3/action/package_search?q=Guarantees&rows=1"
```

On that day the count was 6. The first dataset's Greek title is about state guarantees for December 2023; the English one contains the word Guarantees.

A format filter. The count is in the same field:

```
curl -s "https://data.gov.gr/api/3/action/package_search?fq=res_format:CSV&rows=0"
```

Comparing a translation with the original record. Look at `title_translated` and `notes_translated`:

```
curl -s "https://data.gov.gr/api/3/action/package_show?id=eggyiseis-ellinikoy-dimosioy-dekemvrioy-2022"
```

Page views, if the tracking plugin is enabled on CKAN. The search or record response has a `tracking_summary` object with `total` and `recent` fields.

A udata portal summary: the number of datasets, discussions and reuses is in `metrics`.

```
curl -s "https://www.data.gouv.fr/api/1/site/"
```

The list of harvest sources and the number of records, in the `total` field:

```
curl -s "https://www.data.gouv.fr/api/1/harvest/sources/?page_size=1"
```

For a single source, look at `last_job.status`, `last_job.ended` and `last_job.items`. For the Landivisiau source that night the status was `done`, with no errors and 4 datasets.

On any other portal, the same questions without these addresses: a word from the middle of a description is found; there are filters by organisation and format when there are hundreds of datasets; a restricted file's preview is empty; after switching language, search in that language finds something, and the translated description matches the original; a harvested dataset has a link to the source and the date of the last run; the counter for script downloads is not zero if scripts are the main way of access; an error report reaches the data owner.

Sources

- Search in the CKAN 2.12 user guide, data viewer, page view tracking, DataStore, interface languages via the `ckan.locales_offered` and `ckan.locale_default` keys in the CKAN 2.12 config declaration
- ckanext-fluent, ckanext-scheming, ckanext-harvest, ckanext-spatial, ckanext-googleanalytics, ckanext-showcase, ckanext-pages, XLoader, ckanext-geoview: automated test files and releases, 14 October 2026. ckanext-pages's latest release, v0.5.2, is from 7 June 2023; CKAN 2.12 support was merged into the main branch on 26 June 2026. ckanext-showcase's latest release is 1.8.4 of 22 July 2025
- Dataverse 6.12 external tools, dataverse-previewers, preview and restricted files, harvesting client, Make Data Count, Solr in the installation prerequisites, interface languages

- Invenio previewers, statistics in the v12 release notes, interface languages, OpenSearch in the InvenioRDM installation guide
- GeoNetwork 4.4 interface languages, multilingual fields in the index, installing Elasticsearch
- Harvesting in udata: six harvesters, off by default; search is updated by a separate index rebuild
- Huwise harvesters, checked 14 October 2026; CSW harvesting is off by default: CSW harvester
- Socrata's Catalog Connector: datasets from `data.json` become public after synchronisation; Site Analytics: views, downloads and API reads
- Piveau
- The data.gov.gr and data.gouv.fr APIs, 14 October 2026: search for the word Guarantees, the CSV filter, the December 2022 dataset record, the portal summary and the list of harvest sources
- Federation and Harvesting
- Capability Map

PART V.

Platforms and Extensions

Which software data catalogs are built on, what the platform cores contain, and what extension ecosystems add.

1. Classes of Software
2. Extension Ecosystems
3. CKAN: Core and Extensions
4. Platform Profiles

Classes of Software

A catalog type says **what** is published. A software class says **who** updates it, whether a feature can be added and what it costs to leave. Within one class, platforms differ in extensions: see extension ecosystems.

Classes

Class	Examples	When it fits
Open source portal platform	CKAN, DKAN, uData, Piveau	there is your own team or a contractor, and you need your own schema and extensions
Research repository	Dataverse, DSpace, InvenioRDM	DOIs, versions and citation are needed
Geocatalog	GeoNetwork, GeoNode; the cloud-based ArcGIS Hub	a catalog of layers and services
Indicator platform	PxWeb, .Stat Suite, OpenSDG	official statistics and indicators
Microdata catalog	NADA	surveys with different access levels
Cloud service	Huwise (OpenDataSoft), Socrata, Figshare	no operations of your own, a subscription fee instead
A site on a CMS	WordPress, Drupal	the catalog is a section of an existing site, and the CMS can serve descriptions to machines
Protocol server	THREDDS, ERDDAP, a STAC server, SPARQL	programs read the data; a showcase site is secondary
Map server	GeoServer, MapServer, QGIS Server	serves layers; is not a catalog by itself
Domain system	GBIF IPT and similar networks	data of one scientific discipline
Custom build	a site written for one organisation	no platform covers a mandatory feature

A map server often sits **next to** a geocatalog, not instead of it. GeoServer serves the map, GeoNetwork describes the layers so they can be found. A protocol server likewise sits next to a catalog: THREDDS and ERDDAP serve arrays and tables to programs, while the page for people is kept by another site.

Four questions for a class

Brands are compared after the class has answered four questions. The answers by brand are in the profiles and in the chapter on build models.

Question	Open source platform	Cloud service	Static site or Git	Custom build
Who updates	your own team or a contractor	the provider	whoever edits the files	your own team, entirely
Can a feature be added	yes, with an extension or customisation	only what is in the subscription	by editing the template; a search API does not appear that way	yes, at the cost of maintenance

Question	Open source platform	Cloud service	Static site or Git	Custom build
What you take with you when leaving	your files and database, if backups were made	what the provider gives as an export	the files, which you already have	whatever you planned to export yourself
What the class does not do	does not run itself while there is no administrator	does not accept your module	does not serve a search API or harvest other catalogs	does not know other people's standards until someone writes them in

A CMS site and a protocol server are not in this table, because they are chosen not instead of a portal but alongside one. A CMS fits when the catalog is a section of an already living site and the engine can serve descriptions to machines. A protocol server fits when programs read the data and pages for people are secondary.

A class mistake is more expensive than a brand mistake. CKAN with a DOI extension does not become a research repository: the ckanext-doi extension has not been tested on CKAN 2.11 and 2.12. Dataverse does not become an open data portal with daily DCAT-AP federation. ArcGIS Hub does not issue DOIs. These boundaries are written in the profiles, and a shortlist respects them.

What live catalogs show

According to the dataportals-registry export (release 1.27.0 of 15 October 2026):

- custom-built catalogs are about 16%. Among geoportals they are 4%: ready-made platforms won there, ArcGIS above all. Among indicator catalogs, 44% are custom-built, and this type is the least likely to have an API;
- the ArcGIS ecosystem is about a third of geoportals;
- a municipality often does not choose its platform itself. Russia's Gosweb is a state platform: a city is left to publish where the country already collects data. Poland's e-mapa is a commercial product from Geo-System that each gmina buys, but the vendor serves more than half the gminas in the country, and the choice is in effect made by the market.

A change of owner of a cloud platform or the end of a product hits hundreds of catalogs at once. This happened to microdata catalogs with Nesstar: in 2015, with version 4.0, the product was declared obsolete, by 2022 it was finally withdrawn, and archives moved their collections to other platforms; for example, Canada's ODESI chose Borealis, built on Dataverse, for its move. Even a rename of the vendor without a change of product, as when OpenDataSoft became Huwise in 2025, is a reason to reread the contract. That is why the "cloud service" class in the comparison table always comes with the question: how to take the data out if the subscription ends. This is covered in the chapter on build models.

How to choose a class, not a brand

1. If there is little data and it is updated rarely: someone else's catalog or a static site, not a platform.
2. If there is no team of your own: a cloud service or publishing in an existing catalog.
3. If DOIs and citation are needed: a research repository, not an open data portal with customisations.

4. If the main product is map layers: a geocatalog. A map server is added to it, not installed instead of the catalog.
5. If the main product is indicators with dimensions: an indicator platform, not CKAN with a thousand tiny datasets.
6. If there is a team and mandatory features that the cloud does not have: an open source platform and a list of extensions.
7. A custom build only when points 1–6 do not cover a mandatory feature. The test questions are in the same chapter on build models.

Then specific platforms are compared using the profiles and the matrix in the chapter "From Features to a Platform". A class that has not answered the question "who will be on duty two years from now" drops off the list before its features are counted.

Sources

- The dataportals-registry export, release 1.27.0 of 15 October 2026. The queries are in the appendix.
- Platform Profiles
- Protocol servers and domain systems: THREDDS, ERDDAP, GeoServer, GBIF IPT
- Nesstar: product history, the Nesstar Publisher entry in the CESSDA directory (distributed without support), ODESI's move from Nesstar to Borealis
- Announcement of OpenDataSoft's rename to Huwise
- Geo-System, e-Gmina technology deployments in 2025 (in Polish)

Extension Ecosystems

The feature you need often lives not in the platform itself but in a separate module. A module has its own author, its own releases and its own compatibility with the core version. Comparing platforms by the list of "what the box can do" means seeing neither the capabilities nor the cost of maintenance.

The practical conclusion is this. For each mandatory feature, record where it lives: in the platform distribution, in a named extension, or nowhere. An extension without which a feature does not work is checked before the shortlist, by its repository, tests and branches, not by its marketing description. If the only route to a mandatory feature is a module that is not tested on your core version, the platform does not make the shortlist. The cost of an accepted extension is a check at every core upgrade.

The full table of CKAN extensions is in the chapter "CKAN: Core and Extensions". Here: how ecosystems work in general and how to read one before installing.

What counts as an extension

The single word "plugin" covers five different things. The kind determines whom to ask at upgrade time.

Distribution configuration. A file or switch that comes with the platform. InvenioRDM's custom fields are set in the installation configuration. DKAN's dataset schema is a JSON Schema file in DKAN itself. The author is the same as the platform's, and there is one upgrade.

A module from the same distribution that is enabled. In CKAN the table store, text view and statistics are plugins: their names are written into the `ckan.plugins` setting. They are released together with the core and have no separate repository. In GeoNetwork the ISO 19139, ISO 19115-3, Dublin Core and ISO 19110 feature catalogue schemas are part of the release. This is core. They have no outside maintainer.

A separate extension. Its own repository, its own release, its own table of core versions. ckanext-* packages and country schema plugins for GeoNetwork work this way. You may end up maintaining them, even if the author is the platform's community.

An external program. The catalog does not contain its code but opens it by address. In Dataverse a file viewer is a separate application: a JSON manifest with the address is placed in the installation, and the application itself can run on another site. A core upgrade does not overwrite it. What changes instead is the agreement on how the application gets the file.

A cloud service. You cannot install your own module on an ArcGIS Hub, Huwise or Socrata server. What is left is a widget on the page and the API. The provider does the upgrades. You cannot adapt the service to your metadata profile: if the profile you need is not there, that means another platform or your own program alongside.

The InvenioRDM documentation explicitly ranks customisations from cheap to expensive. First configuration. Then files in expected places: your own template, your own image. Then a ready-made module. Then your own package in the installation folder. Then a change to the core with the change contributed back to the project. At the end, your own copy of someone else's repository, a fork, which you maintain yourself. A logo is changed by configuration. A fork for the sake of a logo means you carry every upstream update over by hand.

How ecosystems differ

Platform	How a capability is added	Who usually maintains it	What to check at upgrade
CKAN	a Python package; the plugin name is added to `ckan.plugins`	the CKAN community or a separate organisation	your core version is in the package's automated tests
DKAN	a Drupal module; DKAN itself is also a Drupal module	CivicActions maintains DKAN, the rest are Drupal modules	two lines: Drupal and DKAN. DKAN release 4.1.5 of 7 October 2026 is declared for Drupal 10 from 10.2 and for Drupal 11
Dataverse	an external program via a manifest, or a format exporter JAR file	the GDCC consortium and separate projects	the manifest and the way the file is accessed. Since version 6.1 a short signed URL is recommended for viewers
GeoNetwork	a schema plugin: a folder of XML schemas and transformations, sometimes a separate JAR	national communities and companies	the plugin branch is named after your GeoNetwork line
InvenioRDM	configuration, files in expected places, then your own `site` package	you yourself, inside the installation	compare your templates with the new version's templates
DSpace	your own code in the `additions` module on top of core classes	you yourself	there is no third-party plugin catalog; your change is reapplied to the new version
GeoServer	an official extension or a community module	OSGeo; community modules are shipped "as is"	a community module is built from the tag of the same version. A build from a nightly snapshot often does not fit an official release
ArcGIS Hub, Huwise, Socrata	you cannot install your own module	the provider	a full dump of descriptions and files in case you leave

DKAN has two update calendars. A hole in Drupal and a hole in DKAN are closed by different releases. The module's page on Drupal.org was updated on 8 October 2026: about 164 sites report using the module. This is a report from Drupal installations themselves, not a count from the catalog registry. Release 4.1.5 is declared for Drupal 10 from 10.2 and for Drupal 11.

In GeoServer the STAC module ships as a pending community module: it is in the release, but without official support. More on the role of GeoServer in the chapter on geoportals.

In Dataverse since version 5.14, a third-party export format is connected as a JAR file inside the installation. This is already a module that is checked at a core upgrade, unlike a viewer on another site.

How to assess an extension before installing

Work through the repository page. A marketing description and the phrase "it worked for the neighbours" do not count.

A fresh release does not by itself prove compatibility. ckanext-doi release 4.0.4 came out on 22 September 2025, while the test images in the repository have not changed since 20 September 2023: the instructions still name CKAN 2.9 and 2.10. The absence of a release is not a verdict

either. In the table in the CKAN chapter, ckanext-pages has had no releases since 2023; it is installed from the main branch, and tests run on CKAN 2.10, 2.11 and 2.12. A strong sign is automated tests on exactly your core version. A branch named `4.4.x` on a GeoNetwork plugin is the same sign, if such a branch exists in the repository.

Question	Where to look	What counts as an answer
Who releases it	the organisation in the repository address, the maintainers file	the platform team or an organisation that responds to issues. One person with no response for two years and no tests is a reason to look for another module
Which core version it works with	the compatibility table, the automated tests file, the branch name	your version is named there. A release newer than the core whose tests are older than your version does not count
Whether it is alive	the date of the last fix and responses to issues about the new core version	tests run on the current line, even if there has been no separate release for a long time
How to install it	documentation in the same repository	the install command points to this repository and to a branch that exists in it
What licence	the licence file	there is a licence, and it permits your installation
Who else installed it	the ecosystem catalog, the API response of a live portal	the module is installed on portals running your core version

The CKAN ecosystem catalog in October 2026 knows about 1.4 thousand extensions and 1.3 thousand portals with core versions. This is the best available approximation, not a full census: the number of portals grew when registry records were loaded into it. It is more reliable to ask the portal itself, as in the section below.

Two extensions of one platform

ckanext-dcat is maintained by the CKAN community. The documentation, checked on 11 October 2026, names core versions 2.10 and 2.11. The tests file on the main branch on 13 October 2026 runs 2.10, 2.11 and 2.12. The latest release is 2.4.4 of 11 August 2026. For a portal that search engines and the national catalog must harvest, it is a normal part of a mature setup.

ckanext-doi is maintained by the Natural History Museum in London. The badge in the repository names CKAN 2.9.9 and 2.10.1, and the tests section the 2.9 and 2.10 lines. The latest release, 4.0.4, came out on 22 September 2025. The test images folder has not changed since: the last commit there is dated 20 September 2023, which is exactly when tests for 2.10 were added. The extension has not been tested on CKAN 2.11 and 2.12. If a DOI is mandatory, look at a platform where it is in the core: Dataverse or InvenioRDM. The requirement is analysed in the chapter on identifiers.

The other CKAN packages are summarised in the table in the chapter on CKAN: the documentation column and the tests column diverge in a few more rows.

A fork

A fork is a copy of a repository maintained by another team. Copies diverge. Install the one whose tests run on your core version, and record its address in the decision record.

Two copies of table validation diverged this way. The original ckanext-validation is documented only for CKAN 2.9. The living copy is the Queensland Government fork: tests on 2.10, 2.11 and 2.12. The Queensland Government maintains it. If you fix the module locally and do not contribute the fix back, a third copy appears, and its maintainer is you.

Two copies of one schema plugin

The same thing on GeoNetwork. The British GEMINI 2.3 profile lives in two repositories.

Astun Technology has a `4.4.x` branch. On 6 October 2026 an update to the instructions went into it. The instructions name GeoNetwork 4.4 as the recommended version and the plugin file as a build for 4.4.12. This is the copy installed on the current GeoNetwork line.

In the shared metadata101 collection there is no `4.4.x` branch. The last commit on the main branch is from 14 February 2023. The `4.2.x` branch stopped on 27 June 2024 on a commit that itself is marked as work in progress.

The trap in Astun's instructions: the automatic install command points not to Astun's repository but to metadata101 and to the `4.4.x` branch, which does not exist there. The heading and the command beneath it disagree. Take the repository and branch you have opened and that have commits for your version. Double-check the command from the text.

Next to it is an example where there is only one copy. The Canadian HNAP profile in metadata101 is released with GeoNetwork version numbers: release 4.4.12 came out on 8 July 2026. The instructions say to install it on GeoNetwork 4.4 and state plainly that it has not been tested on earlier versions.

A bad schema plugin breaks the whole catalog, as the GeoNetwork administrator guide warns. It is checked on a test installation before the production catalog.

An external program

Dataverse viewers work differently. They were developed by the Qualitative Data Repository, and the collection is now maintained by GDCC: version 1.5, 19 viewers, more than 40 file types. Their code is not placed in the Dataverse installation. The administrator registers a manifest, and the file page opens the application by address. The application can stay on github.io.

The risk here is different. A Dataverse upgrade does not hinge on a forgotten package inside the server. But the application gets access to the file. Since version 6.1 the documentation recommends a short signed URL: it is valid for a limited time and allows only the listed API calls. The documentation does not recommend a long-lived API key after Dataverse 6.0: the key acts like the user's password and allows everything that user is allowed.

Before enabling a viewer, record which of the two access methods has been chosen.

Why an extension is dangerous later

- It stopped being released, and the new core version will not install without it. The portal is left without security fixes. In CKAN the latest minor version is fully supported, and the previous one gets only bug and security fixes. When the next minor version comes out, today's 2.11 will

lose those too. One such extension holds back the whole portal.

- Two extensions modify the same form or the same template. Each installs on its own, but together they break the dataset page. The CKAN documentation asks extensions to prefix template, setting and API function names with the extension name so they do not collide.
- A fix was kept locally. From then on every upstream update is carried over by hand. This is now your program, with its own release and its own tests.
- Two calendars, as with DKAN and Drupal. One was patched and the other missed.
- A schema plugin was copied inside the application. When upgrading GeoNetwork, take the plugin branch of the same line, otherwise records in that profile stop opening.
- An external program was given a long-lived key. The key will outlive both the employee and the catalog upgrade.
- A GeoServer community module was taken from a nightly build and put on an official release. The documentation says that this way it often does not start: build it from the tag of the same version.

On the length of maintenance, see the chapter on deployment speed.

How many extensions is too many

Count not "how many plugins are fashionable" but how many separate packages you will have to check and how many of them you maintain yourself. Five tested community packages is an ordinary national setup, analysed in the CKAN chapter. One untested package on a mandatory feature costs more than ten tested ones.

A live portal is almost always longer than the short list in a profile. On 13 October 2026 the `status_show` address responded as follows:

Portal	CKAN version	Names in the list	What the list shows
data.gov.gr	2.11.3	66	familiar community packages (dcat, scheming, harvest, fluent, showcase, pages, archiver) and the agencies' own harvesters, including those of Attica and the Bank of Greece
data.humdata.org	2.11.4	34	18 names start with `hdx_`: this is the portal's own code, not the community catalog

One package registers several names. On the Greek portal, `scheming_datasets` and `scheming_organizations` are one ckanext-scheming. The number in the API response is an upper estimate, not the number of repositories. Names from the CKAN distribution (`datastore`, `datatables_view`, `activity`, `text_view`, `stats`) are subtracted from this estimate: the core team updates them.

Names with the portal's prefix are subtracted in the other direction. No outside author will abandon them, because there is no outside author. Their maintainer is the portal team. For HDX there are 18 such names out of 34. This is a normal state for a large portal and at the same time a budget line: when moving to the next core version, these modules are checked by whoever wrote them.

The same address on data.gov.au responded with code 403 on 13 October 2026. The list is closed from outside, and the installation's administrator has it.

How to record this in the choice

In the platform table, each mandatory feature has three cells: in the core; through which extension and who maintains it; or your own customisation. The decision record contains the repository address, the branch, the core version the tests ran on, and what to do if maintenance stops: stay on the old core, take a named fork or change platform. The template is in the chapter "Justifying a Decision".

The decision is revisited when a new minor core version comes out, when the extension's tests stop including your version, when issues about the new version stop getting responses, or when a second extension starts modifying the same templates.

A platform on which a mandatory feature is reachable only through an abandoned extension does not make the shortlist. How the list comes out of features is covered in the chapter on selecting features.

How to see what is installed on a live portal

In CKAN the list of enabled plugins and the core version are open at `/api/3/action/status_show`. The names in the `extensions` field are plugins, not packages: one package can provide several names.

```
curl -s "https://data.gov.gr/api/3/action/status_show" \
  | jq '{version: .result.ckan_version, n: (.result.extensions | length), extensions:
.result.extensions}'
```

Then each name that is not in the CKAN distribution is looked up in the ecosystem catalog or in the portal's repository and run through the table in the section "How to assess an extension before installing".

In GeoNetwork the full list of schemas is visible to the administrator: "Metadata and templates", then "Standards". From outside you see the schema of a particular record, not the set of installed plugins. In Dataverse, file view buttons are visible on the file page; the manifest with the application address and access method is seen by the administrator.

A summary of "core, extension or no" by platform is in the profiles and the appendix.

Sources

- Writing CKAN 2.12 extensions, best practices for names and interfaces: an extension as a Python package, plugins enabled in `ckan.plugins`
- CKAN release support policy
- ckanext-dcat, the tests file test.yml: versions 2.10, 2.11 and 2.12, checked 13 October 2026; release 2.4.4 of 11 August 2026

- ckanext-doi: release 4.0.4 of 22 September 2025; compatibility badge 2.9.9 and 2.10.1; the last commit to the `docker/` folder on 20 September 2023
- ckanext-validation and the qld-gov-au fork: diverging copies, checked in the CKAN chapter on 12 October 2026
- CKAN ecosystem catalog: about 1.4 thousand extensions and 1.3 thousand portals, October 2026
- `/api/3/action/status_show` responses from data.gov.gr (CKAN 2.11.3, 66 names) and data.humdata.org (CKAN 2.11.4, 34 names), 13 October 2026. The same address on data.gov.au responded with code 403 that day
- InvenioRDM customisation order, custom code in the site folder
- Dataverse 6.12 external tools, GDCC previewers: signed URLs since 6.1, API keys not recommended after 6.0
- Dataverse metadata exporters: a third-party format as a JAR since 5.14
- The DKAN module on Drupal.org: release 4.1.5 of 7 October 2026, Drupal 10.2 and later or Drupal 11; page updated 8 October 2026
- GeoNetwork schema plugins, the warning about breaking the catalog
- GEMINI 2.3, Astun Technology's 4.4.x branch: instructions commit of 6 October 2026; the copy in metadata101: no `4.4.x` branch, the last commit on the main branch 14 February 2023, on the `4.2.x` branch 27 June 2024
- HNAP for GeoNetwork: release 4.4.12 of 8 July 2026
- GeoServer community modules, OpenSearch for EO and STAC: shipped, not officially supported
- The additions module in DSpace 10

CKAN: Core and Extensions

CKAN is an open source platform for open data portals (AGPL-3.0 licence, the CKAN Association). According to the dataportals-registry export (release 1.27.0 of 15 October 2026), about 1,900 open data portals run on it. Almost none of them runs on the bare core: extensions are installed on top of it for specific tasks.

The current core version is CKAN 2.12, released on 26 August 2026. Under the project's rules the latest minor version is fully supported, while the previous one (currently 2.11) gets only bug and security fixes. The CKAN entry in the registry still shows version 2.11.4. The information on extensions below was checked on 11 October 2026 against their documentation and the automated tests in their repositories.

What is in the core

This is enough to publish datasets, assign permissions and serve data to programs:

- datasets, files or links, organisations, groups, tags;
- the Action API for search and writing;
- file upload (FileStore) and a database for tables with an API (DataStore);
- roles and private datasets;
- search on Solr.

The core has no DCAT export, no schema.org markup, no custom fields beyond the standard schema, no harvesting of other catalogs, no spatial search and no DOIs. All of these are extensions.

Open registration. The `ckan.auth.create_user_via_web` parameter lets any visitor create an account. Before CKAN 2.9.7 it was on by default. In 2.9.8 it was turned off, in 2.10.0–2.10.3 it was on again, and it has been off for good since 2.10.4 and 2.11. Portals upgraded from older versions often kept the old setting. On such sites robots register hundreds of thousands of accounts and publish advertising and fraudulent pages in their name: the portal of Tunisia's meteorological service had more than 1.3 million accounts in 2025 and tens of thousands of spam texts in its data requests section. If only staff publish, registration is turned off, and an organisation administrator invites outside publishers.

What changed in 2.12. Files became separate objects: they can be uploaded and managed separately from a dataset's resources and kept in several stores at once. The DataStore got fast pagination for large tables and complex filters. There is also a new interface design that will become the default in CKAN 3.0. For a portal owner this means that moving to 2.12 requires a database upgrade and a check of the extensions that work with files. For example, ckanext-showcase on 2.12 requires storage to be configured separately. Only disk storage ships with 2.12: a portal that keeps files in the cloud changes its storage extension when it moves. More in the chapter on storage.

Extensions that cover typical requirements

The "Tests" column shows which core versions an extension is automatically tested on with every change. This is a strong sign of compatibility even if the documentation text has not yet been updated. The "Documentation" column is what the extension declares itself.

Task	Extension	Maintainer	Documentation	Tests
DCAT, DCAT-AP, DCAT-US, schema.org, Croissant	ckanext-dcat	CKAN community	2.10 and 2.11	2.10, 2.11, 2.12
Custom fields and forms	ckanext-scheming	CKAN community	—	2.10, 2.11, 2.12
Descriptions in several languages	ckanext-fluent	CKAN community	installed together with scheming	2.10, 2.11, 2.12
Harvesting datasets from other catalogs	ckanext-harvest	CKAN community	—	2.10, 2.11, 2.12; the latest release (1.6.2) is older than CKAN 2.12
Spatial search and CSW harvesting	ckanext-spatial	CKAN community	2.10 and 2.11	2.10, 2.11, 2.12
Loading CSV into the DataStore	XLoader	CKAN community	2.10, 2.11, 2.12	2.10, 2.11, 2.12
Storing files in S3 and other clouds	ckanext-file-keeper-cloud for 2.12; before 2.12, ckanext-s3filestore	CKAN community; s3filestore by Keitaro	file-keeper-cloud is named in the CKAN 2.12 documentation	file-keeper-cloud: 2.12 only, no releases; s3filestore: 2.10 and 2.11
Checking links and keeping copies of files	ckanext-archiver	CKAN community	—	2.10, 2.11, 2.12
Validating a table against a data dictionary (Table Schema)	ckanext-validation	Frictionless Data; the living branch is the Queensland Government fork	original repository: 2.9 only, "no" for 2.10 and 2.11	original: 2.9; fork: 2.10, 2.11, 2.12
"About the portal" pages	ckanext-pages	CKAN community	—	2.10, 2.11, 2.12 on the main branch; the latest release, v0.5.2, is from 7 June 2023; 2.12 support was merged on 26 June 2026 without a new release
Dataset reuse showcase	ckanext-showcase	CKAN community	2.10–2.12	2.10, 2.11, 2.12
Statistics via Google Analytics	ckanext-googleanalytics	CKAN community	up to 2.11	2.10, 2.11, 2.12
SAML login	ckanext-saml2auth	Keitaro	2.10 and 2.11	2.10, 2.11; the latest release is from 2022
DOIs in DataCite	ckanext-doi	Natural History Museum (London)	2.9 and 2.10	test images not updated since September 2023, before 2.11 came out; the latest release is from September 2025

Harvesting RDF from someone else's DCAT is done not by harvest itself but by a harvester inside ckanext-dcat. Harvest provides the queue, the schedule and the error screen.

Three working setups

A small portal that search engines and the national catalog must find. The core, ckanext-dcat (the DCAT-AP or national profile), and ckanext-scheming if the standard fields are not enough. XLoader only if tables need to be viewed and queried through the API, not just downloaded.

A regional portal that harvests municipalities. The same setup plus ckanext-harvest. Municipalities then need not "an equally pretty site" but a stable DCAT export.

A portal with many maps. ckanext-spatial is added. If maps are the main product and tables are incidental, it is more honest to look at a geo platform than to keep building on CKAN. Types are compared in the chapter on hybrids.

As of October 2026 all three setups can be installed on CKAN 2.12: their extensions are tested on that version.

DOIs are not part of these setups. An open data portal usually does not need them, and the ckanext-doi extension has not been tested on CKAN 2.11 and 2.12. Research data belongs in Dataverse or InvenioRDM, where DOIs are in the core.

The cost of such a setup

Upgrading CKAN means upgrading the core and checking every extension in the table. A CKAN community extension that is tested on the new core version is predictable. An extension whose documentation and tests are both silent about the new version is a risk: either you stay on the old version or you take on the testing yourself.

Right now two extensions in the table carry such a risk. ckanext-doi has not been tested on 2.11 and 2.12 at all: its test images are older than CKAN 2.11. ckanext-saml2auth is not tested on 2.12. A portal that needs SAML login checks login on a test instance before moving to 2.12, or looks for another single sign-on extension.

Delaying an upgrade is also expensive. The project calls 2.12 the last release of the 2.x line, with CKAN 3.0 planned next. While 2.11 is the previous version in the supported pair, it gets bug and security fixes. When the pair moves on, 2.11 will lose them. After three skipped releases an upgrade turns into a re-implementation.

How to make this decision is described in the chapter on ecosystems and in the example decision record on a core upgrade.

Where to see who else installs an extension, and on which core version. The community maintains the CKAN ecosystem catalog, still in beta: in October 2026 it had about 1.4 thousand extensions and 1.3 thousand portals with core versions. The old extensions.ckan.org address redirects there too. The number of portals in the catalog doubled in autumn 2026, when records from the

dataportals-registry were loaded into it, so it is not a full census but the best available approximation. The most reliable is to ask the portal itself: the `/api/3/action/status_show` address returns the core version and the list of enabled extensions.

Sources

- CKAN 2.12 documentation, release support policy, 2.12.0 changelog, 2.12 release announcement: 2.12 is called the last release of the 2.x line
- CKAN releases on GitHub: 2.12.0, 2.11.6 and 2.10.11 of 26 August 2026
- ckanext-dcat, ckanext-scheming, ckanext-fluent, ckanext-harvest, ckanext-spatial, XLoader, ckanext-pages, ckanext-showcase, ckanext-googleanalytics, ckanext-saml2auth, ckanext-doi: documentation and automated test files, checked 11 October 2026; ckanext-pages's release v0.5.2 of 7 June 2023 and the merge of CKAN 2.12 support on 26 June 2026 checked 14 October 2026
- ckanext-validation and the qld-gov-au fork: compatibility table and automated test files, checked 12 October 2026
- CKAN ecosystem catalog: extensions and portals, checked 12 October 2026; I. Begtin, on the ecosystem catalog, Telegram (in Russian), 5 February 2026, and on loading registry data into it, 9 October 2026
- The `ckan.auth.create_user_via_web` parameter in the 2.9 documentation (on by default) and 2.12 (off); I. Begtin, on the price of openness, Telegram (in Russian), 25 November 2025: spam on open CKAN portals
- The number of portals and the version in the platform entry: a query to the dataportals-registry export, release 1.27.0 of 15 October 2026.

Platform Profiles

A profile is for crossing a platform off before procurement. A mandatory feature must live in the platform itself or in an extension tested on your core version. If the feature is missing, or the extension is not tested on that version, the platform does not go on the shortlist, even if the rest of the table matches.

Then two to four profiles of the same class are compared. Capabilities are covered in Part IV, the full matrix of cells is in the appendix, and checking a module is covered in the chapter on ecosystems. Installation counts come from the dataportals-registry export (release 1.27.0 of 15 October 2026). Software releases were checked against project sites in October 2026. Check the number again before procurement.

How to read a profile

One platform usually has three numbers: the release on the project site, the version in the registry entry and the version on a live site. They diverge. Install the release from the project site and check extensions on exactly that one. A large installation lagging one or two versions behind is normal if an upgrade is planned. Lagging several years behind is a reason to ask who is going to close the gap.

Platform	Release on the project site	Registry entry 1.27.0	Live installation
CKAN	2.12, 26 August 2026	2.11.4	data.gov.au 2.11.6; data.humdata.org 2.11.4; data.gov.gr 2.11.3
Dataverse	6.12.1, 8 October 2026	6.9	Harvard Dataverse 6.10.1
DSpace	10.1, 14 September 2026; 9.4 and 8.5 came out the same day	9.2	DSpace@MIT 8.2
GeoNetwork	4.4.12, 8 July 2026	4.4.9	the Ifremer Sextant catalog showed 4.4.13 on 13 October 2026
NADA	5.7.1, 4 October 2026	5.5	the World Bank library's version was not recorded in this check
udata	17.13.0, 13 October 2026	not given	data.gouv.fr answered as 17.12.1 the same day

A "not given" cell means the version from the registry's software entry was not taken into the text. Take the number for installation from the project site.

A cell in the capabilities table means one of four things.

- **Core**: the feature ships with the platform. Enabling it can still take hours: a DOI in GeoNetwork and Dataverse requires an agreement with DataCite.
- **Extension**: a separate module with its own author and its own table of core versions. Who maintains it is written in the profile.
- **No**: this feature is covered by another platform or by your own program alongside.
- **Not checked**: in October 2026 the documentation gave no answer. Check yourself before choosing.

Each profile has the same sections: class and licence, whom it suits, a capabilities table, ecosystem, launch and maintenance, limitations, examples.

Which profile to start with

The row sets the class. A platform from another row is taken as a second catalog when an organisation has two tasks at once. How to set up such a pair is covered in the chapter on hybrids.

Task	Profiles
An open data portal, with your own team or a contractor	CKAN, DKAN, udata, Piveau
The same portal, with no operations of your own	Huwise and Socrata, ArcGIS Hub
Dozens of datasets, edited as files in Git	JKAN
DOIs, versions and citation	Dataverse, InvenioRDM; DSpace if the library already runs it; without your own installation, Figshare
Map layers and services	GeoNetwork, GeoNode, ArcGIS Hub
Indicators with dimensions	PxWeb; SDMX exchange: .Stat Suite
Surveys with different access levels	NADA; a university archive with DOIs: Dataverse

CKAN

An open source platform for open data portals, AGPL-3.0 licence, the CKAN Association. The current release is 2.12 of 26 August 2026, the last in the 2.x line. It is fully supported. The previous one, 2.11, gets bug and security fixes until the pair of supported versions moves on. The registry lists 1,906 open data portals on CKAN.

Whom it suits. A public authority or organisation with its own team or a contractor, if the catalog must serve DCAT and live for many years.

Whom it does not suit. A catalog that needs DOIs in a setup tested on the current core: Dataverse or InvenioRDM cover that. An organisation with nobody to install updates.

Capability	Where
Search API, files, roles, table API	in the core
DCAT and schema.org	the ckanext-dcat extension; support declared for 2.10 and 2.11, tests also run on 2.12
Custom fields	the ckanext-scheming extension; tests on 2.10, 2.11 and 2.12
Harvesting other catalogs	the ckanext-harvest extension; tests on 2.10–2.12, the latest release 1.6.2 is older than 2.12
Cloud file storage	the 2.12 distribution has only disk; cloud via ckanext-file-keeper-cloud, tests on 2.12, no releases
DOI	the ckanext-doi extension; not tested on 2.11 and 2.12
Citation	extensions: ckanext-doi shows a citation only with a DOI, ckanext-citeproc formats it with CSL styles (2.10 and 2.11 declared)
SAML single sign-on	the ckanext-saml2auth extension; not tested on 2.12

Capability	Where
Sitemap	not in the core

Ecosystem. Large, and a significant part is maintained by the CKAN community itself. A mature setup is the core, ckanext-dcat and, if the standard fields are not enough, ckanext-scheming. A regional harvester adds ckanext-harvest. In October 2026 these packages are tested on 2.12, but the ckanext-dcat documentation does not yet declare 2.12 support. DOIs are not part of such a setup. The list of packages, who maintains them and which versions the tests run on is in the chapter "CKAN: Core and Extensions".

Launch and maintenance. An official Docker setup: days or weeks to a pilot, weeks or months if custom fields and harvesting are needed. Every extension is checked at a core upgrade. The move to 2.12 upgrades the database and the way files are stored.

Limitations.

- Without ckanext-dcat the portal does not get into the national catalog or dataset search engines: there is no DCAT export in the core.
- This platform does not cover a mandatory DOI on the current core.
- A private dataset is hidden entirely. The "description open, file on request" model is assembled with a second system here.
- Registration from the site has been off by default since version 2.10.4. A portal upgraded from an old version often kept the old setting: robots create accounts and publish spam.

Examples. open.canada.ca (Canada), Humanitarian Data Exchange (CKAN 2.11.4), data.gov.au (Australia, CKAN 2.11.6), data.gov.gr (Greece, CKAN 2.11.3). In the October check the Australian portal showed ckanext-scheming, ckanext-dcat, ckanext-harvest, ckanext-spatial and xloader. On 13 October 2026 a repeat `status_show` request from outside returned code 403. The same day the Greek portal returned 66 plugin names, among them scheming, dcat, harvest, fluent, showcase and high-value dataset validation, as well as the agencies' own harvesters. How to read such a list is covered in the chapter on ecosystems.

Platforms are changed in both directions. The US data.gov ran on CKAN until April 2026, then moved to its own software; Mongolia also left CKAN for its own engine. Australia, on the contrary, returned to CKAN from Magda in 2025, and Greece moved to CKAN from its own build in 2025–2026. The UK's data.gov.uk became the National Data Library with a new front end in 2026, with the former catalog kept as a separate section.

DKAN

An open source platform on Drupal, GPL-2.0 licence. The current module release is 4.1.5 of 7 October 2026, declared for Drupal 10 from 10.2 and for Drupal 11. The registry lists 161 open data portals. The module's page on Drupal.org reported about 164 sites on 8 October 2026: this is a report from Drupal installations themselves, not a registry count.

Whom it suits. An organisation whose site is already on Drupal and whose catalog must be part of it.

Whom it does not suit. An organisation that does not want to maintain Drupal. A catalog that needs DOIs and citation export.

Capability	Where
DCAT, API, datasets	in the DKAN core: a `data.json` file; the core schema is DCAT-US 1.1
Custom fields	in the core: the dataset schema is set by a JSON Schema file; the default file is replaced with your own
DOI and separate citation export	no
schema.org, harvesting other catalogs, single sign-on	not checked; login and permissions are Drupal modules
Site modules, permissions, languages	the Drupal ecosystem

Ecosystem. Two update calendars: a hole in Drupal and a hole in DKAN are closed by different releases. The dataset states in the 2.x branch documentation are draft, published, hidden and archived; by default a dataset is published immediately. Before installing 4.1.5, check that this documentation still refers to the branch you are installing.

Launch and maintenance. In terms of time this is open source software with extensions: weeks or months, not days. DCAT does not need to be added as a separate package, unlike CKAN.

Limitations. A custom schema means replacing a file in the distribution, not a catalog of packages like ckanext-scheming. There are no DOIs and no ready citation for a reference manager. Harvesting other catalogs was not confirmed in the October check: if a portal must harvest agencies, look at CKAN or Piveau first.

Examples. The Medicaid data catalog, a CMS portal. The description file is at `/data.json`.

udata

The open source platform of data.gouv.fr, AGPL-3.0 licence, maintained by the Open Data Team. The current release is 17.13.0 of 13 October 2026. The registry lists 15 open data portals, the main one French: in the check on 12 October 2026 it ran 17.12.1 and held 74,376 datasets.

Whom it suits. A portal where API descriptions, discussions and harvesting of other catalogs live alongside datasets, with a team for a Python application.

Whom it does not suit. A portal that needs page markup for Google Dataset Search without its own customisation. An organisation that wants to pick features from an extension catalog, as with CKAN.

Capability	Where
DCAT	in the core: paginated and as a file per dataset
Dump of the whole catalog	in the core: a nightly task writes CSV
API catalog and harvesting other catalogs	in the core: data.gouv.fr has 1,265 data services and 507 harvest sources
Roles	in the core: administrator, editor, and since version 15.4.0 (March 2026) also partial editor
Draft	in the core: it is a private dataset

Capability	Where
schema.org on the page	no: the HTML has no markup block
Curator review before publication	no
DOI	no

Ecosystem. Customisations are made in your own installation. There is no separate catalog of extensions with outside maintainers, as with CKAN. The platform can flag suspicious datasets and discussions by a word list and pass them to a moderator.

Launch and maintenance. Your own team or a contractor, as with any other open source portal platform: weeks or months to a working catalog. On the day 17.13.0 was released, data.gouv.fr was still responding with the previous number: a large site is upgraded after testing, not on the day of the tag.

Limitations. A catalog harvester will see DCAT. A search engine that reads only page markup will not see a dataset until you add markup yourself. There is no country profile "out of the box" in this profile, like the list of specifications in Piveau: the French portal is a model of how to build things, not a ready national profile for another country.

Piveau

An open source catalog platform from Fraunhofer FOKUS, built around DCAT and harvesting other catalogs. The catalog component piveau-hub-repo is released under Apache-2.0; the current release is 4.4.3 of 24 September 2026. The registry lists 19 portals. data.europa.eu runs on Piveau.

Whom it suits. A national or thematic portal that mainly harvests other catalogs, if the DCAT profile it needs is already in the product.

Whom it does not suit. A region with its own fields that are not in the input form: such a profile is then maintained by a contractor. An organisation without a team for several platform services.

Capability	Where
DCAT in several formats, SPARQL	in the core
Harvesting other catalogs and checking description quality	in the core; this is the platform's main job
Custom fields by ready-made specifications	in the input form core: as of October 2026, DCAT-AP and DCAT-AP.de are documented
schema.org on the dataset page	no: the page on data.europa.eu is drawn by a script, and the HTML has no markup
DOI	no

Ecosystem. The platform is assembled from several services. Version 4.4.3 belongs to the catalog component, not a single number for the whole distribution. A third-party module "like ckanext" is not installed here: your own profile is added as separate work.

Launch and maintenance. An open source platform with your own schema: weeks or months. Your own team does the upgrades. On data.europa.eu a dataset description in Turtle opens at an API address even when the page for people is built by a script.

Limitations. A profile not in the form's list is not put on the shortlist. There is no page markup for Google Dataset Search. Your own extensions on the server, as with CKAN, are not provided for.

Huwise (OpenDataSoft) and Socrata

Commercial cloud open data portals. The registry lists 454 portals recorded as OpenDataSoft and 245 Socrata portals. You cannot install your own module on the server: what is left is what the subscription includes, plus a widget and the API.

On 30 September 2025 OpenDataSoft renamed itself Huwise and now calls the product a data marketplace: an open data portal and a showcase for sharing within the organisation and with partners. The platform and API stayed the same, and portal addresses did not change. The registry records the platform under its old name. From here on the book uses Huwise, with the old name in brackets. Socrata is owned by Tyler Technologies and sold as Tyler Data & Insights.

Whom it suits. A public authority without its own team that needs an API, DCAT and tables in the browser within weeks.

Whom it does not suit. An organisation that may not move data to a provider. A catalog that needs DOIs or an export in a profile the provider does not have.

Capability	Huwise	Socrata
API, DCAT, table storage, roles	in the service core	in the service core; a `data.json` file per DCAT-US
Custom description fields	field templates with translation	fieldsets with requiredness and validation
Table data dictionary	label, description and column type	the same
schema.org	in the core, enabled by the administrator	JSON-LD on the dataset page
Private datasets	the dataset is visible in the catalog, the contents to selected users; access requests enabled in settings	a private dataset is hidden entirely; no access requests
Harvesting other catalogs	harvesters for CKAN, Socrata, ArcGIS, ArcGIS Hub, `data.json`, data.gouv.fr and other Huwise portals; CSW harvesting enabled on request through support	the Catalog Connector takes `data.json` and a public ArcGIS Server; a dataset from `data.json` becomes public after synchronisation
DOI, your own extensions	no	no
Single sign-on	SAML and OpenID Connect	OpenID Connect; SAML through support

Launch and maintenance. Days, as with any cloud service. The provider does the upgrades. Harvesting other catalogs is in the service itself; there is nowhere to install a separate module.

Limitations. Leaving later costs a dump of all datasets and files and an address migration. The provider will not write a national DCAT profile that is not in the subscription as a CKAN-style extension: either it is enabled during the pilot, or another platform is chosen. A change of name or owner is a reason to reread the export clauses of the contract, not a reason to reject the service in advance.

Examples. The provider's showcase, formerly data.opendatasoft.com, is now hub.huwise.com.

JKAN

An open source static catalog on Jekyll, MIT licence. The registry lists 48 open data portals.

Whom it suits. An organisation with dozens to a few thousand datasets, if they are edited as files in Git or through a form that saves the edit to the same repository.

Whom it does not suit. A catalog with roles, moderation, requests for restricted files and a search API.

Capability	Where
Dataset pages, a `data.json` file	in the template core. The file declares the DCAT-US 1.1 schema and fills a short list of fields
Editor form	Decap CMS: an edit becomes a Git commit. On GitHub Pages the instructions ask for a separate authentication server
Search API, DOI, roles, your own storage with permissions	no. Search on the page filters a ready JSON file in the browser
schema.org	the page template does not add markup; it is added by editing the template
Change feed	the repository's feed, if the site is built from Git

Launch and maintenance. Hours or days. No application server is needed. Whoever edits the files maintains it: when they leave, either the next person knows how to build the site or the catalog freezes.

Limitations. The template has no search API and no roles. Other catalogs are harvested by a separate script, as on opendata.scot. What to check in `data.json` and on the page is covered in the chapter on static sites.

Examples. opendata.scot. Other lightweight options are in the part "Publishing Without a Portal".

Dataverse

An open source research data repository, Apache-2.0 licence, a project of Harvard University's IQSS. The current release is 6.12.1 of 8 October 2026. The registry lists 257 repositories. The registry entry gives version 6.9, while Harvard Dataverse ran 6.10.1 in the same check.

Whom it suits. A university or research consortium that needs DOIs, versions, citation and moderation, and has a library or computing centre.

Whom it does not suit. An open data portal with daily updates and DCAT-AP federation: CKAN or Piveau are taken for that task.

Capability	Where
Files, versions, OAI-PMH, citation string, BibTeX and RIS	in the core
DOI	in the core: DataCite, since release 6.12.1 also Crossref, plus EZID, Handle and installation permalinks
schema.org	in the core; since version 6.10 a short Croissant markup is embedded in the page
FAIR Signposting	in the core since version 5.14

Capability	Where
Custom metadata blocks	in the core
File viewers	external tools: a manifest goes into the installation, and the application itself can run on another site
DCAT	not in the core; a third-party exporter, and the documentation calls the exporter mechanism experimental
Cloud storage	in the core: S3, Swift, Globus, several stores
Single sign-on	in the core: SAML, OpenID Connect, ORCID

Ecosystem. The essentials are in the core. Outside are viewers and additional export formats, maintained by the GDCC consortium. Since version 6.1 a short signed URL to the file is recommended for viewers. The documentation does not recommend a long-lived API key after version 6.0: it acts like the user's password. A third-party export format is connected since version 5.14 as a JAR file inside the installation, and it is checked at upgrade.

Launch and maintenance. There are official instructions. The first installation is heavier than a static site. Since version 6.8 containers are considered ready for production; several application servers in containers are not supported. The DataCite agreement is concluded separately from the installation. Maintenance is a job for a library or computing centre, not a one-off grant.

Limitations.

- Being able to issue DOIs and issuing them are different things. In the registry, the mark is set for 172 of 257 installations: the rest have no agency agreement connected or the mark was removed manually.
- The Harvard Dataverse dataset page answers a robot with an AWS WAF check, although the API and OAI-PMH work. Check "does the site open" both with a browser and with an API request.
- The DCAT exporter may need rework at upgrade. For a portal that must be harvested via DCAT-AP, this is a weak route.

Examples. Harvard Dataverse. Limits for free publication are in the chapter on research repositories.

InvenioRDM

An open source repository, MIT licence. The current release is v14 of 7 August 2026. Version v13 is supported until 7 February 2027, and v15 was in beta in October 2026. The project's policy: one major version a year, with fixes for at least a year and another six months after the next one. The registry lists 74 installations, including Zenodo.

Whom it suits. An organisation that needs a repository with DOIs, communities and restricted records, and is ready to upgrade a Python application once a year.

Whom it does not suit. A small group that only needs to deposit files: Zenodo itself exists for that. An open data portal.

Capability	Where
Files, DOI, versions, OAI-PMH, DCAT	in the core
Citation	in the core: CSL styles, BibTeX, CSL-JSON
RIS	no
Additional fields	core configuration, not a third-party plugin catalog
Restricted records and access requests	in the core
Cloud storage	in the core since v13: S3
FAIR Signposting	in the core since v12
Single sign-on	in the core: ORCID, OpenID Connect, SAML
Croissant	no
schema.org	in the core

Ecosystem. The documentation ranks customisations from cheap to expensive: configuration, files in expected places, a ready module, your own package in the installation folder, a core change contributed back to the project, your own copy of someone else's repository. A logo is changed by configuration. A copy of your own for the sake of a logo is only worth it if you are ready to carry every update over by hand. More in the chapter on ecosystems.

Launch and maintenance. There are official instructions; the first installation is heavier than a static site. Next to it is the option of publishing in Zenodo and not setting up your own installation. For a small group it is usually better. A large installation lags behind the release just like Harvard Dataverse: this is normal within the support window and becomes a risk once the window has closed.

Limitations. The metadata documentation is based on DataCite schema 4.3, while the current schema is 4.7 of March 2026. New fields, such as a link to a project, are checked against this documentation before choosing. RIS is not exported from the core. Your own installation makes sense when a library moderates many groups and needs its own DOI prefix.

Examples. Zenodo, CaltechDATA. Zenodo's limits are in the chapter on research repositories.

DSpace

An open source repository, BSD licence. The registry lists 829 research repositories: it is above all a publications repository into which data is also deposited. The current release is 10.1 of 14 September 2026; fixes 9.4 and 8.5 came out the same day. DSpace 10 introduced a "dataset" item type. The registry entry gives version 9.2, and DSpace@MIT ran 8.2.

Whom it suits. A library whose DSpace already holds publications, and data is deposited there too.

Whom it does not suit. A new data repository with no publications: Dataverse and InvenioRDM are closer to data. Up to and including version 10.1, DSpace does not cover citation export, so it is not put on a data repository shortlist.

Capability	Where
Files, Dublin Core, OAI-PMH	in the core
DOI	in the core, enabled in configuration; by default the item identifier is a Handle, and the address is built from it
Custom fields and input forms	core configuration: the metadata schema registry and the `submission-forms.xml` file
A ready citation and export to BibTeX and RIS	no, up to and including 10.1: the core has only meta tags for Google Scholar and Zotero; the "Cite" button is being ported from DSpace-CRIS, and the port is not complete
Importing records from BibTeX, RIS and EndNote	in the core
FAIR Signposting	in the core since version 7.6
schema.org and DCAT	not found in the core
Cloud storage	in the core: S3, several stores
Theme and some integrations	your own code in the `additions` module on top of core classes

Ecosystem. There is no third-party plugin catalog. Your own changes are reapplied to the new version. This is cheaper than writing a repository from scratch and more expensive than enabling a feature in Dataverse.

Launch and maintenance. Official instructions; the first installation is heavier than a static site. For a library that already has DSpace, an upgrade is cheaper than a new installation for data. For a new data repository without an existing DSpace, this effort usually does not pay off.

Limitations. In the registry the DOI mark is set for 218 of 829 installations: the identifier is enabled separately, and many repositories live on Handle alone. Meta tags for Zotero are groundwork, not a "copy citation" button. The "dataset" item type appeared in version 10; an installation on 8.x does not have it until it is upgraded.

Examples. DSpace@MIT.

Figshare

A Digital Science cloud service: the shared figshare.com site and institutional repositories. The registry lists 288 research repositories. You cannot install your own module on the server.

Whom it suits. A university or author who needs a DOI without their own installation and accepts the limits and pricing.

Whom it does not suit. An open data portal. An organisation that may not hand files to a provider. A dataset larger than the free tier limit, if a one-off storage fee is not budgeted.

Capability	Where
Files, DOI, citation string, BibTeX and RIS	in the service core
API	in the service core
Your own extensions	no

Launch and maintenance. Days, as with any cloud service. The provider does the upgrades. On 12 October 2026, 20 GB of personal space and files up to 20 GB were free. Figshare+ charges a one-off fee of 1,225 dollars per 250 GB, with files up to 5 TB. In the registry the DOI mark is set for 266 of 288 installations.

Limitations. You can leave with what the provider gives as an export. Automated harvesting of commercial repositories is limited: the chapter on research repositories names this as a separate difficulty. Check pricing and hosting location in the contract, not from someone else's summary.

GeoNetwork

An open source catalog of geodata descriptions, GPL-2.0 licence. The current release of the 4.4 line is 4.4.12 of 8 July 2026, which closes vulnerabilities in earlier 4.4 versions. The registry lists 762 geoportals. The registry entry gives version 4.4.9.

Whom it suits. A spatial data infrastructure that needs ISO 19115, CSW and harvesting of records from other catalogs.

Whom it does not suit. A portal of tables and registers: CKAN or a cloud service is simpler there. An organisation that needs one "map and catalog" site without a separate layer server: GeoNode is closer.

Capability	Where
ISO 19115, ISO 19139, ISO 19115-3, Dublin Core, the ISO 19110 feature catalogue	in the core
CSW, OAI-PMH, harvesting records (CSW, OAI-PMH, OGC services, STAC collections)	in the core
DCAT, including GeoDCAT-AP	in the core
schema.org and sitemap	in the core: JSON-LD on the record page (Croissant fields for ISO 19115-3 are in the branch of the upcoming 4.4.13 release), a sitemap named in `robots.txt`
DOI	in the core; has to be enabled and connected to DataCite
A ready citation for a record	in the core: HTML, text, RIS, BibTeX; the composition of the string is set in the schema's XSL file
Restricting download while the description is open	in the core: a separate permission for files uploaded into GeoNetwork
Cloud storage	in the distribution: S3, JCloud and CMIS modules
Record versions	no; with approval workflow there is a working copy
OGC API Records	a separate microservice, not a catalog page
Country metadata profiles	schema plugins

Ecosystem. The ISO schemas in the distribution are core. A country profile is a separate plugin: a folder of schemas and transformations, sometimes its own JAR. A bad plugin breaks the whole catalog, as the administrator guide warns. Install the branch named after your GeoNetwork line and check that such a branch exists in the repository. Two copies of the British GEMINI profile are analysed in the chapter on ecosystems.

Launch and maintenance. Official instructions, including Docker. The first installation is heavier than a static site. A map server is almost always needed alongside: GeoNetwork describes a layer, and GeoServer or another server serves it. Country profiles are checked at every upgrade.

Limitations.

- The DOI mark in the registry means "the platform can". In DataCite on 12 October 2026, DOIs led to about 45 GeoNetwork installations, while the mark in the registry was set for 555. Before the requirement "all our layers will have DOIs", look at the DataCite agreement, not at the registry cell.
- The number on a live site and the number worth downloading are checked separately. The Ifremer Sextant catalog showed 4.4.13 on 13 October 2026, newer than the announced release 4.4.12.
- An open data portal with DCAT-AP federation as its main task is not built here: the geocatalog serves DCAT, but not a daily showcase of tables.

Examples. Nationaal Georegister, the national catalog of the Netherlands. The European Environment Agency catalog and EU national catalogs are analysed in the chapter on geoportals.

GeoNode

An open source OSGeo geo platform, GPL version 2 or later. The registry lists 770 geoportals. The book's chapters checked release 5.1.0 (2026): the map and catalog are built together.

Whom it suits. An organisation that publishes files by hand and wants one site: upload a file and get a layer, a record, a map and a download link.

Whom it does not suit. A national infrastructure where the catalog and the map server live separately and descriptions go into CSW with a country profile: that is GeoNetwork. For a small town its own server is usually harmful: there is nobody to update it after the project that paid for it.

Capability	Where
File upload, map, download, API	in the core; the layer is served by GeoServer, records via CSW by pycsw
ISO 19115 and Dublin Core via CSW	in the core, through pycsw
Documents, map stories, dashboards	in the version 5.1 core
DCAT and a `data.json` file	no: not found in the 5.1 code
schema.org on the page	not present in the check of 13 October 2026
DOI	no

Ecosystem. Your data and the showcase are one installation. GeoServer and pycsw are already inside and are upgraded together with GeoNode. A separate catalog of schema plugins, as in GeoNetwork, is not needed for ordinary publishing. A national ISO profile is not "switched on with a checkbox" here.

Launch and maintenance. An open source platform with its own server: weeks or months, then your own team. In the registry, 11% of GeoNode geoportals are not working. A common story: a geoportal was built under a donor programme, the programme ended, and the server was switched off.

Limitations. This platform does not serve descriptions in DCAT. An open data search engine and a national DCAT-AP portal will not harvest its catalog by themselves: a second export or a second catalog is needed. For layers that already sit in a regional infrastructure, your own GeoNode duplicates a server that someone already maintains.

Examples. Kazakhstan's national geoportal map.gov.kz: on 12 October 2026 its API served descriptions of 378 datasets without a login.

ArcGIS Hub

Esri's cloud catalog, a commercial subscription. The registry lists 3,464 geoportals. The sites look like both a geoportal and an open data portal: a layer can be downloaded as a file, and the catalog serves DCAT. The registry rarely records Hub as an open data portal: 19 times. You cannot install your own module on the server.

Whom it suits. An organisation whose maps are already in ArcGIS and which needs to quickly give them a page with downloads.

Whom it does not suit. An organisation that does not want to depend on Esri licences. A catalog that needs DOIs.

Capability	Where
Catalog, files and layers, OGC API Records	in the service core
DCAT	in the core: DCAT-US 1.1 and DCAT-AP 2.1.1 feeds by default; DCAT-US 3.0 and DCAT-AP 3.0.0 feeds are enabled separately
Description standard	one style per organisation: ISO 19139, ISO 19115-3, INSPIRE, FGDC, the North American Profile or Dublin Core+ (the default for organisations created after July 2025)
Layer field descriptions	in the core: description, value type, lists of allowed values
DOI, custom fields beyond the chosen standard, your own extensions	no
schema.org	not found on the page in the October 2026 check
Harvesting other catalogs	not checked

Launch and maintenance. Days, as with any cloud service. The provider does the upgrades. Leaving later costs a dump and an address migration.

Limitations. One metadata style for the whole organisation: a second agency with a different standard needs a different site or a different platform. The DCAT-AP 3 feed does not turn itself on. A dataset that can only be seen on a map and cannot be downloaded does not count as an open data catalog; check this during the pilot.

Examples. Organisations' catalogs are collected at hub.arcgis.com. The District of Columbia portal opendata.dc.gov is an example of a site serving both a map and DCAT.

PxWeb

An open source platform for statistical tables from Statistics Sweden, Apache-2.0 licence. The registry lists 110 indicator catalogs, among them Statbank Denmark. The last release of the old line is 2025.v1 (February 2025). The new generation is the PxWebApi 2 API (at Statistics Sweden since October 2025, release 2.7.0 in September 2026) and the PxWeb 2 interface, release 2.12.0 of 7 October 2026. The interface takes its data from that API.

Whom it suits. A statistical office that publishes indicators with dimensions and, for now, serves them to people and programs in JSON-stat, CSV and PX.

Whom it does not suit. A catalog of files. An office that needs SDMX exchange with structures and code lists: .Stat Suite or SDMX-RI covers that, often alongside PxWeb rather than instead of it.

Capability	Where
API	in the core; PxWebApi 2 accepts queries in the URL and serves JSON-stat 2, CSV, PX, Excel, HTML; Parquet in beta
A list of all tables and a selection of those changed in N days	in the PxWebApi 2 core
SDMX	only an SDMX-ML 2.0 file in the old API; SDMX is not among PxWebApi 2 formats
DCAT, schema.org, DOI	no
SDMX structure registry	no

Launch and maintenance. Open source software without its own field schema: days or weeks to a pilot. New installations should go straight onto PxWebApi 2. The move changes query addresses; Statistics Sweden's converter translates users' old links. More in the chapter on indicator catalogs.

Limitations. The registry found a DCAT export for fewer than ten indicator catalogs on any platform. For tables to reach an open data portal, descriptions are made separately. PxWeb has no full SDMX service, and the new API generation has not closed this gap.

.Stat Suite

An open source platform from the SIS-CC community, led by the OECD. The registry lists 40 indicator catalogs. It is built around SDMX: a Data Explorer (release 27.7.0, July 2026), a data loading tool and an SDMX service. Inside, the service is Eurostat's SDMX-RI under the EUPL licence. An office installing .Stat Suite in effect installs that too.

Whom it suits. A statistical office that needs exchange with Eurostat, the IMF or the OECD and has consultants or its own team that knows SDMX.

Whom it does not suit. An office for which tables for people and JSON-stat are enough for now: PxWeb is faster there. An organisation that wants "an indicator portal in weeks" without structures and code lists.

Capability	Where
SDMX API, structure registry	in the core; structures can also be maintained in Fusion Metadata Registry
Data explorer separate from the service	in the distribution core: they are different modules
DCAT	not checked
DOI	no

Launch and maintenance. Implementation is heavy and is rarely done without SDMX knowledge. Several modules are upgraded, not one container. The model interface is the OECD Data Explorer. FAO, the ILO and the statistical offices of Australia, New Zealand, Italy and Switzerland run on the same platform.

Limitations. It is not a replacement for PxWeb for a simple showcase, and not a catalog of files. The combination "PxWeb for people, .Stat Suite for exchange" is a standard statistical office setup. Gluing it to a NADA microdata catalog in one program is not worth it: the standards and access levels differ. See the statistical office example.

NADA

An open source microdata catalog, MIT licence, the World Bank and the IHSN network. The current release is 5.7.1 of 4 October 2026 (5.6 in July 2026, 5.7 in September). The registry entry gives version 5.5. The registry lists 158 microdata catalogs, 47 of which do not open.

Whom it suits. A statistical office or survey archive that needs access levels, requests and DDI descriptions.

Whom it does not suit. A portal of open tables. A university archive that needs DOIs and delivery of descriptions to the European CESSDA catalog via OAI-PMH: Dataverse is closer there.

Capability	Where
A catalog of studies, search by variables, seven access types, requests, API	in the core
DDI Codebook export, schema.org markup (JSON-LD)	in the core
DOI	a field for a DOI minted separately in DataCite; NADA itself does not register DOIs
Citation export	in the core, through DataCite, if the study has a DOI
OAI-PMH, DCAT	no
Indicators, documents, tables and geodata	in the core since release 5.6; no SDMX service or layer services
Description templates	since release 5.7, World Bank metadata editor templates

Ecosystem. There is little in the way of a separate extension catalog. Administrators have API clients for R and Python (NADAR and PyNADA). Descriptions are written in the metadata editor, and NADA displays them. A replacement for the discontinued Nesstar is looked for here or in Dataverse; Nesstar itself is not chosen for a new catalog.

Launch and maintenance. Days or weeks to a pilot, if access levels do not have to be written anew. The main risk is not installation but maintenance after the project: 47 of 158 catalogs in the registry do not open. These are old statistical office installations, often without HTTPS, not a trace of the withdrawn Nesstar. More in the chapter on microdata catalogs.

Limitations.

- Descriptions reach the European archive federation only through an intermediate export: CESSDA harvests DDI via OAI-PMH, and NADA does not have that protocol.
- DOIs are registered yourself in DataCite and entered in the field. While the field is empty, citation export is silent.
- Release 5.6 can describe an indicator but does not replace an indicator catalog. In the World Bank library all records remain surveys. Indicators and microdata are kept in different catalogs.

Examples. The World Bank Microdata Library, the IHSN central catalog.

Platforms without a separate profile

udata, Piveau, GeoNode, .Stat Suite and Figshare are covered above. Below are platforms for specific niches. Check the licence and timelines in the documentation before deciding, just as for the profiles. Installation counts are from the same registry export unless another source is given.

Portals and aggregators

- **Magda**: an open source (Apache 2.0) Australian aggregator catalog that collects descriptions from many sources into one search. In 2019–2025 it powered search on data.gov.au. Today it runs, for example, Kazakhstan's Ashyq Data. It is heavy to maintain and, by a 2024 estimate, pays off only at tens of thousands of datasets.
- **EntryScape**: a commercial platform from the Swedish company MetaSolutions: a DCAT-AP catalog with publication on your own site and into the national dataportal.se. The registry lists 99 open data portals, almost all in Sweden.

Geo platforms (more on their roles in the chapter on geoportals)

- **pycsw**: an open source (MIT) lightweight catalog of geodata descriptions: CSW, OGC API Records, STAC API, OAI-PMH. Built into GeoNode and connectable to CKAN; 44 standalone installations.
- **Oskari**: an open source (MIT or EUPL) platform for map showcases, coordinated by the National Land Survey of Finland. It displays layers from GeoServer, GeoNetwork and other services. 23 installations, 21 of them in Finland, including the national Paikkatietoikkuna.
- **TerriaJS**: an open source (Apache 2.0) library for map showcases, created at CSIRO. It brings WMS, WFS and ArcGIS layers and datasets from CKAN, Socrata and OpenDataSoft onto one map. 75 installations, for example Digital Earth Africa Map. Australia's National Map on TerriaJS was shut down on 27 June 2025.
- **QGIS Web Client 2 (QWC2)**: a map showcase on top of other services. 159 installations, most in Finland and Germany. It stores almost no data of its own.

- **GeoBlacklight**: open source (Apache 2.0) search over maps and geodata based on Blacklight. Maintained by university libraries; an example is the Big Ten Academic Alliance geoportal. 23 installations.

Science, statistics and microdata

- **ERDDAP**: NOAA's open source scientific data server, in the "protocol server" class: subsetting gridded and tabular datasets in various formats, charts and maps. Widely used in oceanography and meteorology; 175 installations.
- **GBIF IPT**: an open source (Apache 2.0) tool for publishing biodiversity data into the GBIF network, a domain system. As of 12 October 2026, 392 installations were registered in GBIF; the registry lists 11 Russian ones, run by institutes and universities.
- **WEKO3**: an open source (BSD-3) repository from Japan's National Institute of Informatics based on Invenio 3. It powers the JAIRO Cloud service: at the end of fiscal year 2025, 822 organisations used it. The registry lists 161 Japanese repositories on WEKO3. Research repositories have their own chapter.
- **InstDB**: a research data repository from the Computer Network Information Center of the Chinese Academy of Sciences, part of the FairStack suite: review before publication, open and restricted access, DOI and CSTR identifiers. 83 installations, almost all in China, for example the repository of the CAS Science Data Center.
- **OpenSDG**: an open source (MIT) static site for Sustainable Development Goal indicators. 72 installations; users include Armenia, Kazakhstan and Kyrgyzstan. More in the chapter on static sites.
- **PxStat**: an open source (MIT) indicator platform from Ireland's statistical office: tables in JSON-stat, PX, CSV and XLSX, an API, a release calendar. It has hardly been replicated: the registry lists 2 installations, the main one data.cso.ie; the latest release is from February 2025.
- **Redatam**: software from the population division of UN ECLAC (CELADE) for processing and publishing census and survey microdata, developed since 1985. Distributed free of charge; the current generation, RedatamX (since 2024), runs on Windows, Linux and macOS and as an R package. The database format is proprietary, with no published specification. Class: microdata catalog; main region: Latin America; 38 catalogs in the registry. See the chapter on microdata catalogs.
- **OBiBa Mica**: an open source (GPL-3.0) portal for epidemiological studies and consortia: study descriptions, variable dictionaries, data access requests. Closer to a microdata catalog; 27 installations.

When to reopen a profile

A decision record is not rewritten at every tag. It is reopened when the answer to "is the feature still in a tested setup" changes.

- The next minor core version has come out, and the needed extension's tests do not include it. For CKAN the nearest such milestone is 2.11 leaving the pair that still gets fixes.
- A support window has closed. For InvenioRDM v13 this is 7 February 2027.

- A cloud provider changed its name, owner or the contents of its export. This has already happened with OpenDataSoft and with Socrata.
- A live site lags behind the release beyond the support window, and there is nobody to upgrade it.
- A mandatory feature has moved: a national profile appeared that is not in the Piveau form, or an agency demanded DOIs.

The record template is in the chapter "Justifying a Decision".

Sources

- CKAN 2.12 documentation, release support policy, 2.12 release announcement: 2.12 is called the last release of the 2.x line
- The DKAN module on Drupal.org: release 4.1.5 of 7 October 2026; GPL-2.0 licence
- udata changelog: 17.13.0 of 13 October 2026; AGPL-3.0 licence
- piveau-hub-repo licence, Apache-2.0, changelog: release 4.4.3 of 24 September 2026
- GeoNetwork licence, GPL-2.0, release 4.4.12 of 8 July 2026
- GeoNode licence: GPL version 2 or later
- InvenioRDM licence policy, MIT, maintenance policy
- Announcement of OpenDataSoft's rename to Huwise, 30 September 2025; Huwise portal security settings; Huwise harvesters; Socrata's Catalog Connector
- Announcement of the new data.gov catalog: leaving CKAN in April 2026
- Link Digital, moving data.gov.au to CKAN, 2025; GDS, What's changing on data.gov.uk and why, 2026
- Versions and extensions of data.gov.au and data.gov.gr from the `/api/3/action/status_show` response, October 2026. Repeat check on 13 October 2026: data.gov.gr, CKAN 2.11.3 and 66 plugin names; data.humdata.org, CKAN 2.11.4 and 34 names; data.gov.au returned code 403. opendata.gov.mn leaving CKAN: I. Begtin, Telegram (in Russian), 25 August 2026
- Installation counts: the dataportals-registry export, release 1.27.0 of 15 October 2026. The queries are in the appendix.
- Comparison table

PART VI.

Choosing a Catalog

From tasks to a feature set, from features to a platform or a custom build, with a justification for every decision.

1. Decision Route Map
2. Selecting Features
3. From Features to a Platform
4. Justifying a Decision
5. Build, Open Source or SaaS
6. Typical Scenarios
7. Requirements Checklist
8. Worked Examples

Decision Route Map

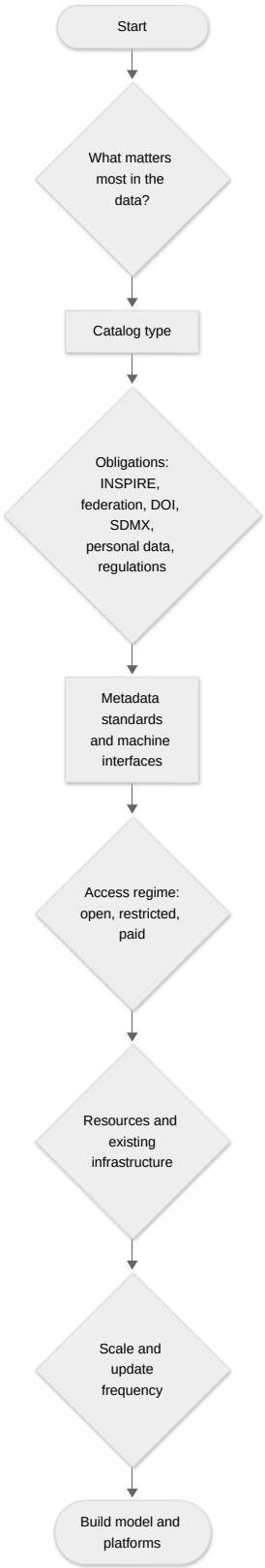
This chapter answers the book's fourth question: which data catalog to choose for its creator's tasks. The choice goes in three stages. First the contents determine the **catalog type**. Then obligations and the access regime set the **standards and interfaces**. Finally, resources, existing infrastructure and scale determine the **build model and platforms**. If the owner is already clear (a ministry, a city, a university, an agency), the chapter "Typical Scenarios" gives the starting row. The route below is for when the type is not yet clear or the scenario row did not fit.

Take the route

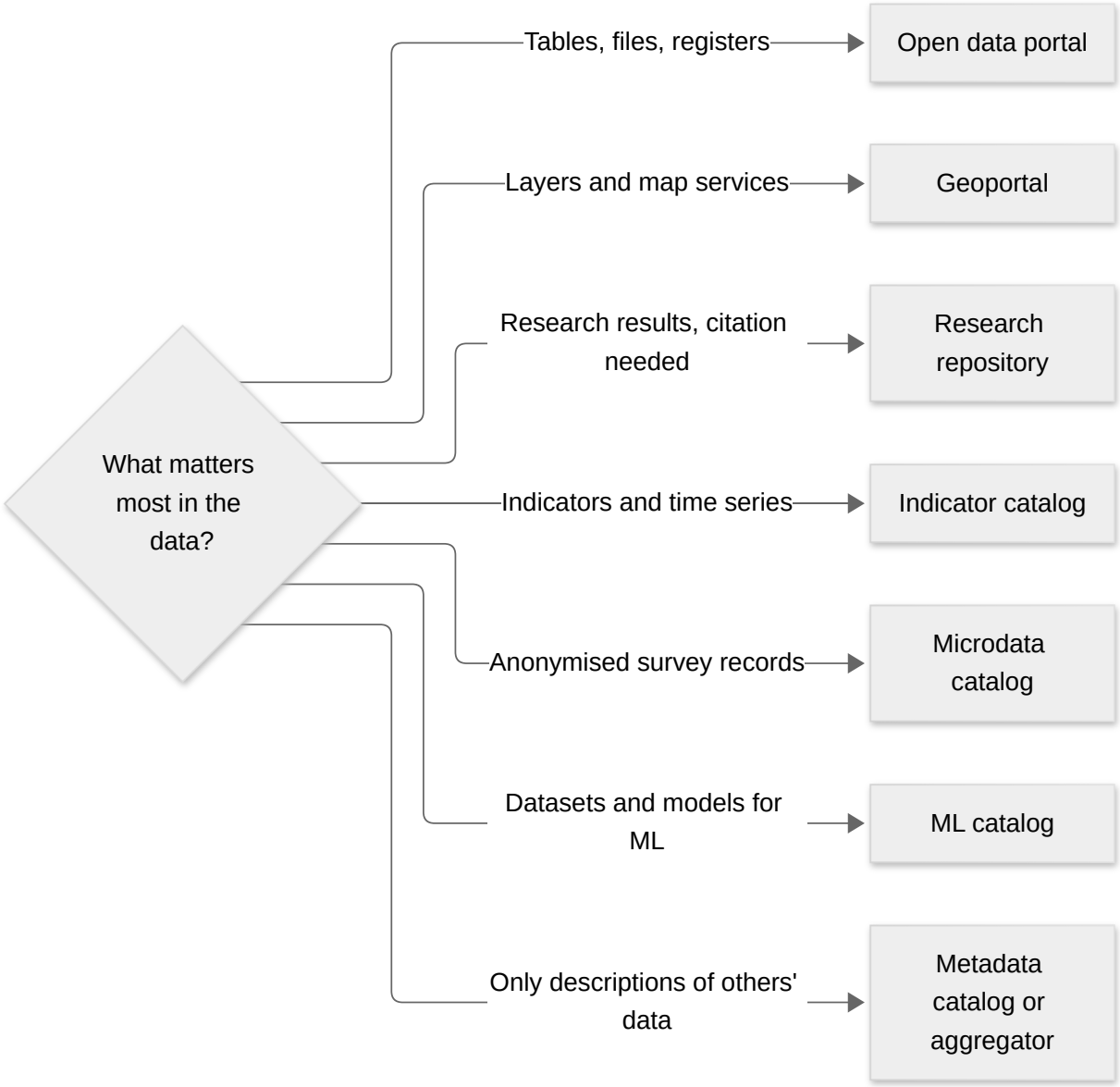
Answer eight questions. The recommendation is a starting point for comparison, not a final choice. Every platform, standard and interface in the answer has a short justification. Marked capabilities come with a note on whether they live in the platform core or in an extension. The detailed method, from tasks to features and from features to a platform, with a justification for each decision, is described in the chapters "Selecting Features", "From Features to a Platform" and "Justifying a Decision".

> The interactive route map works in the online edition of the book: https://datacatalogs.begtin.tech/en/choice/route-map/. The route diagrams and the recommendation logic follow below.

Route diagram



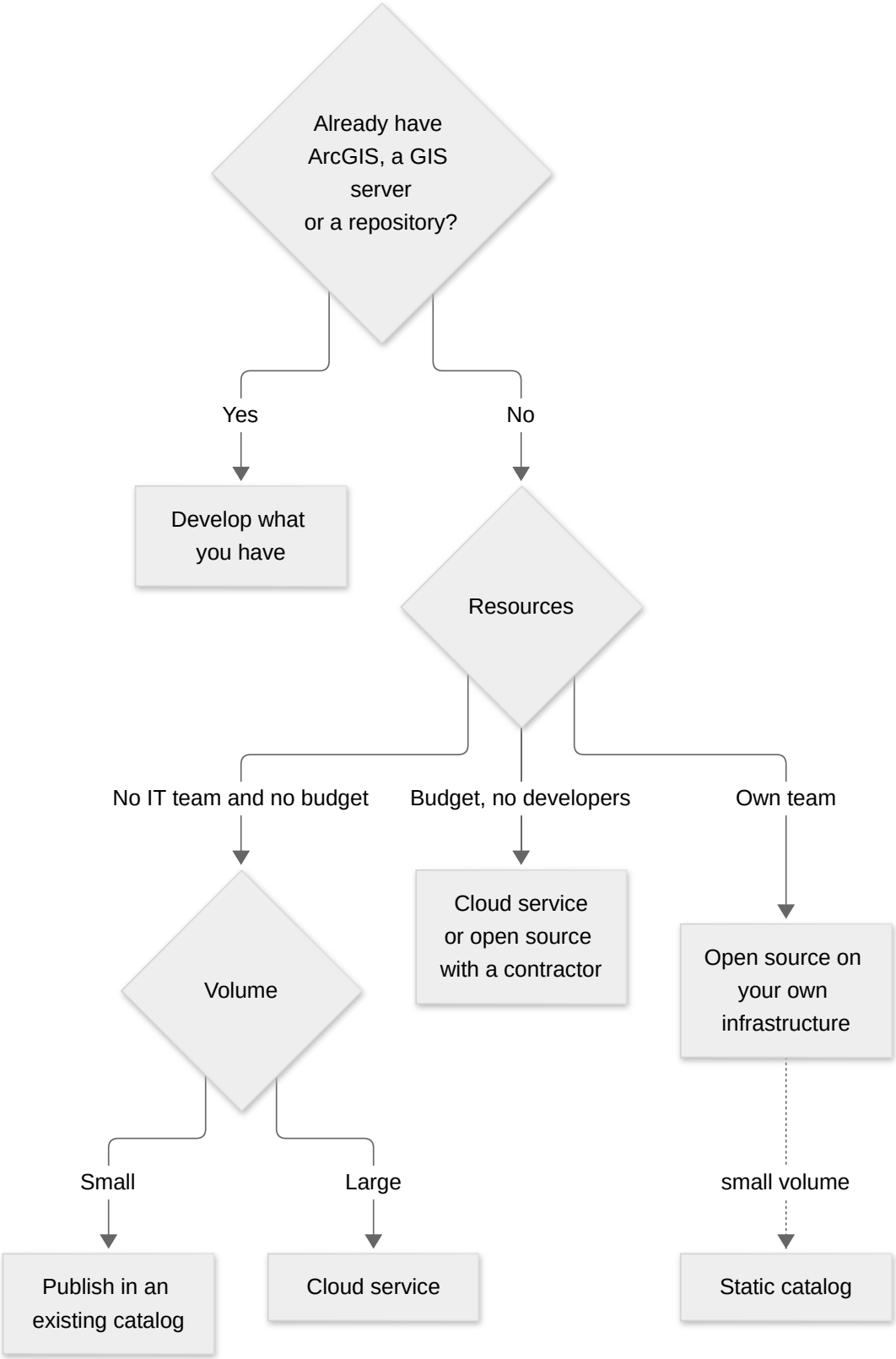
Stage 1. Catalog type by contents



Stage 2. Standards and interfaces

This stage has no diagram of its own: the standards follow from the catalog type and the owner's obligations, such as INSPIRE, harvesting by a higher-level portal, DOI, SDMX or personal data. Which standards and interfaces each type requires is summarised in the capability map, and for typical owners in the scenario table.

Stage 3. Build model



How the logic works

- The recommendation matches the diagrams above.

- One of the answer options is "don't build your own catalog". For small owners without an IT team this is often the best choice.
- Options worth comparing with the recommendation are shown next to it.

Sources

- Selecting Features
- From Features to a Platform
- Justifying a Decision
- Requirements Checklist
- Typical Scenarios

Selecting Features

A catalog's feature list is built from tasks, not from a platform's advertising and not from someone else's specification. Each feature in the final list carries five marks: whose task stands behind it, how mandatory it is, why, how to check it on the live site and when to return to the decision. Such a list can go into a specification and be used to compare platforms.

The route map gives a first bearing in a few minutes, and the capability map shows what exists at all and what is typical for your type of catalog. This chapter is a way to apply both maps to your own case.

In brief:

- **Start with what already exists.** Some tasks are covered by existing sites, services and other people's catalogs. A new catalog is often needed to put order into what is already published, not to publish from scratch.
- **Every feature has a source**: an owner's task, a user's task or an external obligation. A feature without a source is a candidate for deletion, even if "everyone has it".
- **The minimum is mandatory for everyone**; the rest is switched on by the catalog type and specific obligations.
- **"Mandatory" is checked with a question**: which task will fail on launch day without this feature, and who will notice. No answer means "desirable".
- **A feature has a cost**, and the cost depends on where the feature lives: in the platform core, in an extension, in your own customisation or outside the portal.
- **A rejection is recorded just like a choice**, together with the event after which it will be revisited.

Capability, feature, requirement

The book uses three close words, and they are worth telling apart.

- **Capability**: what catalogs can do in general: serve descriptions in DCAT, issue a DOI, restrict a file while the description stays open. Capabilities are the subject of the part "Catalog Capabilities".
- **Feature**: a capability you have chosen for your catalog because someone's task stands behind it.
- **Requirement**: a feature written into a **specification** (the document from which a contractor builds or configures the system), together with its level, justification and verification method.

This chapter turns tasks into features and features into requirements. The international requirements engineering standard ISO/IEC/IEEE 29148:2018 lists the properties of a good requirement. Four matter most for a catalog:

- **necessary**: someone's need stands behind it, and that need can be named;

- **says "what", not "how"**: "the national portal harvests our descriptions in the DCAT-AP profile" is a requirement, "install ckanext-dcat" is already a solution, made at the next step;
- **singular**: one feature per item. "Convenient modern search with an API and a map" is three items, not one;
- **verifiable**: you can show whether it is met or not.

Step 1. Describe what already exists

Before the feature list, make an inventory: what data the organisation already publishes and where. A "Documents" section on the website, tables attached to news items, information system APIs, a map server, pages in other people's catalogs, dumps sent on request. For each item, write down who uses it and what will break if it is moved.

The inventory gives two things. First, some tasks are already solved: layers live on a map server, research data sits in Zenodo, statistics are served through an API. The catalog only has to describe and link them, not repeat them. Second, it shows what must not break: addresses people link to and programs that fetch files on a schedule.

An example is Kazakhstan. In February 2023 the national open data portal data.egov.kz released data only after login with an individual identification number through the state identity service, so it was not really open. Meanwhile, government bodies had more publicly available machine-readable data than the portal did: the open API of the "Taldau" statistics system without authorisation, the programming interface of the gov.kz government site, the geological service's map services on ArcGIS. In this situation the portal's main job is to systematise what is already published, and its features are different: describing and harvesting other people's services, not accepting files by hand. In April 2023 mandatory registration on the portal was abolished, but it began serving no more than 100 records at a time: see the chapter on anti-patterns. The Taldau API documentation is open in October 2026.

If the inventory shows that the tasks are already covered, you may not need your own catalog. When this happens is covered in the chapter "When You Don't Need a Portal".

Step 2. Collect the tasks

A feature has three possible sources:

- **an owner's task**: report under the law, meet the conditions of a grant, sell a service, stop answering the same emails;
- **the task of someone who uses the data**: download a table, cite it in an article, fetch descriptions with a program, build an application;
- **an external obligation**: the national portal harvests descriptions in DCAT, a journal requires a DOI, a spatial data infrastructure requires a catalog of layers.

Where to find tasks:

- **Emails and requests** asking for data. Every request is a user's task; a recurring one is a candidate for mandatory.

- **Statistics of the existing site**: what is downloaded and what is searched for but not found. Searches that returned nothing are a ready-made list of gaps.
- **Conversations with real users** in each group: analysts, researchers, developers. The conversations can stop when the answers start repeating. What different groups expect is in the chapter "Why Catalogs Are Built and Who Uses Them".
- **A list of obligations**: the law and regulations, the higher-level portal's requirements for the description profile, grant conditions, sector infrastructures such as INSPIRE or SDMX exchange. Each obligation is recorded with a link to the document.
- **Editorial tasks**: who will create descriptions, how many, how often they will be updated, where they will come from. This group is forgotten most often, yet roles, bulk upload of descriptions and harvesting from records systems depend on it.

Separately, write down the tasks of programs: the higher-level portal, dataset search engines, harvesters, AI agents. They do not write emails, so their tasks are formulated on their behalf. Usually this is exactly the minimum from the next step.

A task is written in the form "who does what, and why". "A ministry analyst compares air quality by district every month" is a task. "We need a modern portal" is not: it is unclear who will do what on it.

Step 3. Turn a task into a feature

One task usually yields several features, and not always the ones that come to mind first.

Task	Features that follow from it	What is often asked for instead
A journalist downloads budget execution once a year and compares it with the previous year	a CSV or XLSX file, a licence, an update date, previous years at persistent addresses	a preview with charts
The national portal harvests the region's descriptions	a dump in the DCAT-AP profile at a persistent address, a licence on every dataset	a table of datasets emailed once a quarter
An app developer shows hourly air measurements	a file or API at a persistent address, scheduled updates, a description of the fields	a "modern API" without a description of what it serves
A PhD student cites data in a dissertation	a persistent address for the dataset and the file; a DOI and versions only if the data changes and a specific state must be cited	a DOI for every dataset
A grant requires depositing data with an article	a DOI, a version, a citation string, often in a general research repository	a separate repository in every lab
A dataset search engine finds a dataset	schema.org markup, a sitemap, descriptions without registration	promoting the portal on social media
Twenty departments publish on their own	organisations, roles, review before publication	one account for everyone
An agency reports the number of datasets	this is not a user's task, features are not derived from it	splitting one series into hundreds of datasets for the counter

If a ready-made solution arrives instead of a task ("install CKAN", "do it like the neighbours", "we need an API"), ask which task it solves. Behind the word "API" there can be three different features at different costs: a file with all descriptions for the higher-level portal, a selection of table rows for a developer, catalog search for a program. How they work is in the chapter "APIs and Bulk Download".

Step 4. First, what makes the catalog visible

The minimum is the same for everyone, from a national portal to a research group's page:

- a persistent address for the dataset and the file;
- a licence on every dataset, not only on the site;
- an update date and the contact of the person responsible;
- a full metadata dump in one file or request (a small site needs only one `data.json` file);
- schema.org markup and a sitemap;
- descriptions readable without registration or a captcha.

These features are mandatory without separate justification; the reasons are written in the capability map. Faceted search, preview and statistics come after the minimum. A beautiful site without a metadata dump gets into neither the national portal nor a dataset search engine.

The minimum does not require a portal: a static site or a Git repository covers it entirely.

Step 5. Set the levels

Level	Test	Example justification
Mandatory	without the feature a specific task fails on launch day, and it is clear who will notice. An external obligation always gives "mandatory", with a link to the document	"The national portal harvests DCAT-AP, otherwise our datasets will not get into the common showcase"
Desirable	the task is done without the feature, but worse, slower or at greater cost	"A table preview will reduce 'what is in the file' emails, but the file can be downloaded anyway"
Not needed	there is no task, or someone else covers it	"No DOI needed: the data is operational, nobody cites a specific version"

A "not needed" mark does not mean "never". A review condition is written next to it: "No DOI needed while the data is operational; revisit if we start releasing a validated annual archive for researchers." Without a condition, the rejection will have to be explained all over again a year later; with one, it is clear when the explanation is out of date.

If mandatory features are more than half the list, take the test again: most likely some were written down "just in case". Platforms cannot be compared on such a list: they will all turn out "almost suitable", and the cost of the project will grow because of features nobody uses.

Step 6. Estimate the cost of a feature

The same feature costs differently depending on where it lives.

Where the feature is	Launch cost	Maintenance cost	Main risk
In the platform core	configuration	updated together with the platform	the feature is changed or removed in a new version, but the release notes say so
In an extension	installation and configuration	a check at every core upgrade	the extension lags behind the core or is abandoned

Where the feature is	Launch cost	Maintenance cost	Main risk
In a cloud service	switch on in settings or buy a plan	a subscription fee	the feature depends on the plan and the provider's decisions
Your own customisation	developer work	all maintenance is on you	the author leaves, and the feature has no owner
Outside the portal	a contract or working rules	making sure the link does not break	someone else's service changes its terms or closes

The difference is visible on live catalogs. According to the dataportals-registry (release 1.27.0 of 15 October 2026), descriptions in DCAT are served by 239 of 245 portals on Socrata, where the feature is in the core; by 633 of 1,906 portals on CKAN, where the ckanext-dcat extension adds it; and by 24 of 1,144 custom-built open data portals. A core feature is present in almost all installations, an extension feature in a third, a custom-built one in 2%.

An extension without which a mandatory feature does not work is checked before the choice: who maintains it, which core versions the tests run on, when the last release was. The method is in the chapter "Extension Ecosystems". An example of the risk: the ckanext-doi documentation names only CKAN 2.9 and 2.10, and the extension has not been tested on 2.11 and 2.12. If a DOI is mandatory, this is an argument for looking for a platform where it is in the core.

Launch cost and maintenance cost are estimated separately. A cloud service launches in weeks but has to be paid for every year; your own customisation is written once and maintained for the catalog's whole life. More in the chapter "Deployment Speed and Maintenance".

A feature does not have to live in your catalog

Some features are better handed to whoever already does them. The feature then stays on the list, but the "where it lives" column names someone else's service.

Feature	Who can cover it instead of you	What is left to you
DOIs and versions	Zenodo, a university repository, a consortium repository	a link from the dataset page to the DOI and a rule for what counts as a new version
Search for residents of the whole country	a national or regional portal that harvests your descriptions	a metadata dump in its profile
Map layers and map services	a map server or a regional geoportal	a dataset page that points to the service
Change history of small datasets	a Git repository on GitHub or GitLab	tagged releases and a `CITATION.cff` file
Storing large files	cloud storage or a shared data hub	a persistent file address, not a temporary link

The price of such a decision is dependence on someone else's rules: Zenodo's limits, the regional portal's harvesting schedule, the cloud storage pricing. It is written into the decision record just like the price of an extension. If other people's services cover everything except the minimum, you may not need your own portal: the options are analysed in the part "Publishing Without a Portal".

Step 7. Don't close off your next step

Ask what will be needed in three to five years, and check which platform property keeps the road open:

- more publishers: organisations and roles in the core, not one account for everyone;
- a second language: multilingual description fields, not just a translated interface;
- harvesting municipal catalogs: harvesting via DCAT or an API in the core or in a living extension;
- DOIs for research uses: a link to a research repository or to DataCite;
- changing platform: a full dump of descriptions and files, and addresses that can be redirected. See the section "What it costs to leave".

Rejecting a feature today is acceptable. Choosing a platform from which you cannot later approach that feature is not. A cloud service will give you your own fields in a template, but not your own program around them: validation against an external reference list, export in a profile the provider does not have, a new record type. CKAN will grow all this too, but at the cost of extensions.

What comes out

A table, not a paragraph "we want a modern portal". The example below is hypothetical but assembled from typical tasks.

A regional environmental agency publishes hourly measurements from air quality stations, emission permits and the boundaries of protected areas. The layers are already on the regional geoportal. The national portal harvests descriptions in the DCAT-AP profile. Three departments prepare the data, and for now one employee publishes it.

Feature	Where the need comes from	Level	Why	How we will check	When to review
Persistent address for the dataset and file	links from reports, news and apps	mandatory	minimum	the address does not change after the dataset is renamed or its description edited	—
Licence on the dataset	all users	mandatory	without it the data cannot be used legally	twenty random datasets have a licence both on the page and in the dump	—
DCAT-AP dump	the national portal	mandatory	otherwise the agency's datasets will not get into the common showcase	the dump file passes the profile validator, and the national portal has harvested the descriptions	the national portal changes its profile
Air measurements as a file and an API, updated every hour	developers of air quality apps	mandatory	without this, apps scrape other people's pages and make mistakes	data at the address is no older than two hours, the field description is published	—
Protected area layers held locally	already on the regional geoportal	not needed	the layers live on the map server, a copy will drift from the original	the dataset page points to the WFS service and the geoportal page	the geoportal stops updating the layers or closes
Roles for three departments	an editorial task	desirable	at launch one employee publishes at the departments' request	—	departments start publishing on their own

Feature	Where the need comes from	Level	Why	How we will check	When to review
Table preview	"what is in the file" emails	desirable	the file can be downloaded anyway, and the data dictionary answers the same questions	—	the emails do not decrease after the data dictionary is published
DOI	nobody cites a version	not needed	the measurements are operational, nobody cites a specific state	—	the agency starts releasing a validated annual archive for researchers
Harvesting municipal catalogs	—	not needed	this is the job of the regional open data portal, not the agency	—	the region tasks the agency with harvesting municipal environmental data

This table then turns into a platform choice: the next chapter adds "core, extension, customisation, impossible" columns for each platform on the shortlist. The DOI rejection from this table is set out as a decision record. The other forks (platform, extension, custom build) are set out the same way. Four complete chains from task to platform are in the worked examples.

How to check your list

- [ ] Every feature has a recorded source: whose task or which document.
- [ ] The whole minimum is marked mandatory.
- [ ] The list has no solutions instead of tasks: platform names, "like the neighbours", "modern".
- [ ] Each requirement is one feature, not three in one line.
- [ ] Every mandatory feature has a way to be checked on the live site.
- [ ] Mandatory features are fewer than half the list, or it is clear for each why it cannot be done without.
- [ ] Rejections are recorded with a reason and a review condition.
- [ ] It is marked which features are covered by something other than your catalog, and by whom.
- [ ] The list has tasks of programs and of the editorial team, not only of people on the site.

The same items, arranged by specification sections, are in the requirements checklist.

Common mistakes

- **A list from the platform's brochure.** A brochure lists what the platform has, not what you need. A comparison on such a list is won by the platform whose brochure it was.
- **A list from someone else's specification.** A national portal needs harvesting of other catalogs and roles for a hundred agencies; a city with thirty datasets does not. A copied specification carries over other people's tasks and other people's costs.
- **Features for the report.** The number of datasets and views is an owner's metric, not a user's task. When features are chosen for it, the result is split data and a showcase without a dump.

- **The forgotten editorial team.** The portal was chosen for its showcase, but there is nobody to create five hundred descriptions by hand. Bulk upload of descriptions or harvesting from a records system is needed, and that is a feature, not an implementation detail.
- **An "API" without content.** The requirement "there must be an API" is met by any interface, even a useless one. Write what it serves and to whom: all descriptions, table rows, search results.
- **A rejection without a condition.** "No DOI needed" without a review condition looks like an oversight a year later, not a decision.

Sources

- Capability Map: the minimum, features by catalog type, external reasons for being mandatory.
- Why Catalogs Are Built and Who Uses Them: user groups and their expectations.
- Requirements Checklist
- ISO/IEC/IEEE 29148:2018, Systems and software engineering — Life cycle processes — Requirements engineering: properties of a good requirement (necessary, singular, verifiable and others); a requirement describes "what", not "how".
- Shares of catalogs with DCAT by platform: the dataportals-registry export, release 1.27.0 of 15 October 2026. The queries are in the appendix "Using the Registry".
- ckanext-doi compatibility: the chapter "CKAN: Core and Extensions".
- I. Begtin, Kazakhstan's open data portal and publicly available government APIs, Telegram (in Russian), 19 February 2023; abolition of registration and the record limit on data.egov.kz, Telegram (in Russian), 26 April 2023; Taldau API documentation, checked 12 October 2026.

From Features to a Platform

A shortlist of two to four platforms is compared on mandatory features. A platform for which a mandatory feature is impossible drops out before any talk of price or the number of ticks.

By this point the feature list already exists: it is built in the chapter "Selecting Features". Here it is applied to platforms.

Step 1. The approach, then the brand

First not CKAN or Dataverse, but the approach: someone else's catalog, a static site, a cloud service, open source, a custom build. The criteria and the cost of leaving are in the chapter "Build, Open Source or SaaS". If there is little data and no team, a comparison of CKAN with Dataverse is not needed: publish in an existing catalog.

Step 2. The shortlist

Two to four platforms of the class that matched the data type and the approach. Guidance: classes of software and profiles.

What you publish	Usual shortlist
A public authority's datasets, with its own team	CKAN, DKAN, udata
A public authority's datasets, no team	Huwise, Socrata, publishing in a regional or national portal
Research data with DOIs	Dataverse, InvenioRDM, publishing in Zenodo
Map layers and services	GeoNetwork, ArcGIS Hub
Indicators	PxWeb, .Stat Suite, OpenSDG
Survey microdata	NADA, Dataverse
A dozen files, no team	a static site or a Git repository

A platform from the neighbouring row does not get on this list. A research repository does not cover a daily open data portal, and CKAN with an untested DOI extension does not replace Dataverse.

Step 3. The matrix

For each mandatory and desirable feature, put one of four marks:

- **core**: present in the platform itself;
- **extension**: present in a named module, and you have checked who maintains it and which core version it works with;
- **customisation**: the platform has a place to build it in, and a developer estimates the effort;
- **impossible**: the platform cannot do it and cannot be extended to.

A platform with "impossible" on a mandatory feature drops out. The rest are compared on launch and maintenance cost, not on the number of ticks.

"Customisation" and "impossible" belong only to this comparison. The platform summary in the appendix uses different words: core, extension, no, not checked. A "no" cell goes into the matrix as "impossible" only if the feature cannot be added. A "not checked" cell does not go into the matrix: it is closed from the documentation before the platform gets on the shortlist.

An example for a region that needs a DCAT dump for the national portal, harvesting of municipal catalogs and staff login through the organisation's accounts. A DOI is not mandatory for the region.

Feature	CKAN	Huwise	Socrata	Your own site on a generator
DCAT dump	the ckanext-dcat extension, DCAT-AP 3 by default	core; the national profile is not available on every installation and has to be checked	core: a `data.json` file per DCAT-US, not DCAT-AP	template customisation
Search API	core	core	core	impossible without a separate program
Custom fields required by law	the ckanext-scheming extension	core (field templates)	core (fieldsets)	customisation
Harvesting municipal catalogs	the ckanext-harvest extension	core: harvesters for CKAN, Socrata, ArcGIS, `data.json` and other Huwise portals; CSW enabled on request through support	core: the Catalog Connector takes `data.json` and a public ArcGIS Server	impossible
Login through the organisation's accounts	extension (SAML or OpenID Connect)	core: SAML and OpenID Connect	core: OpenID Connect; SAML through support	no
DOI	extension; not tested on CKAN 2.11 and 2.12	no	no	no

A cloud service sets up custom fields in templates without a programmer, but they reach the DCAT dump only if the provider has provided a mapping. In Socrata, fields with names known in advance go into the `data.json` file.

If the national portal expects DCAT-AP, Socrata drops out of this list: its standard file is DCAT-US. Huwise stays until the pilot shows whether the needed profile is there. CKAN stays if ckanext-dcat and the login extension have been checked on the core version that is going to be installed. If a DOI suddenly becomes mandatory, nobody in this four remains: a research repository is taken, not "CKAN plus an unknown extension".

Step 4. Testing on real data

Set up a pilot or open a demo instance and publish five to ten real datasets, including an awkward one: without a licence in the source, with a large file, with daily updates. Then check as described in the chapters on capabilities.

- The DCAT dump opens without a login and passes the validator of the profile the higher-level portal expects.
- The dataset page has schema.org markup, if search engines are supposed to find the dataset.
- The dataset address does not change when the title is corrected.

- The file downloads via a direct link.
- If the matrix says "extension", exactly that extension is installed on the instance, and its version is visible.

A pilot on the provider's demo data does not replace this check. For a cloud service, the pilot also covers the question of leaving: all descriptions and files can be exported in full.

When a custom build is justified

All four statements are true at the same time:

- no platform on the shortlist covers a mandatory feature, even with a living extension;
- the feature does not go away if the task is narrowed (for example, publishing research data in Dataverse and operational data on the portal);
- there is a team not only for the launch but for five years of maintenance;
- your own system will still have persistent addresses, a licence per dataset, a metadata dump and schema.org.

If the fourth is "we'll do it later", the custom build will repeat the most common problem of custom-built catalogs: people use the site, but programs cannot see it. According to the dataportals-registry export (release 1.27.0 of 15 October 2026), among indicator catalogs, where custom builds are most common, only 37% have an API.

Step 5. Record the decision

Every fork in the road (platform, extension, rejecting a feature, custom build) is set out as a decision record. Four complete examples of such a chain are in the chapter "Worked Examples".

Common mistakes

- **Eight platforms in one table.** A shortlist longer than four can no longer be compared: the differences drown. First the class is cut off, then brands are compared.
- **A "not checked" cell rewritten as "no".** In an earlier version of this chapter, Huwise's harvesting was listed as not checked, although the service's documentation has harvesters. An open cell is closed from the documentation, not by crossing out the platform.
- **An extension recorded as a weakness.** ckanext-dcat is an ordinary part of a mature CKAN portal. It becomes a weakness when it has not been tested on the needed core version.
- **A pilot on the provider's data.** Ten demo datasets from someone else do not show your large file, your DCAT profile and your login.

Sources

- Platform Comparison Tables
- CKAN: Core and Extensions
- Huwise harvesters, checked 14 October 2026
- Socrata's Catalog Connector (Tyler Data & Insights)

- Share of indicator catalogs with an API: the dataportals-registry export, release 1.27.0 of 15 October 2026.

Justifying a Decision

A catalog lives longer than the people who chose it. A decision record is one or two pages about one fork in the road: what the surroundings were, which options were really considered, which criteria they were compared on, what was chosen, why the choice beats the rest, what it will cost and which event reopens the choice. From it, the next employee sees which obligation the platform covers.

Without a record they have two blind moves. The first is not to touch the platform, because it is unclear what will break. The second is to replace it and drop an obligation they did not know about: the dump the national portal harvests, or the login staff use to publish.

In brief:

- A record is written for a fork in the road, not for every item of the specification and not for the colour of the theme.
- A requirement says what is needed. A record says which approach was chosen and why the others were rejected.
- In the "why" section, each rejected option loses on a named criterion. "Popular" and "best" are not such criteria.
- Rejecting a feature that someone has already asked for is recorded in the same kind of record as choosing a platform.
- "When to review" is an event the organisation will notice. A calendar date is not such an event.
- An empty "why" cell when the others are filled means the decision has not yet been made.
- A superseded record is not deleted: it is marked "superseded" and links to the new one.

Requirement and record

In the chapter on selecting features, a list of tasks turns into requirements. A good requirement says what is needed and can be checked: "the national portal harvests descriptions in the DCAT-AP profile". A platform or extension name in a requirement is already a decision, made at the next step.

The record is that step. To the same requirement it answers: "we install ckanext-dcat with the DCAT-AP 3 profile and do not write our own export". The full record on the dump is in the chapter on metadata export. Chains from task to platform are in the worked examples.

One requirement can give several records: a platform, an extension, a rejection of a neighbouring feature. One record can close several requirements if one choice covers them. How one record ends often becomes the context of the next: CKAN was chosen, and now there is a separate decision on which extension covers login.

Which forks it is written for

Fork	Why a record	Where it is analysed
Build approach: your own portal, someone else's catalog, a static site, a custom build	there are several approaches, and a mistake is expensive	a municipality, an agency with twenty datasets, a quarterly table, the main address on your own page, an image archive, a catalog run by two people
Platform	more than one on the shortlist	a region and a university
An extension without which a mandatory feature does not work	the feature is not in the core	DCAT dump
Rejecting a feature someone already asked for	otherwise in a year it will be installed as a forgotten one	rejecting DOIs, a city copies a large portal
A core upgrade when an extension is not checked on the new version	publishers' login and security fixes are at stake	core upgrade
Custom build	no ready platform covers a mandatory feature	your own data marketplace
Two catalogs instead of one	two tasks do not live well in one program	a statistical office, a region with maps
Which quality level to fix first	otherwise the editors fill in pages for empty files	"raise quality"
How to publish: a form, a script or a file for a harvester	the catalog lags behind the system where the data already lives	datasets already sit in a system
Which licence to put in the field	your own text and a well-known licence conflict, or a closed file is marked open	your own text or a well-known licence

Small things do not need a record: the theme, colours, button labels. Nor do the items of the minimum that nobody argued against: a persistent address and the very habit of putting a licence on a dataset. A record appears when someone proposes doing without them or replacing a well-known licence with your own text.

A city often needs just one record. A region needs records on the platform, on the set of extensions and on rejections. Twenty records before launch mean that settings got into them.

Template

The technique comes from software development. In 2011 Michael Nygard described a short decision record: context, the decision itself, consequences, including unpleasant ones, and a note if the decision was later superseded. The record is written for a person who comes later, one or two pages long, and kept next to what it is about. A superseded one is not erased.

For a catalog, a comparison and a review condition are added. The comparison is needed because the choice is between products: without rejected options the "why" section has nothing to apply to. The review condition is named in advance, because the trigger usually comes from outside (the country's profile, a grant or a core version changes), and waiting until someone notices in hindsight is too late.

Section	What to write
Context	facts that pull in different directions: a task, a constraint, a system already in place
Options	two to four options that were really considered
Criteria	what they were compared on, and which criterion matters more if they conflict
Decision	what was chosen, in one sentence, with the names of the platform and extensions
Why	which criterion each rejected option lost on
Consequences	what we get and what we pay
When to review	the event after which the record is reopened

In the chapters on capabilities the records are sometimes shorter: the surrounding text has already named the price. A record kept next to a contract includes all seven sections.

What a verifiable section sounds like

The same choice: a dump for the national portal. On the left, a wording against which nothing can be checked. On the right, one that can be checked against a document, a validator or the extension's repository.

Section	What people usually write	What can be checked
Context	We want a modern portal	The region is already on CKAN. The national portal harvests descriptions in DCAT-AP. There is no team of our own, there is a contractor
Options	Consider the best solutions	ckanext-dcat; our own export by the contractor; a manual dump once a quarter
Criteria	Popularity, convenience, price	The file matches the profile the national portal expects. Who fixes the export when the profile changes. Whether CKAN can be upgraded
Decision	A reliable open platform	ckanext-dcat with the DCAT-AP profile
Why	CKAN is popular	The extension is maintained by the CKAN community and already supports this profile. Our own export would become one more program on the contractor's books. A manual dump would fall behind in the first month
Consequences	The portal will get better	At every CKAN upgrade the contractor checks the extension. The CKAN core has no such dump, and this is a standard setup
When to review	In five years or when needed	The national portal changes its profile, or the extension stops being released for a new CKAN version

Popularity sometimes hides a real criterion: "there are contractors in the country who already maintain the current version of CKAN and ckanext-dcat". Then that sentence goes into the record. The word "popular" on its own cannot be checked against anything.

Context is written as facts, without judgement. "The contractor can write an export, and the export will remain their program" is a fact. "Our own export is a bad idea" is a conclusion, and it belongs in the "why" section.

Options are named concretely: the name of an extension, someone else's catalog, a static site with a `data.json` file. "A simpler path" is not an option. A weak move that nobody proposed, inserted to make the winner look obvious, spoils the record: the next reader cannot tell it from a real rejection. If only one option was looked at, write that. The record then shows there was no comparison, and that too is a useful record.

Three criteria are enough. Each can be checked: the file passes the profile validator, the contractor is named, the extension's documentation lists core versions. If the criteria conflict, write which one you will not give up. For a region, matching the country's profile matters more than a pretty showcase.

The decision is one sentence with names. "CKAN with ckanext-dcat" is a decision. "A modern open platform" is a wish.

The "why" section compares. Each rejected option gets a sentence on which criterion it lost. Praise for the chosen option without that sentence repeats the decision in other words. An empty cell when the others are filled means the choice has not yet been made.

Consequences name the price: an extension checked at every upgrade; a second system at the library; the city's dependence on the regional portal; there will be no separate domain with a showcase. A section with only benefits advertises the decision. This is also where you write that the feature is not in the core. Otherwise the next employee will take the extension for something unfinished.

"When to review" is an event that will be noticed without rereading the justification. The national portal changed its profile. The extension was not released for a new core version. The number of datasets exceeded a figure named in advance. A maintenance position appeared. A grant required DOIs. Next to it, in one sentence, what is done then, if it is already clear: "we close our own CKAN and move the addresses". If the move is not yet clear, the record says to reopen the question. A new decision is a new record, not a retroactive edit.

"In five years", "when needed", "when technology changes" and "at the next procurement" are not events: nobody will notice them on the day. A calendar date is appropriate when it stands in someone else's document, and then the record names that document.

Example: rejecting DOIs

This is the record for the "DOI: not needed" row in the requirements table of the environmental agency. It is a model of the record, not an answer for every portal.

Context. The agency publishes hourly air quality measurements and emission permits. The boundaries of protected areas already sit on the regional geoportal. The national portal harvests descriptions in DCAT-AP. A researcher asks for DOIs "like research portals have". There is no agreement with DataCite. The portal is expected to run on CKAN.

Options. Issue a DOI for every dataset with the ckanext-doi extension. Issue a DOI only for a validated annual archive, if one starts being published, through Zenodo or the library. Make do with a persistent address for the dataset and the file.

Criteria. Whether the citation promise matches how the data changes. Whether the extension has been tested on the CKAN version that will be installed. Whether the agency needs an agreement with DataCite.

Decision. Do not issue DOIs. The dataset and the file have a persistent address and an update date.

Why. The measurements change every hour. A DOI promises a fixed state but would open today's data. ckanext-doi has not been tested on CKAN 2.11 and 2.12, as recorded in the chapter on CKAN. The agency does not need a DataCite agreement for operational series. An annual archive for articles is a different publication, in a research repository, and it will get a separate record when the archive appears.

Consequences. There is no DOI button on the portal. The researcher's email is answered with a persistent address and the date of the dump. The reply says plainly that this portal does not issue DOIs.

When to review. The agency starts releasing a validated annual archive that articles cite. Then it is decided where the archive gets its DOI, and it is not quietly switched on on this portal.

Example: a core upgrade

A record for a situation from the chapter on CKAN: a new core version has come out, and the extension without which the portal does not let publishers in has not been checked on it. The facts about the extensions were checked on 13 October 2026. This is a model of the record, not a prescription for every ministry.

Context. A ministry portal on CKAN 2.11. Staff log in with their agency account through ckanext-saml2auth. The extension's latest release is 1.3.0 of 14 December 2022, which declares support for CKAN 2.10. Automated tests run on 2.10 and 2.11, not on 2.12. CKAN 2.12 came out on 26 August 2026, and the project calls it the last release of the 2.x line. Version 2.11 still gets bug and security fixes. When newer releases become the supported pair, 2.11 will lose these fixes. The contractor proposes upgrading.

Options. Stay on 2.11 until ckanext-saml2auth is checked on 2.12. Move to 2.12, checking login on a test instance first. Switch to the ckanext-saml extension: release 0.4.1 of 2 June 2026 includes a fix for the SQLAlchemy changes in CKAN 2.12; the extension has no automated tests.

Criteria. Publishers can log in on the day of the move. The portal is not left without security fixes for an indefinite period. Testing happens on a test instance, not on the production site.

Decision. The contractor sets up a CKAN 2.12 test instance with the current ckanext-saml2auth and checks a staff login. If login works, the production portal is moved. If not, ckanext-saml 0.4.1 is tried on the same instance. While neither option passes login, the production portal stays on 2.11.

Why. A line in a changelog does not replace a staff login: ckanext-saml fixed one incompatibility and shows no tests on 2.12. Staying on 2.11 indefinitely means missing security fixes once 2.11 drops out of the supported pair. Switching extensions on upgrade day without a test instance leaves the portal without publishers.

Consequences. The upgrade work includes a test instance. The record names the address of the repository being installed: ckanext-saml and ckanext-saml2auth have different teams and different evidence. If neither lets anyone in on the test instance, the upgrade is not quietly postponed: a new record on how to replace login is opened at once.

When to review. Login passed on the test instance: this record is marked superseded, and the new one records the move and which extension stayed. Neither extension lets anyone in: a new record on replacing login and on how much longer the portal can live on 2.11. A version came out after which 2.11 no longer gets security fixes: staying on it is not an option, even if the test instance is not ready, and for that period the portal is closed to writing.

If an extension stops being released

This fork is recorded on the day the extension is installed, not on the day it goes silent. The "when to review" section or the consequences name one of three moves:

- stay on the last core version the extension was checked on, and write until which event this is tolerable: usually while that core version gets security fixes;
- replace the extension with another one, named in advance, and test the replacement on a test instance before the move;
- drop the feature, if it was desirable, and record the rejection in a separate record.

The move is chosen by whether the feature is mandatory. A dump harvested by the national portal cannot be dropped: a replacement is sought, or the portal stays on the old core while that is safe. A citation button can be dropped: the persistent address and access date remain. How to evaluate the extension itself is in the chapter on ecosystems.

Where the record lives and who writes it

The record is written by whoever makes the choice, before the contract is signed or before installation. A record written after launch usually defends what has already been done.

It is kept next to the specification, in the same folder. If the portal configuration lives in a repository, the records go there too. One file per decision, numbered in order, numbers never reused. At the top: the date, the status and the author's role, e.g. "accepted, October 2026, portal owner". There are three statuses: proposed, accepted, superseded. A superseded one gives the number of the record that replaced it.

No committee is needed for this. What is needed is one page that a person will open three years later.

How to check your record

- [ ] The record covers one fork.
- [ ] The context has an obligation or task and a constraint: a team, a contractor, a system already in place.
- [ ] There are two to four options, each named concretely, and they are the options that were considered.
- [ ] The criteria can be checked, and it says which matters more if they conflict.
- [ ] The "why" section says which criterion each rejected option lost on.
- [ ] The consequences include a price, not only a benefit.

- [ ] "When to review" is an event. It makes clear what to do next, or says a new record will be opened.
- [ ] Rejecting a feature someone already asked for is recorded separately.
- [ ] For an extension without which a mandatory feature does not work, it says what to do if it stops being released.
- [ ] The record has a date, a number and a status. A superseded one stays in the folder.

Common mistakes

- **One record for the whole procurement.** The platform, the header colour, DOIs and the contractor in one text. A year later one thing cannot be replaced without rewriting everything. Forks are separated.
- **Criteria from a brochure.** "Convenient, modern, scalable" is met by any participant in the comparison. A criterion is tied to your task: the profile the national portal expects, staff login, who fixes the extension.
- **Someone else's record.** The neighbours chose CKAN with three extensions because they harvest a hundred municipalities. For a city with thirty datasets that is someone else's obligation and someone else's price.
- **An option for show.** A deliberately weak move that nobody proposed is added to the list. The real rejected options disappear from the record.
- **Consequences only about benefits.** The price is discovered at the first core upgrade.
- **A date instead of an event.** "Return in 2031" does not say what changed. A calendar date is appropriate as the deadline of someone else's obligation, and then the record names the document.
- **A record after the fact.** It is written to justify a contract already signed. The options are chosen to fit the winner.
- **The old record was deleted.** A new employee sees the current portal and does not see which obligation was dropped in the replacement.

Sources

- Michael Nygard, Documenting Architecture Decisions, 15 November 2011: a short decision record with context, decision and consequences, including unpleasant ones; a superseded record is kept and marked.
- Selecting Features: how a requirement differs from a decision.
- From Features to a Platform, worked examples, build, open source or SaaS.
- Metadata Export: the record on the DCAT dump.
- CKAN: Core and Extensions: core versions and ckanext-doi; access, roles and workflows: login extensions.
- ckanext-dcat.
- ckanext-saml2auth release 1.3.0 of 14 December 2022; ckanext-saml changelog, release 0.4.1 of 2 June 2026. Checked 13 October 2026.

- CKAN release policy; CKAN 2.12 announcement of 26 August 2026: 2.12 is called the last release of the 2.x line.
- Extension Ecosystems: how to evaluate an extension before writing it into a record.

Build, Open Source or SaaS

The build approach is chosen before the brand. A platform that covers the features perfectly is useless if nobody is there to update it.

Six approaches

1. **Publish in an existing catalog**: a regional, national or university one, or a general one such as Zenodo.
2. **A static site or a Git repository**: JKAN, PortalJS, OpenSDG, a static STAC catalog, a repository on GitHub. Details in the part "Publishing Without a Portal".
3. **A section of an existing website**, if the site's engine can serve descriptions to machines, not just draw a table.
4. **A cloud service**: Huwise (formerly OpenDataSoft), Socrata, ArcGIS Hub, Figshare.
5. **An open source platform on your own infrastructure**, with the extensions that cover the mandatory features: CKAN, Dataverse, GeoNetwork, NADA, PxWeb.
6. **A custom build.**

Launch and maintenance timelines are in the chapter on deployment.

What to compare on

- Whether mandatory features are covered by the core or by a living extension.
- How many people and how much money per year, including checking extensions at upgrades.
- Where the data is stored, and whether the organisation's rules allow it.
- What happens if the provider, the contractor or the only administrator disappears.

When each approach fits

Approach	Fits when
Someone else's catalog	there is little data, no team of your own, and a suitable catalog already accepts your datasets
Static site or repository	dozens of datasets, rare edits, someone who can work with Git
A section of a site	the catalog is small and the site can serve DCAT or at least schema.org
Cloud service	a launch in weeks is needed, there is no operations team, the data may be hosted by the provider
Open source platform	there is a team or contractor for years, custom fields and independence from a pricing plan are needed
Custom build	a mandatory feature is not covered by any platform even with an extension, and there is a team for the whole term

What a cloud service costs

Almost nobody publishes cloud portal prices, so reference points have to be gathered from public procurement. What could be found as of October 2026:

Service	What is known about the price	Where it can be seen
PortalJS (Datopian)	the open source version is free; cloud plans are 1,188, 3,588 and 14,990 dollars a year, large projects by arrangement	the vendor's pricing page
ArcGIS Hub (Esri)	the basic version is included in paid ArcGIS Online licences; Premium is a separate annual subscription, priced by the sales team	the purchase options page
Huwise (formerly OpenDataSoft)	no price list. The Grand Paris Seine Ouest territorial authority signed a 600,000-euro contract in 2025 for hosting, maintenance and services; the UK's West of England Combined Authority paid 42,000 pounds for a year of portal operation in 2026–2027	procurement notices
Socrata (Tyler Technologies)	no price list. In 2024 Austin approved a contract for the Data & Insights platform that runs the city portal, for up to five years and up to 5 million dollars	city council materials

Sums from procurement cannot be divided by each other: contracts include different services, terms and numbers of users, and the sum is often a ceiling rather than an actual payment. So the decision record contains not "what the neighbours pay" but a commercial offer for your volume. Where to look for comparable contracts: in the European Union, TED; in France, contract data on data.gouv.fr; in the UK, Find a Tender; in the US, city council agendas and decisions; in Russia, the Unified Procurement Information System.

The market for cloud portals exists mainly where paying for a subscription is customary. In other countries, open source is more often installed, or organisations live on the free version of ArcGIS Hub.

A custom build: four questions before starting

The answers go into a decision record, not into correspondence.

- Which mandatory feature is provided by neither the core nor a living extension? Name the feature, not "we need something special".
- Who maintains the system in five years if the author leaves?
- How will persistent addresses, a dataset licence, a metadata dump and schema.org work in your system? Without them the catalog repeats the typical problem of custom-built sites.
- What does it cost to leave? See the section "What it costs to leave" at the end of the chapter.

According to the dataportals-registry export (release 1.27.0 of 15 October 2026), custom-built catalogs are about 16%. Among indicator catalogs they are 44%, and only 37% of those have an API. A custom build usually loses not on launch day but in the year it stops being developed.

You usually don't need your own engine

The temptation to write the "right" catalog from scratch is understandable: ready-made platforms always have inconveniences. But there are dozens of catalog engines, and they have already solved most tasks. Two examples show the price of the temptation.

The World Bank and the OECD. The World Bank has several separate public data systems: an indicators portal, DataBank, a dataset catalog, a microdata library. In 2025 another portal joined them, Data360, with its own programming interface. As observed in May 2026, it publishes indicator data in SDMX format but does not offer an SDMX-compatible interface. The OECD took another path: its Data Explorer runs on .Stat Suite, an open source platform developed jointly by statistical organisations. The OECD's new showcase is a new version of a shared product, not one more system to be maintained alone.

Moldova's geoportal. The national geodata portal geodata.gov.md showed 16 maps and layers in February 2024 and was built separately from ready-made geo platforms, although GeoNode would have given the same portal out of the box. In October 2026 its interface runs on MapStore on top of GeoServer, the very components GeoNode is made of, only without its catalog and layer pages. The catalog meanwhile is kept separately, on the national spatial data infrastructure geoportal. Assembling your own from ready parts is also a custom build: gluing, upgrading and describing the layers is up to you.

A custom build is justified when a feature is needed that no platform has and without which the catalog loses its purpose, and the answer to the first of the four questions above is named concretely.

A middle path: a ready core and your own interface

Between "take the platform as it is" and "write everything ourselves" there is a stable middle option. Storage for descriptions and files, the programming interface and harvesting from other catalogs are taken ready-made, and the site for people is built separately. This setup is called "headless": the platform's own interface is cut off, and only the core works.

- The Toronto open data portal is a WordPress site, while datasets, their descriptions and the API live in CKAN. In 2023 the city showed the contents and structure of a dataset, a quality score and code examples on the site; the customisations are published as open source. As of 12 October 2026 CKAN holds about 550 datasets.
- The Indian Data Portal of the Bharti Institute of Public Policy is a Next.js site on top of its own CKAN. Through the catalog API the site builds filters that CKAN does not have: granularity, sector, update frequency.
- Canada's national portal open.canada.ca runs on CKAN with a large set of its own extensions, ckanext-canada.

PortalJS is the same idea packaged as a product: a ready interface on top of CKAN or other storage, so you do not have to write your own. More in the chapter on static sites.

The upside is your own interface without losing the standard core: DCAT, harvesting and the API remain the same as on thousands of other portals. The downside is two systems instead of one. Your own site must itself keep persistent dataset addresses and schema.org markup, otherwise search engines will see only the core.

Another option is not to build a portal at all, but to assemble a national search engine over existing catalogs. This is what the Scottish Government did: instead of a common portal it built find.data.gov.scot, which harvests descriptions from dozens of local sources. The analysis is in the

chapter on search engines and aggregators, and how harvesting works is in the chapter on federation.

A contractor instead of a team

A government portal is often built not by its own team but by a contractor under a one-year contract. The risk of this model is that development ends along with the contract, and responsibility for quality gets blurred. Russia's data.gov.ru is a clear example.

- In 2020 the Ministry of Economic Development terminated a 21.8-million-rouble contract to maintain the portal: almost nothing in the specification had been delivered. The contractor was put on the register of unreliable suppliers. This was reported, citing the procurement system, by the Telegram channel "Akh, etot Minfin" in May 2020; the entries in the procurement system could not be checked from abroad in October 2026.
- In March 2023 the portal was closed "for review" to be moved to the "Gostech" platform. Work under the contract ran from August 2023 to December 2024 and cost 30.9 million roubles.
- The new version opened only on 15 July 2025, with 5,826 datasets compared with 26,490 in the last saved snapshot of the old version (February 2022). The old version was closed before the new one was ready, and for more than two years the country had no federal portal.

What follows for the client, whatever the country:

- the code, descriptions and files belong to the client, and a full dump of them is a mandatory deliverable of every stage;
- the old version keeps working until the new one is accepted; if a move is unavoidable, datasets are published as an archive and their addresses are kept;
- work is accepted against working features on the live site, not against a report;
- the portal has at least one person on the client's side who knows how it works and outlasts changes of contractor.

If there is no such person and there will not be, it is more honest to choose a cloud service or a ready platform with minimal customisation: the less of your own code, the less dependence on whoever wrote it.

Example: your own data marketplace

The most common custom-build idea, one that fails not because of the code but because of the economics.

Context. An organisation buys data from several suppliers and wants to build a marketplace where suppliers list datasets and buyers find and pay for them. Ready-made open source catalog platforms do not accept payments.

Options. Your own marketplace. A cloud marketplace (AWS Data Exchange, Snowflake Marketplace), if the suppliers and buyers are already in that cloud. An open catalog of descriptions without payment: the contract and delivery go directly between the parties.

Criteria. Where suppliers and buyers will come from. Why the marketplace is better than a direct contract with the supplier. Who pays for promotion while the marketplace is empty.

Decision. An open catalog of descriptions, and a cloud marketplace for sales if one fits.

Why. Data marketplaces live where buyers already are: at cloud providers, at software vendors for traders, at a government that both promotes the marketplace and may be its regulator. An independent supplier showcase would have had to spend on promotion for years. Owners of valuable data are not looking for an intermediary, and buying data is rare and contract-based. Your own marketplace without a ready audience stays empty.

Consequences. No payment system of your own means no support for it. Descriptions are open, so buyers and search engines see that the data exists.

When to review. Suppliers and buyers have appeared who have agreed to work through you, and there is a budget for promotion.

What it costs to leave

The cost of leaving is calculated before the contract is signed. Otherwise you learn it at the moment you already have to leave.

Add up four items:

1. **The dump.** All descriptions and all files, not a selection. Open source platforms and an honest cloud service have an API or a dump. If there is no dump, that is already the answer.
2. **Addresses.** A table of old and new addresses, and redirects. DOIs issued in advance reduce this item: the `doi.org` link moves together with the DataCite record.
3. **People and downtime.** Who moves things, how many weeks the catalog is read-only, who writes to users.
4. **Retraining publishers.** A new dataset form costs more than it seems from the specification.

A reference for comparison, not an estimate:

Where you are leaving from	What is usually expensive
Someone else's catalog or Zenodo	almost nothing to move, if you did not promise your own addresses
Static site	cheap: the files and pages are already yours
Cloud service	the dump and the change of addresses; read the contract before paying
Open source platform	your own extensions and your own field schema; the files themselves come out easily
Custom build	everything, if the author did not leave a dump and an address map

Lock-in to a cloud service is visible in procurement too. In 2025 Grand Paris Seine Ouest signed a new contract to maintain its OpenDataSoft portal without a tender: the notice says only the vendor holds the rights to the product and its maintenance. This is legal and honestly recorded, but it means there is nobody left to negotiate the price with after implementation.

The closure of someone else's product hits everyone at once. Nesstar, which many microdata archives ran on, stopped being developed in 2015, and each archive moved its collections on its own; Canada's ODESI chose a new platform and started the move before the old one was switched off. The change of ownership of Socrata is the same class of risk for the cloud. A decision record for a cloud service without a paragraph on "how we will get the data out" is incomplete.

Sources

- Worked Examples
- Shares of custom-built catalogs: the dataportals-registry export, release 1.27.0 of 15 October 2026.
- I. Begtin, On data markets and exchanges (in Russian), September 2026: data marketplace models and why they rarely pay off
- Cloud service prices, checked 12 October 2026: PortalJS pricing; ArcGIS Hub purchase options; TED, notice 867108-2025: Grand Paris Seine Ouest and OpenDataSoft, 600,000 euros, without a tender due to exclusive rights; West of England Combined Authority, Open Data Portal Contract, 2026; City of Austin, Legislation Text 24-4611, 2024: Tyler Data & Insights
- I. Begtin, the economics of open data portals, Telegram (in Russian), 7 March 2026: a vendor overview; the prices given there were not confirmed by vendor sites and procurement, except for PortalJS
- The World Bank and the OECD: Data360 and its API, DataBank, the dataset catalog; OECD Data Explorer and the SIS-CC community that develops .Stat Suite; I. Begtin, on Data360, Telegram (in Russian), 15 May 2026
- Moldova: geodata.gov.md, an interface on MapStore and GeoServer, checked 12 October 2026; the national SDI geoportal; I. Begtin, on geodata.gov.md, Telegram (in Russian), 7 February 2024
- A ready core and your own interface: open.toronto.ca and its CKAN customisations; Indian Data Portal; ckanext-canada: dataset counts and extensions from `status_show` and `package_search`, 12 October 2026; I. Begtin, Telegram (in Russian): the Toronto portal, 16 November 2023; Indian Data Portal, 20 September 2024; CKAN as a data management system and PortalJS, 26 May 2023
- data.gov.ru: "Esli byt tochnym" (To Be Precise), The open data portal resumed work after a two-year break (in Russian), 16 July 2025: 30.9 million roubles, contract dates, number of datasets; Kommersant, The Ministry of Economy is preparing to relaunch the data.gov.ru open data portal (in Russian), July 2025; GARANT.RU, The open data portal started work after a major update (in Russian), 15 July 2025; "Akh, etot Minfin", open data portal supplier added to the register of unreliable suppliers (in Russian; repost in the @begtin channel), 8 May 2020, and on the contract to move the portal to Gostech (in Russian), August 2023: 30.9 million roubles against a starting price of 50.7 million, termination of the 2019–2020 and 2020–2022 contracts; I. Begtin, on the contract to move the portal, Telegram (in Russian), 14 August 2023

Typical Scenarios

A scenario answers "where to start", not "what to buy". Each row below names the catalog type, the features without which this owner cannot do their job, and a shortlist of platforms to check. A platform is not carried over from the neighbouring row: a city and a region have different duties and different teams.

In brief:

- First find your row in the table. It sets the type and the mandatory features. The platform is chosen at the next step, from the feature list and the matrix.
- Each row adds to the minimum for any catalog: a persistent address, a licence on every dataset, an update date and a contact, a metadata dump, markup for search engines, descriptions without registration. The row shows only what is added on top of the minimum.
- If an owner does two jobs, they have two rows and often two platforms.
- Four rows are taken all the way to a decision record in the chapter "Worked Examples": a municipality, a region, a university, a statistical office. For your own city, the record's sections are filled in afresh. Someone else's record drags in someone else's obligation and someone else's price.

Figures are from the dataportals-registry export (release 1.27.0 of 15 October 2026).

How to use it

1. Find the closest owner in the table.
2. Write out the "what to add to the minimum" column. These are candidates for mandatory requirements, not a ready specification. The test for being mandatory is in the chapter on selecting features: which task will fail on launch day without this feature, and who will notice.
3. If two rows fit, open the section "Two tasks at once".
4. Check the shortlist from your row against the "feature × platform" matrix. A platform for which a mandatory feature is impossible drops out before any talk of price.
5. Record the fork (your own portal or someone else's, which platform, which extension) as a decision record. Items for the specification are collected in the checklist.

Who	Catalog type	What to add to the minimum	Where next
Central government body	an open data portal or a harvester of other catalogs	the description profile the country expects, and a persistent dump address	below
Region	a portal that harvests cities and itself serves datasets to the country	scheduled intake of other catalogs' descriptions, and its own dump in the national profile	region example
Municipality	more often someone else's catalog, not its own portal	the dataset is visible in the regional or national portal	municipality example
Mapping agency	a geoportal	layer descriptions and a catalog service	geoportals
Statistical office	two catalogs: indicators and microdata	selection by dimensions and SDMX exchange; a survey description and access levels	statistical office example

Who	Catalog type	What to add to the minimum	Where next
University or research organisation	a research repository	DOIs, versions, citation, files held locally	university example
Research consortium	a discipline-specific repository	its own fields, a protocol for discipline aggregators, a long-term preservation plan	research repositories
International organisation	a hub of catalogs of different types	several languages, stable addresses, a dump for aggregators	hybrids
Small organisation, journalists, a group	publishing without a portal	schema.org markup on the dataset page	when you don't need a portal
Company	a dataset portal, an API catalog or a marketplace	a licence a lawyer can understand; the description is open even if the data is paid	other types
A lab or a data programme for models	usually someone else's hub, not its own catalog	a dataset card, a fixed split into parts, a licence for training	ML catalogs

Central government body

Four different jobs are lumped together under this name. Their shortlists differ.

Job	Shortlist
All agencies publish on one site	CKAN, DKAN or udata: organisations, roles, moderation. Harvesting other catalogs is added when some agencies stay on their own sites. This is how data.gouv.fr works: both a platform and a harvester
The site stores almost nothing and harvests other catalogs	Piveau, or CKAN with ckanext-dcat and ckanext-harvest. If the agencies' catalogs already exist, a country can start with a search engine over them: find.data.gov.scot harvests local sources, and the common portal data.gov.scot came later
An agency serves its datasets upwards and does not replace the common portal	CKAN with ckanext-dcat. A cloud service, if its dump file matches the country's profile. Socrata's standard file is DCAT-US. With Huwise, the contractor sees the national profile in the provider's pilot and cannot edit it in a repository
The country hands cities the same distribution	The centre chooses the distribution, not the city: Andino, GD Catalog, a state website platform. Every showcase has common fields and a common dump upwards

Type. An open data portal. For a harvester, it is a catalog of descriptions linking to the original source; it hardly needs files of its own.

Mandatory. A description profile that the recipient already reads, and a stable address from which the descriptions are harvested. In the EU this is DCAT-AP and its national refinements, in the US DCAT-US. In Russia the federal portal and sites on Gosweb expect the "passport" from methodological recommendations 4.0, not DCAT. The passport is kept for the Russian portal. schema.org markup and DCAT are added when someone else must find the datasets or when a harvester names the profile. How to keep these languages apart is in the chapter on metadata standards.

Candidates. From the table above, and only those that have the needed profile in the core or in a living extension. Among custom-built open data portals, the registry found a DCAT dump on 24 of 1,144. People will open the site. The national portal will not harvest the descriptions.

More. Open data portals, metadata export, CKAN and extensions.

When to reopen the record. The country's profile has its own fields that are not in general DCAT-AP. A second shelf has appeared: maps, indicators, microdata. The country is choosing whether to hand cities one distribution.

Region

Type. An open data portal that both harvests municipalities and itself serves descriptions to the country.

Mandatory. Scheduled intake of other catalogs' descriptions and its own dump in the national profile. If the profile has country fields beyond general DCAT-AP, they go both into the form and into the file the national portal harvests. A copy's dataset page links to the dataset page at the city, otherwise one dataset is counted three times: at the city, the region and the country.

Candidates. CKAN: the core, ckanext-dcat and ckanext-harvest. Custom fields are set by ckanext-scheming, and the ckanext-dcat profile puts them into the dump. If the country accepts general DCAT-AP without its own fields, scheming is not installed. Piveau is installed when the portal mainly harvests other catalogs and the country's profile is already in the product: the input form documents DCAT-AP and DCAT-AP.de. A profile not on that list does not go on the shortlist: a contractor would maintain it from then on.

Socrata and Huwise can harvest within the service itself, but on this row they usually drop out on the profile. Socrata's standard file is DCAT-US. In Huwise, custom fields reach DCAT only if the provider has built a mapping. The full comparison and rejected options are in the region example.

More. Federation and harvesting, CKAN and extensions.

When to reopen the record. The record in the examples is written for a DCAT-AP profile with a few custom fields and a contractor without its own development team. A different profile, or an obligation to sit on one state platform, means a new record.

Municipality

Type. Most often not its own portal. The registry has about 17,600 catalogs of local government bodies, about 14,400 of them geoportals. Local bodies have 2,862 open data portals. A noticeable share are showcases of one distribution: 421 sections on the Russian Gosweb website platform, and shared vendor catalogs behind city showcases in Japan and Italy. In that case the city does not choose the platform.

Mandatory. The dataset is visible in the regional or national portal. It has a licence and an update date. The description file matches the profile that this portal harvests. In the JKAN static catalog, the `data.json` file declares the DCAT-US 1.1 schema, and the template fills a short list of fields. A region that expects DCAT-AP will get the descriptions after the template is reworked for its profile.

Candidates. Publishing as an organisation on the regional portal, including in a standard section the region already runs for cities. For maps, the geoportal the region or the country already runs. For a dozen datasets, when the region does not set up other organisations, a page on the city website with markup and a description file.

More. The worked municipality example, when you don't need a portal.

When to reopen the record. In the city example the threshold is: more than two hundred datasets, a position for portal maintenance appears, or the region stops accepting municipal data. Until that day, the city does not install its own CKAN.

Mapping agency

Type. A geoportal. Three different parts hide under this word: a viewer shows layers on a map, a catalog describes them, services serve a layer to programs. A map server without a catalog is not yet a catalog. Of 4,674 geoportals on ArcGIS Server, the registry found a catalog service on one; of 1,927 municipal e-mapa portals, on none.

Mandatory. Layer descriptions per ISO 19115 and a catalog service: CSW or OGC API Records. The layer has a licence. The layer can be downloaded, or the record says plainly that only viewing is available. If the layers are part of the country's spatial data infrastructure, the records are served in its profile. A DCAT dump is needed in addition when the same layer must reach an open data portal.

Candidates. GeoNetwork for the catalog. Services stay on the server that is already in place: GeoServer or ArcGIS Server. ArcGIS Hub, if the organisation already works in ArcGIS and needs both records and file downloads. GeoNode, when one program must both show the map and keep the catalog, and there is no separate team for two systems. For a small town its own server is usually harmful: layers are passed to the regional or national infrastructure.

CKAN is not on this list. The ckanext-spatial extension pulls layer descriptions into the open data portal's search and shows a link to the services. It does not replace the infrastructure catalog.

More. Geoportals, the "region with maps and datasets" record.

When to reopen the record. The agency wants to cover layers and tables with one site, or the country requires both an infrastructure catalog and a DCAT dump at once.

Statistical office

Type. Two catalogs: indicators and microdata. An indicator is selected by year and territory. Microdata are anonymised survey files: the description is open, the file is on request. One custom-built site usually does both jobs badly.

Mandatory. For indicators: selection by dimensions and SDMX exchange with structures and code lists. For microdata: an open survey description in DDI and access levels for files.

Candidates. Microdata: NADA. Indicators: .Stat Suite, if SDMX exchange is the main duty and consultants are available: the SDMX service is in its core. If the team already knows PxWeb, the showcase goes on PxWeb, and an SDMX endpoint is added separately. PxWeb has no full SDMX service: the old API serves an SDMX-ML 2.0 file, and PxWebApi 2 has no SDMX format. Indicators are not put inside NADA. Since release 5.6 the platform can describe them, but in the World Bank Microdata Library all 7,145 records are surveys, and the bank publishes indicators separately. Nesstar is not considered for a new catalog: it was declared obsolete and withdrawn by 2022.

Among indicator catalogs, 44% are custom-built (1,505 showcases). Programmatic access exists for 37% of catalogs of this type and for 8% of custom-built ones. Two complete records are in the statistical office example.

More. Indicator catalogs, microdata catalogs.

When to reopen the record. The records in the examples already separate the shelves and choose the indicator platform. A new one is needed when NADA stops being released or exchange moves away from SDMX. As of October 2026, NADA 5.7.1 came out on 4 October.

University and research organisation

Type. A research repository, not an open data portal with customisations. The administration's open data (the budget, timetables, addresses) is published on the university's page or on a government portal. It does not go into the data repository.

Mandatory. DOIs, versions and citation that can be exported to BibTeX. Files live in the repository, not as a link to the group's site. The library moderates the record. Its own DOI prefix is needed when the university wants the identifier to be its own, not Zenodo's shared prefix.

Candidates. Dataverse, if the library is ready to run a second system next to the existing publications repository: DOIs, versions and citation are in the Dataverse core, a DCAT dump is not. InvenioRDM, if there is someone to maintain a Python application and the university itself serves descriptions in DCAT: the DCAT dump is in its core. An existing DSpace stays as the publications repository. DOIs in it are switched on in configuration and versions exist, but up to and including version 10.1 there is no ready citation and no export to BibTeX and RIS in the core. Zenodo or a shared consortium repository, when the library has nobody to run a second system: DOIs and citation are there, but no own prefix and no group moderation.

CKAN with the ckanext-doi extension is not on this list. The extension has not been tested on CKAN 2.11 and 2.12, and dataset versions are not in the core.

More. The worked university example, identifiers and DOIs, research repositories.

When to reopen the record. The record in the examples is written for a library without a separate team for a new application and without an obligation to serve DCAT at the first step. When that obligation appears, or a DSpace version with citation export in the core comes out, the record is reopened.

Research consortium

Type. A discipline-specific repository with its discipline's schema. An example is PANGAEA for the Earth sciences. NIH advises authors to look first for a repository of their discipline, and only if there is none, a general or university one. A consortium sets up its own when the discipline does not yet have a repository that accepts its format.

Mandatory. Custom description fields: here they matter more than for a city portal. DOIs and versions. A protocol through which the discipline's common aggregators harvest the data, such as OAI-PMH. If the repository will be used as an archive, not as a project folder, a long-term

preservation plan and, in time, certification such as CoreTrustSeal.

Candidates. Dataverse or InvenioRDM with its own metadata schema. A domain system, if the discipline has already agreed on one: GBIF IPT for biodiversity data, other tools of the same discipline for Earth observation and climate. A custom build, when none of them supports the mandatory data format, and there is a team for years of maintenance, not only for the launch.

More. Research repositories, metadata completeness, methodologies and indices.

When to reopen the record. The discipline already runs a shared repository, and the consortium is deciding whether to join it or install its own. Preservation certification becomes a funder's condition.

International organisation

Type. Often a hub: several catalogs of different types under one sign with a common search. The World Bank keeps them separate: indicators, a microdata library on NADA and a common dataset catalog that harvests descriptions from the others. As of 12 October 2026 the dataset catalog had 7,900 records. The Pacific Community assembled the Pacific Data Hub: a dataset portal on CKAN, microdata on NADA, indicators on .Stat Suite, geoportals on GeoNode, a map showcase on TerriaJS, THREDDS servers and a STAC image catalog. A person who has come for one country's data does not need to know which shelf it is on.

Mandatory. Several interface and description languages. Stable addresses: reports will link to the data for years to come. A metadata dump for aggregators, and from each shelf in its own language: DCAT from the dataset portal, SDMX from indicators, DDI from microdata. If the organisation funds development projects, data on them is published to the IATI standard: donors and aid aggregators read it. The hub's common search is built by harvesting descriptions into one catalog. This harvesting is one more chain, and it needs a person behind it. A hub that a person opens in a browser but a program cannot read will not be harvested by an aggregator from outside.

Candidates. For each showcase, a platform of its type: CKAN, DKAN, udata or a cloud service for files, .Stat Suite or PxWeb for indicators, NADA for microdata, GeoNetwork or GeoNode for layers. One engine for all shelves is a common reason for a custom-built system that later nobody can maintain. The working setup is several clear entry points and a common search over them.

More. Comparison of types and hybrids, other capabilities: multiple languages.

When to reopen the record. A hub is set up when there are already three or more catalogs and it matters to people to search across all of them at once. Until then, two addresses on the organisation's site and a caption explaining how the shelves differ are enough.

Small organisation, journalists, research group

Type. No portal is needed. A dozen files, one team and updates a few times a year are covered by a page and a description file.

Mandatory. A persistent address, a licence, schema.org markup, a date and a contact. If a release needs to be cited in an article, the DOI is given to the release in a general repository, not to every file on your own portal.

Candidates. A static site. A Git repository with releases in Zenodo. A shared hub, if the audience is already there. Your own CKAN, Dataverse or a cloud subscription at this size adds maintenance that nobody is there to do.

More. The part "Publishing Without a Portal": static sites, Git repositories, making a publication findable.

When to reopen the record. There are so many datasets that a description file and a search page no longer cope, or the law requires a portal specifically. Thresholds by type are collected in the chapter "When You Don't Need a Portal". For an agency with twenty datasets, the record is already written there.

Company

Type. Depends on what the company opens. Open datasets: a small portal or a page. Programming interfaces: an API catalog for developers. Selling data: a marketplace. These are three different catalogs, and a successful model of one does not carry over to another.

Mandatory. A licence that the buyer's or user's lawyer can understand. The description is open even if the file is paid: otherwise the buyer does not see what they are buying, and a search engine does not find the dataset.

Candidates. A cloud service or a tidy static site for open datasets. A developer portal from an API management system for an API catalog; the interface description is served in OpenAPI. For sales, a cloud marketplace such as AWS Data Exchange, if suppliers and buyers are already in that cloud, or an open catalog of descriptions, with the contract and delivery going directly between the parties. Open source catalog platforms do not accept payments. Your own marketplace without a ready audience usually stays empty: see the "your own marketplace" record.

More. API catalogs, marketplaces and dataset lists.

When to reopen the record. The company wants to publish open datasets, sell others and show developers its APIs. These are three rows, each with its own shortlist.

Lab and data for models

Type. Usually not its own catalog. Datasets for training models are searched for on a few large platforms, and that is where conferences and libraries look too. The registry has 100 ML catalogs. On 12 October 2026 Hugging Face had more than a million datasets. Your own catalog is justified when the data cannot be handed to someone else's platform, or when a government programme tracks datasets from many organisations.

Mandatory. A dataset card: composition, collection method, known biases, what the dataset is not fit for. The split into training and test parts is fixed and does not change between releases. The licence answers whether a commercial model can be trained on the dataset. If the dataset

accompanies a conference paper that requires a Croissant description, the card is served in it. A copy of the release with a DOI goes into a research repository: the platform may close, the link from the article must not.

Candidates. Hugging Face or Kaggle, and a copy of the release in Zenodo or a university repository. A competition with a hidden test part: Codabench or Kaggle competitions. Data that cannot be put on someone else's platform: Dataverse, which has Croissant in the core since 6.10, with the description open and the file on request. An open data portal does not set up a separate ML catalog: tables are served with a data dictionary and, if the platform can, with a Croissant description. A collection of a dozen datasets: a static site with a Croissant file for each dataset.

More. ML catalogs, cloud storage and shared hubs.

When to reopen the record. The programme must accept datasets from many organisations and cannot hand files to someone else's hub. Until that day, no catalog of its own is installed.

Two tasks at once

One owner often recognises themselves in two rows. A portal without a DCAT dump loses open data federation, a geoportal without a catalog service loses the spatial data infrastructure, a repository without DOIs loses citation. These chains hardly overlap, so a platform that is good for one row rarely covers the neighbouring one.

Together	How it is usually arranged	Where it is analysed
A region publishes datasets and layers	an open data portal and a geocatalog side by side; a common search harvests layer descriptions and links to the services	the region-with-maps record
An office runs indicators and microdata	two platforms, two shelves on the site with different captions	statistical office example
A university keeps articles and data	a publications repository and a data repository; the link is kept by identifiers in the descriptions	university example
An organisation runs files, series and surveys	a hub: each shelf has its own platform, with a common search on top	three hybrid setups

A second, small task is sometimes covered by an extension of the first platform: a layer preview on an open data portal, an API description next to datasets in udata. The mandatory standard of the second task is not replaced by an extension. If the spatial data infrastructure expects ISO 19115 and layer services, files in CKAN do not meet that obligation.

If your row is missing

A catalog of descriptions only (reference lists, a registry of repositories, an indicator passport without the numbers themselves) is a metadata catalog. Search over other catalogs without datasets of your own is a search engine or aggregator. If the type is not yet clear, the route map gives it from the question "what matters most in the data", not from a job title.

Sources

- Worked examples, route map, selecting features, from features to a platform, justifying a decision, requirements checklist
- Capability Map: the minimum for any catalog and what a type adds
- The dataportals-registry export, release 1.27.0 of 15 October 2026: about 17,600 catalogs of local government bodies, about 14,400 of them geoportals; 2,862 open data portals of local bodies; 421 Gosweb showcases; a catalog service on 1 of 4,674 geoportals on ArcGIS Server and on none of 1,927 e-mapa portals; DCAT on 24 of 1,144 custom-built open data portals; 1,505 custom-built indicator catalogs, an API for 37% of catalogs of the type and 8% of custom-built ones. The queries are in the appendix "Using the Registry": local bodies, open data portals, geoportals, capability map, indicator catalogs
- Open data portals: national portal models, standard distributions, the passport and Gosweb; metadata standards: the recommendations 4.0 passport and when DCAT is added to it
- Geoportals, indicator catalogs, microdata catalogs: NADA 5.7.1 of 4 October 2026 and the withdrawal of Nesstar, research repositories, ML catalogs
- Comparison of types and hybrids: three setups, the region-with-maps record, the composition of the Pacific Data Hub and the World Bank dataset catalog (7,900 datasets as of 12 October 2026, 7,145 surveys in the microdata library)
- PANGAEA; the World Bank: indicators, microdata library, dataset catalog; Pacific Data Hub; the IATI standard
- find.data.gov.scot: a search engine over existing catalogs; analysis in the chapter on search engines
- I. Begtin, Publicly available data catalogs of international organisations (in Russian), January 2023: hubs of several catalogs

Requirements Checklist

An item without a link to an explanation turns into a line copied from someone else's specification. Below, every item has a chapter that says when you can do without it. Which items are typical for a ministry, a city, a university or a company is in the chapter "Typical Scenarios". The scenario row gives a draft of what is mandatory; in the specification it is checked again.

In a specification each item has three marks (mandatory, desirable, not needed) and one sentence on where the need comes from. A blank "mandatory" on everything makes comparing platforms pointless.

Minimum for any publishing approach

Even for a static site with a dozen datasets.

- [ ] A persistent address for every dataset and file: identifiers. Mandatory for everyone.
- [ ] A licence on every dataset, not only on the site: licences. Mandatory for everyone.
- [ ] A file with all descriptions: metadata export. Mandatory for any catalog of several datasets, including a small site: one `data.json` or `datapackage.json` file is enough. A page with a single dataset does not need it.
- [ ] A sitemap and schema.org markup: metadata export. Mandatory if the dataset must be found by search.
- [ ] Update date and frequency: metadata completeness. Mandatory if the data is updated.
- [ ] The contact of the person responsible. Mandatory for everyone.
- [ ] Descriptions readable without registration or a captcha: APIs and bulk download. Mandatory for everyone who may be harvested automatically.

Include only if there is a task

- [ ] A search API: mandatory when there are hundreds of datasets or more. Not needed for a static site.
- [ ] Your own file storage and a table API: storage. Mandatory for a research repository (files). Desirable for an open data portal. Not needed for a geoportal that links to services.
- [ ] DOIs and versions: identifiers. Mandatory for a research repository. Not needed for operational data.
- [ ] Citation export: citation. Mandatory for a research repository. Desirable if the data is cited in analytics.
- [ ] Custom fields and multiple languages: metadata completeness. Mandatory if the law or the country requires it. Otherwise desirable or not needed.
- [ ] Roles, moderation, restricted access: access and workflows. Roles are mandatory when more than one team publishes. Restricted access is mandatory for microdata and sensitive files.

- [ ] Harvesting other catalogs: other capabilities. Mandatory for a regional and national portal. Not needed for an agency that publishes only its own data.
- [ ] Full-text search over descriptions, preview, statistics and a feedback form: other capabilities. Search is mandatory when there are hundreds of datasets or more. The rest are included by the threshold from the chapter: there is a task and someone to use it.

New capabilities

These are not yet in the core of mainstream platforms. The item is written only if the task is named. What lies behind each is in the chapter on leading catalogs.

- [ ] An API for every tabular file without manual loading, with a column profile and a query description. Desirable for a portal with hundreds of tables that people filter. Not needed if files are downloaded whole.
- [ ] Validating a file against a schema on upload, with rejection on error. Desirable if common schemas already exist for the tables. Mandatory for a domain archive with a common standard, such as BIDS in neuroimaging.
- [ ] A read-only MCP server, without a key for open datasets, on top of the public API, with instructions for the agent: AI and data spaces. Desirable if the platform provides it and licences and the dump are already in place. Do not write your own for the sake of a report.
- [ ] A DOI for a saved query or releases with a stated retention period: citation. Desirable if slices of the data are cited in papers. Not needed for operational data.
- [ ] Persistent identifiers for objects within the data and a mapping table from old numbers: identifiers. Mandatory for a register or a reference list. Not needed for a one-off report.
- [ ] Tiered access: the description and summary statistics are open, the data comes in tiers. Mandatory for sensitive data that researchers need access to. Not needed for open data.
- [ ] Server-side selection and computation. Desirable if the data does not fit in a file. The catalog and the computation are described as separate items: shutting down the computation must not shut down the catalog.

By catalog type

- Geoportal: an ISO 19115 catalog and a CSW or OGC API Records service; the layer can be downloaded, or it says honestly that only viewing is available.
- Research repository: DOIs, versions, OAI-PMH, citation export.
- Indicator catalog: SDMX or other machine access to the table, a description of the methodology.
- Microdata catalog: access levels, a study description, a log of requests.
- ML catalog: a Croissant description, dataset versions with an unchanging split into training and test parts, a licence and data provenance.

When comparing platforms

- [ ] Every mandatory feature is marked: core, which extension and who maintains it, customisation, or impossible: from features to a platform.
- [ ] Every fork (platform, mandatory extension, rejection of a requested feature) has a decision record with the event that reopens it: justifying a decision.
- [ ] It has been counted how many extensions will have to be checked at each upgrade.
- [ ] It is written how to get the data out and keep the addresses if the build approach has to change: the cost of leaving.

After launch

- [ ] Someone checks that links and services open.
- [ ] Open registration is switched off if outside people do not publish, or moderation is switched on: access and roles. Mandatory for portals upgraded from old CKAN versions.
- [ ] There is a closure plan: where the data will go and which addresses will redirect: lifecycle.
- [ ] The catalog can be found from outside: registration in relevant registries and markup, not just a news item on the site.

Sources

- Capability Map
- Worked Examples
- What Leading Catalogs Already Do

Worked Examples

The four records below take a task all the way to a named platform. Which row to start with is in the chapter "Typical Scenarios". Each is written in the seven sections from the chapter "Justifying a Decision": context, options, criteria, decision, why, consequences, when to review. They are models of the record. For another city or office the sections are filled in afresh: someone else's record drags in someone else's obligation and someone else's price.

The numbers of residents and datasets in the examples are hypothetical. Facts about platforms are taken from the profiles and the October 2026 check.

In brief:

- A record covers one fork. A neighbouring rejection that someone already asked for is named separately so that it is not quietly implemented.
- In the "why" section, each rejected option loses on a named criterion.
- The price stands next to the choice.
- "When to review" is an event the organisation will notice. A new decision is a new record.

Example	Fork	Where it ends up
Municipality	its own portal, someone else's catalog or a static site	publish on the regional portal
Region	which platform will harvest the cities and serve the country	CKAN, ckanext-dcat and ckanext-harvest
University	a data repository or an open data portal	Dataverse next to the existing DSpace
Statistical office	one site or two catalogs	indicators and microdata on different platforms

How tasks turn into a feature list and a platform shortlist is in the chapters on features and on the platform.

Municipality: don't set up your own portal

Context. A city of 40,000 residents. About thirty datasets: the budget, routes, school addresses. They are updated twice a year. The administration has one IT person, who has never maintained a portal. The region already accepts municipal datasets and harvests descriptions in the DCAT-AP profile. The city map is on the regional geoportal. Nobody has asked for a geoportal of its own.

Options. Its own CKAN. A cloud portal, Huwise or Socrata. Publishing datasets as an organisation on the regional portal, including in a standard section the region already runs for cities. A static catalog on the city website: JKAN or pages with a description file.

Criteria. The region harvests the descriptions without rewriting the records, and the file matches its profile. One person without portal experience updates the datasets twice a year, and still can in three years. The cost is proportionate to thirty datasets. If the criteria conflict, the first matters more: the datasets must reach the region.

Decision. Publish the datasets as an organisation on the regional portal. If the region does not set up outside organisations, the city puts up a static catalog with a description file in the profile the region harvests.

Why. Its own CKAN loses on the second criterion. Its DCAT dump is not in the core but in the ckanext-dcat extension, and an upgrade means the core plus extensions. With one person without experience, the portal will be left without updates in a year or two. A cloud service loses on the third criterion. It would cover the dataset page, licence, file and dump for the region, but these features already exist on the regional portal. A subscription and a future address migration add a price where there is no installation of one's own. Someone else's contract is not carried over into the record: the sum is taken from an offer for these thirty datasets, as analysed in the chapter on the build approach. A static catalog remains the fallback if the region does not set up outside organisations. In JKAN the `data.json` file declares the DCAT-US 1.1 schema, and the template fills a short list of fields. The region in this record expects DCAT-AP, so the template is reworked for its profile before publication. opendata.scot shows a different case: a static catalog was built when there was no government portal yet. The city in this record already has a higher-level portal.

Consequences. The city depends on the regional portal's fields and schedule. There will be no separate domain with a showcase. The dataset gets its licence and persistent address there too: the region will not invent them for the city. Maps stay on the geoportal. If the dataset needs to be linked from the city website, the page points to the region's record rather than creating a second copy.

When to review. More than two hundred datasets, a position for portal maintenance appears, or the region stops accepting municipal data. Then a new record is opened. CKAN is not quietly installed on the strength of this one.

Region: harvest the municipalities and serve the country

Context. The region publishes its own open data and must serve descriptions to the national portal. The profile is DCAT-AP plus a few mandatory fields of its own. At the same time the region harvests municipal datasets. There is a contractor, but no development team of its own. Some cities already run a catalog, others will publish directly here, as in the municipality example.

Options. CKAN with ckanext-dcat and ckanext-harvest. Piveau. Huwise. Socrata. Its own portal. Not harvesting the cities and leaving them to themselves.

Criteria. The dump file passes the validator of the profile the national portal expects, including its own fields, and when the profile changes it is not fixed by a single provider. A city without a catalog publishes a dataset with a file on the regional portal; a city with a catalog serves a description file, and the region harvests it on a schedule. The contractor upgrades the platform on a test instance. If the criteria conflict, matching the country's profile matters more than the showcase and launch speed.

Decision. CKAN: the core, ckanext-dcat and ckanext-harvest. A city without its own catalog publishes datasets as an organisation on the regional portal. A city with its own catalog serves a description file, and the region harvests it. The national profile's own fields are set by ckanext-scheming, and the ckanext-dcat profile puts them into the dump. On the same day a separate

extension record is written for scheming: what is installed, who maintains it, what to do if the extension goes silent. If the country accepts general DCAT-AP without its own fields, scheming is not installed.

Why. Socrata loses on the first criterion, before any talk of price. It has harvesting: the Catalog Connector takes `data.json` and a public ArcGIS Server. Its standard dump file is DCAT-US, while the country expects DCAT-AP.

Huwise loses on the same criterion. Its harvesting is also in the service itself: harvesters for CKAN, Socrata, ArcGIS, `data.json` and other Huwise portals, with CSW harvesting enabled on request through support. Custom fields are set by templates and reach the DCAT dump only if the provider has built a mapping. The contractor sees the national profile in the provider's pilot and cannot edit it in a repository.

Piveau loses on the first criterion as regards custom fields. It can harvest DCAT and publish datasets: data.europa.eu runs on it, and as of October 2026 its input form documents the DCAT-AP and DCAT-AP.de specifications. This country's profile is not on the list. It is added separately and then maintained by the contractor. In CKAN the general DCAT-AP 3 profile is already in ckanext-dcat, custom fields are set by ckanext-scheming, and the same extension's profile puts them into the dump. Automated tests run on CKAN 2.10, 2.11 and 2.12.

Its own portal loses on the first criterion too. Among custom-built open data portals, the registry found a DCAT dump on 24 of 1,144. People will open the site. The national portal will not harvest the descriptions.

The move "leave the cities to themselves" loses on the second criterion: the country will not get the municipal datasets.

Harvesting and the dump in this setup are two extensions, and they do different work. The outgoing DCAT file and parsing someone else's DCAT come from ckanext-dcat. ckanext-harvest provides the queue, the schedule and the error screen. The latest harvest release, 1.6.2, is older than CKAN 2.12, but the main branch's tests include 2.12. The record notes the address of the repository being installed and a harvesting test on an instance of the chosen core version. Details are in the chapter on CKAN.

Consequences. At every CKAN upgrade the contractor checks ckanext-dcat, ckanext-harvest and, if installed, ckanext-scheming. A person watches harvesting errors. A copy's dataset page links to the dataset page at the city, otherwise one dataset is counted at the city, the region and the country. The CKAN core has neither the dump nor harvesting: this is a standard setup. The same combination already runs on a live portal: in the 13 October 2026 check, data.gov.gr on CKAN 2.11.3 had scheming, dcat and harvest among its plugins.

When to review. The national portal changes its profile, or ckanext-dcat or ckanext-harvest stop being tested on a new CKAN version. Then a replacement extension is sought, or the portal stays on the last core version that still gets security fixes. How to choose between these moves is in the chapter on justification. If the country obliges regions to work on a single state platform, the region's own setup is closed according to an address migration plan.

University: a data repository, not "another CKAN"

Context. The university already runs a publications repository on DSpace. Grants require depositing data with articles, with a DOI, a version and a citation string that can be exported to BibTeX. Research groups publish, and the library is ready to moderate. There is no separate team for a new application: the same library runs the system. There is little open data from the administration, and it belongs on the university's page or on a government portal. DCAT-AP federation with the government portal is not mandatory at the first step.

Options. Extend the existing DSpace. Install Dataverse alongside. Install InvenioRDM alongside. Install nothing and send authors to Zenodo. Install CKAN and add the ckanext-doi extension.

Criteria. DOIs, versions, a citation string and BibTeX export are in the core of the version that will be installed. The library moderates records, and a file can be restricted while the description is open. The university has its own DOI prefix. The existing publications repository is not replaced by this record, and the library maintains the new system without a separate team. If the criteria conflict, citation in the core matters more than the wish to make do with one program.

Decision. Dataverse next to DSpace. The common entry point is on the library website. The dataset description carries the article's DOI, and the article description carries the dataset's DOI. Groups with one-off datasets can still use Zenodo.

Why. CKAN loses on the first criterion. Its DOI is the ckanext-doi extension, which has not been tested on CKAN 2.11 and 2.12. Dataset versions are not in the core. OAI-PMH, which research aggregators use to harvest descriptions, is not in the core either.

DSpace loses on the same criterion, although DOIs in its core are switched on in configuration. Up to and including version 10.1 there is no ready citation and no export to BibTeX and RIS: the core has only meta tags for Google Scholar, and the "Cite" button is being ported from DSpace-CRIS. DSpace 10 introduced a "dataset" item type, but in DSpace repositories datasets remain a small share of records, and the type filter in search is off by default. The library will be able to extend the publications repository into a data repository when citation export appears in the core. Until then, data goes into a separate system.

Zenodo loses on the third criterion. It has DOIs, versions and citation, no installation is needed, and a record can hold up to 50 GB and 100 files. The DOI prefix is shared, and the library does not moderate groups.

InvenioRDM covers DOIs, versions and BibTeX in the core. Its platform profile assigns it to an organisation ready to maintain a Python application. This library has no such readiness, so it does not stay on the shortlist. It suits a university with a separate team and an obligation to serve DCAT itself: InvenioRDM has a DCAT dump in its core, Dataverse does not. In this record federation with the government portal is not mandatory, and a DCAT dump does not tip the decision.

Consequences. The library has two systems. The "article and data" link is kept by identifiers in the descriptions. This record does not promise a common search across them. An agreement with DataCite is needed if there is none yet. A direct member pays 2,000 euros a year plus an infrastructure fee of 200 to 700 euros; a consortium participant pays only the fee. Universities

usually get a prefix through a library consortium. Until there is a prefix, groups use Zenodo. Production DOIs are not issued from a test account: a mistake in the production system is hard to fix, and DOIs are not deleted.

When to review. Nobody is left to maintain two systems: the university's own Dataverse is closed, and an arrangement is made with Zenodo or a shared consortium repository. A DSpace version comes out with a citation string and BibTeX in the core on a par with Dataverse: a new record on merging is opened. An obligation appears to serve descriptions in DCAT-AP itself: the record is reopened and InvenioRDM is looked at again.

Statistical office: indicators and microdata are different catalogs

Context. A national statistical office runs two shelves that people often try to cover with one site. The first is indicators with dimensions of time and territory, and SDMX exchange with international organisations. The second is anonymised survey files: the description is open, the file is on request, and the description follows DDI.

Options. One custom-built "all statistics" site. Two catalogs: an indicator platform and NADA for microdata. Only an indicator platform, with microdata sent by email.

Criteria. A program gets a slice of an indicator by year and territory, and an international partner harvests SDMX. A survey has an open description, access levels and a DDI description. The office does not write both systems itself. Losing either of the first two criteria means the catalog is not ready for work.

Decision. Two catalogs. Microdata go on NADA. The next record chooses the indicator platform. On the office website both entry points stand side by side, and the captions say plainly how the indicator shelf differs from the survey shelf. Indicators are not put inside NADA. Since release 5.6 the platform can describe them, but a capability does not yet make the catalog a hybrid: in the World Bank Microdata Library all 7,145 records are surveys, and the bank publishes indicators on a different platform.

Why. A custom-built site loses on the third criterion and usually on the first. Among indicator catalogs, 44% are custom-built, which is 1,505 showcases according to the dataportals-registry export (release 1.27.0 of 15 October 2026). Programmatic access exists for 37% of catalogs of this type. Among custom-built showcases, an API is marked for 8%. SDMX and access levels would have to be written into such a site, and not abandoned. Email instead of a catalog loses on the second criterion: a study cannot be found or cited while it is not in a catalog. NADA covers access levels and DDI in the core. Dataverse is closer to a university archive: DOIs and OAI-PMH are in its core, but it lacks the seven access types.

Consequences. Two systems to maintain and two addresses. Users get confused if the captions on the site do not distinguish the shelves. A live model of this arrangement is Mexico's INEGI: indicators, microdata on NADA, a geoportal and open data stand side by side and are recorded as different catalogs. NADA is installed on the office's domain, and maintenance, not only installation, is budgeted. Of 59 non-working microdata catalogs, 47 are abandoned NADA installations left after support projects. NADA has no OAI-PMH. It does not register DOIs: there is a field for an already minted DOI, and minting is a separate task. If the office needs exchange with

archives such as CESSDA or its own DOI prefix, that is the next record, not a reason to quietly move to Dataverse. Nesstar is not considered for a new catalog: it was declared obsolete with version 4.0 in 2015 and withdrawn by 2022.

When to review. NADA stops being released: a successor is sought in advance, without waiting until the site is left without updates. As of October 2026 this event has not happened: release 5.7.1 came out on 4 October. Exchange with social science data archives becomes a duty: a record on OAI-PMH and DOIs is opened.

Which indicator platform

Context. The previous record has already separated indicators and microdata. This one chooses only the indicator shelf. SDMX exchange is mandatory. There is no large development team of its own, and the budget for consultants is limited.

Options. PxWeb only. .Stat Suite only. PxWeb for the showcase, with .Stat Suite or SDMX-RI separately as the exchange point.

Criteria. There is an API and selection by dimensions. An international partner receives SDMX with structures and code lists. The office can sustain maintenance of what it installs. If the familiar interface does not cover exchange, exchange matters more than the interface.

Decision. If SDMX consultants are available and exchange is the main duty, indicators go on .Stat Suite: the SDMX service is in its core, using SDMX-RI inside. If the team already knows PxWeb and exchange is needed alongside, the showcase goes on PxWeb. New installations go straight onto PxWebApi 2. The SDMX point is added as a separate product, .Stat Suite or SDMX-RI. Two statistical products are a normal setup here: a showcase for people and a service for exchange.

Why. PxWeb alone loses on the second criterion. It has no full SDMX service: the old API serves an SDMX-ML 2.0 file, and PxWebApi 2 has no SDMX format. The analysis is in the chapter on indicator catalogs. .Stat Suite alone covers the API and SDMX and loses on the third criterion if there are no consultants: without them it is rarely installed. A custom-built showcase on top of this pair is not made. The ECB and Statistics Canada portals sit on top of an API and bulk download. The office in this record does not have such a team.

Consequences. With "PxWeb plus an exchange point", two statistical products are maintained, not counting NADA. Moving to PxWebApi 2 changes query addresses. A converter translates old links, the old API is kept until users have moved, and a switch-off date is announced in advance. In .Stat Suite the data explorer and the SDMX service are different modules, and both are upgraded.

When to review. International exchange moves from SDMX to another mandatory protocol. A team that knew only PxWeb gets SDMX consultants and decides to remove the separate exchange point: that is a new record.

What the examples have in common

A rejection is recorded as explicitly as a choice. The price is named. Review is tied to an event.

Example	What was rejected and on which criterion	What it costs	When to reopen the record
City	its own CKAN: nobody to upgrade it; the cloud: the same features already exist at the region	dependence on the region's fields and schedule; no showcase on its own domain	more than two hundred datasets, a maintenance position, the region stops accepting data
Region	Socrata: a DCAT-US file; Huwise and Piveau: the country's own profile is maintained by the provider or a contractor; its own site: a DCAT dump is rare	extensions checked at upgrades, and a person for harvesting errors	the profile changes, an extension is not tested on a new core version, the country requires one platform
University	CKAN and DSpace alone: no citation in the core; Zenodo: no prefix of its own	two systems and an agreement with DataCite	nobody to run two systems; DSpace gains citation export; its own DCAT-AP is needed
Statistical office	one custom-built site: the API and SDMX would have to be written in-house; email: surveys cannot be found	two addresses for indicators and microdata, and with PxWeb an SDMX exchange point as well	NADA stops being released; exchange moves away from SDMX

Someone else's row of this table does not suit another owner. A region needs a setup that a city has nobody to maintain. A university Dataverse does not become an open data portal for the administration. NADA since release 5.6 does not replace an indicator catalog.

Sources

- Justifying a Decision, selecting features, from features to a platform, build, open source or SaaS: the record template, the region's matrix, and why someone else's contract is not carried over into a record.
- CKAN: Core and Extensions: ckanext-dcat, ckanext-harvest 1.6.2 and tests on CKAN 2.12, ckanext-scheming, ckanext-doi; extensions checked 11 October 2026. The live plugin list of data.gov.gr, 13 October 2026, in the chapter on platform profiles.
- Metadata Export: JKAN's `data.json` file declares the DCAT-US 1.1 schema and fills a short list of fields; Socrata's standard file is DCAT-US; ckanext-dcat's default profile is DCAT-AP 3.
- Huwise harvesters, checked 14 October 2026; Socrata's Catalog Connector.
- The Piveau input form: declared support for DCAT-AP and DCAT-AP.de, checked 14 October 2026. Harvesting as the core's main job is covered in the chapter on federation.
- Research Repositories: DOIs, versions and citation in Dataverse, InvenioRDM and DSpace 10.1; Zenodo limits. DataCite fees are in the chapter on identifiers.
- Indicator catalogs: 1,505 custom-built showcases, an API for 37% of catalogs of the type and 8% of custom-built ones, PxWeb and SDMX. Microdata catalogs: NADA 5.7.1 of 4 October 2026, 47 of 59 non-working catalogs, the withdrawal of Nesstar. Comparison of types and hybrids: INEGI and the 7,145 surveys of the World Bank library.
- opendata.scot: a static catalog built while there was no government portal; analysis in the chapter on publishing without a portal.
- Shares of DCAT among custom-built portals and of APIs among indicator catalogs: the dataportals-registry export, release 1.27.0 of 15 October 2026.

PART VII.

Publishing Data

The publishing process, metadata standards, federation, licences and discoverability.

1. The Publishing Process
2. Metadata Standards
3. Federation and Harvesting
4. Licences and Access Conditions
5. Discoverability

The Publishing Process

A dataset is published when an outsider finds its page, opens the file or sees how to request it, reads the licence and can name who will answer for the rows. A file put up once alongside a report does not achieve this: the page exists, but nobody is there to update it.

The order is the same for a portal, a static site and a repository. What changes is the tool at the "publication" step. The life of the catalog itself (launch, change of platform, closure) is analysed in the chapter on the lifecycle.

In brief:

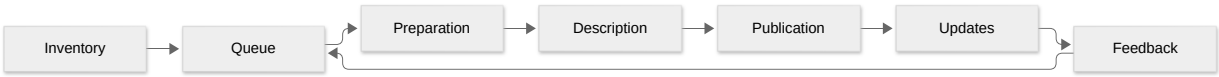
- First write down what the organisation already has. Then decide what to publish from it and in what order.
- Publication means three checks: the page is in search, the file or service opens, the description is in the common dump.
- Each dataset has a named person who will answer an error in the rows.
- A dataset someone may already have cited is not removed silently: the address remains and explains what happened.
- A request for a missing dataset goes back into the queue.

When a dataset can be considered published

What is visible from outside	What must be true
The page is found in search	what reached search is a released dataset, not a draft
The file has a date	the date refers to the rows and is no older than the promised frequency
Somewhere it says "free to use"	the licence is on the dataset as a link and goes into the dump
The site footer has an `info@` address	the dataset names the person and address of whoever will answer on substance
The dataset disappeared from the catalog	the old address opens and explains why the release is gone or what replaced it

The address, the licence, an open file or an honest refusal, the date and the common dump are already in the requirements checklist. Below is the order in which you arrive at that list.

Seven steps



The data owner answers for the rows. The publisher puts up the file and the page. The editor runs the queue and the dump. The full division of roles, and what breaks when a role is empty, is in the table below.

Step	Done when	Who closes the step
Inventory	there is a list of what the organisation already produces, including what must not be published	data owners together with the catalog editor
Queue	each item records why it comes before the others, and the list survives a year	the catalog editor
Preparation	the file opens without paid software, the columns are named, people are anonymised, the licence is agreed	the data owner and a lawyer
Description	the page minimum is filled in, plus the fields without which a dataset in your discipline cannot be understood	the publisher, following the owner's template
Publication	the dataset is in search, the file or service opens, the line is in the common dump	the publisher; with curator review, a second person
Updates	the frequency is recorded, a delay is visible, the address does not change when the file is replaced	the data owner
Feedback	an error report reaches the owner, and a request for a new dataset gets into the queue	the data owner and the editor

Inventory

An inventory is a list of what already exists and who holds it: registers, reports, layers, surveys, dumps from a records system. It is a list of assets, not a plan of "what we'd like to put up" and not the catalog itself. An empty list at this step is more useful than a portal without data: there is nothing to publish yet.

A few columns are enough: the name, where it currently lives, who answers for the rows, whether the data contains people, how often the rows change, whether the law or a grant requires publication, whether anyone has asked for this dataset. For a public authority, a starting point often already exists in a records system. For a university, in a grant's data management plan.

Under OMB memorandum M-25-05 of 15 January 2025, a US federal agency keeps such a list of all data assets except national security systems. A new asset is added no later than 90 days after it is created or found. The whole inventory is reconciled as work goes on and at least once a year. The file is published in JSON at `www.[agency].gov/data.json`, in the DCAT-US 3.0 schema. The inventory itself can be downloaded in full. Not every record has a data file: the decision to release data is made separately, and a closed asset stays in the inventory with a note on how to request it. The deadline to bring the inventory to schema 3.0 and include all assets passed on 30 September 2026; the example below shows what such an inventory looks like two weeks after the deadline.

On 14 October 2026 the General Services Administration (GSA) file at `www.gsa.gov/data.json` opened and redirected to `open.gsa.gov/data.json`. It has 344 records, and the schema is declared as DCAT-US 3.0. All 344 have a contact point filled in: a name and an email. 310 records are public, 33 are not for publication, and one is restricted. The closed record "Concur - Reporting Authorization Model" has no file, and its access conditions say the dataset is not intended for public release and you should write to the publisher. This is how an inventory differs from a download showcase: you can see that the asset exists, and you can see the refusal.

A licence link, meanwhile, is present in 320 of 344 records. In 24 public records the field is empty. The inventory already separates closed from open, but that does not make it a finished publication.

A deadline does not rewrite a file by itself. On 14 October 2026 the US Election Assistance Commission's `eac.gov/data.json` was still declared as DCAT-US 1.1 and contained 11 records. How the national catalog harvests such files is in the chapter "When You Don't Need a Portal".

Queue

First, what the law, a grant or a higher-level portal requires, and what people are already asking for. The rest can wait. The queue is written down, otherwise every year starts from zero.

In the EU, high-value datasets go to the front of the queue: six categories that since 9 June 2024 have been released free of charge, via an API and as bulk download. In the US an agency's open data plan must describe how a member of the public asks to move a specific asset up the queue. Among the reasons, the memorandum names a release already under way, direct benefit to people, frequent freedom of information requests, and data used to check calculations and models. A reference list nobody asked for comes last in this queue.

Preparation

A format that opens without paid software. A data dictionary. Anonymisation, if the data contains people. A licence the lawyer agrees to. How to choose a format is in the chapter on formats, how to choose a licence in the chapter on licences. A form or a script does not replace this step: they publish what a person has already decided.

Description

The minimum from the chapter on metadata completeness: title, publisher, licence, date, persistent address. Plus the fields without which a dataset in your discipline cannot be understood: period and territory, survey variables, a layer's coordinate system. In the Russian dataset "passport" under methodological recommendations 4.0, the responsible person and their contacts are part of the form itself. How the passport works is in the chapter on metadata standards. An international harvester does not read this form. Someone looking for whom to write to still needs the field.

Publication

The dataset is visible in search, the file or service opens, the description is served in the common dump. A draft does not get into search. Who has the right to press "publish" is set by roles. In a research repository this step is often taken by a second person: the author submits the dataset, a curator releases it. How drafts, roles and review work is in the chapter on access.

A page can be put up in three ways.

Method	When it is enough
A form on the portal	dozens of datasets, edited by two people, and these fields are not in the records system
A script from the system where the data already lives	hundreds of datasets, and the page is built from the same fields a person would fill in by hand
A description file harvested by someone else's catalog	there is no portal of your own, or the national catalog itself fetches the file

The data.gouv.fr guide recommends the programming interface when a dataset is updated regularly, and the form when it is published rarely. A script does not assign the licence or the "may be published" flag: a person fills in these fields in the system before launch. The udata platform that runs data.gouv.fr has a separate trap. A dataset created through the API becomes public unless the private flag is passed with every request. A draft is set explicitly in the script.

Updates

The dataset states how often its rows change. The W3C's "Data on the Web" best practices tie this directly to the process: web publication is part of releasing a new version, and a specific person is responsible for the release. The check is short. The frequency is named, and the latest file is no older than the date that frequency promises. A catalog that counts the deadline itself and writes to the dataset owner is the Humanitarian Data Exchange. How the count works is in the chapter on good practices.

In the GSA inventory, the "quarterly" frequency is on 121 records. In the file this is the repeating interval `R/P3M`. For 112 of them the date in the `modified` field is 2020 or earlier. Another 89 records have no frequency at all. Twenty file links point to the host `www.asap.gsa.gov`, which did not open on 14 October 2026. Among them is the dataset "Networx Business Volume FY2013, 3rd Qtr": quarterly frequency and a modification date of 12 June 2019. The inventory itself is being edited: its own record was modified on 5 October 2026. Updating the records has fallen behind updating the inventory file.

A dataset someone may already have cited is not deleted silently. The GSA guide to DCAT-US 3.0 says the same: removal without explanation looks like concealment. The retention period is written in advance, in the description or the rights field. The record keeps a contact so that people can ask why the dataset was removed. If the format is outdated, a simple static file is tried before full removal. Three outcomes, and in each the dataset's address is kept.

- The file is replaced, and the date of the rows changes.
- A new version comes out, and the old one can still be found.
- The release is closed. The old address says why and what to use instead.

Silent deletion is analysed among the anti-patterns. The page that stays in place of a removed dataset with a DOI is in the chapter on identifiers.

A "closed" mark without checking the link is leaky too. 14 of GSA's 33 closed records still give a download link in the description, although the access text says the dataset is not for public release.

Feedback

There needs to be a place to report an error and a place to request a missing dataset. For a public authority's portal this is part of the job. The message reaches the data owner.

The W3C recommendations split this into two steps. The way to write is easy to find on the dataset page. Answers are visible to later readers, so the same error is not sent again. On data.gouv.fr this is the discussion on the dataset page. How the portal serves discussions in the common dump is in the chapter on good practices.

A request "please publish this" without a queue stays in the mailbox. It is written down as the next item in the queue, and the requester is told whether it has been taken on.

Who is responsible for what

Role	Responsible for	Does not	If the role is missing
Data owner	contents, update schedule, answering errors on substance	choose the platform or lay out the page at random	the dataset was put up once, and nobody answers emails
Publisher	puts up the file and fills in the page from the template	assign the licence on the day of upload	the owner edits the portal themselves, and pages drift apart
Catalog editor	pages are structured the same way, the queue does not get stuck, the dump is alive	stand in for the data owner	the inventory starts from zero every year
Lawyer and personal data specialist	the licence and anonymisation before publication	appear for the first time after a complaint	the publisher writes the licence on the day of upload
Platform administrator	the server, software updates, roles, backups	decide which dataset to publish	the catalog opens until the version expires or the disk fills up

In a small group one person can cover two neighbouring roles. They do not cover all five: whoever answers for the rows should not be the only one who can bring up the server. New York City's open data coordinators and the portal's own team are separated in this way. How this is written down is in the chapter on the lifecycle.

What the catalog type changes

The seven steps stay. The type shifts one or two of them.

Type	Which step is different	What changes
Open data portal	inventory and queue	a list of what the agency already keeps; the law and requests come first
Research repository	publication and updates	a second person releases the dataset; editing a file means a new version
Geoportal	preparation	the coordinate system and an ISO description; the common catalog gets a link to that description
Indicator catalog	updates	a series comes out on a calendar, and the methodology sits next to the series
Microdata catalog	preparation and publication	the access level is decided before upload; the description is open even when the file is closed, and it says how to request it
ML catalog	preparation and description	the split into parts and provenance are fixed before publication
Publishing without a portal	publication	the same steps; at the last one, a page and a file

The GSA guide for geodata asks not to rewrite a detailed ISO description inside DCAT. The catalog record links to the existing metadata file and the data dictionary. The threshold at which you need your own portal is in the chapter "When You Don't Need a Portal".

Example: datasets already sit in a system

Context. A regional agency has about two hundred datasets. They already exist in a records system: title, date, department. The national portal harvests descriptions in DCAT. The form on its own portal is filled in by staff who have no time for a second record. The catalog lags behind the system by about a quarter.

Options. Keep filling in the form by hand. A script that creates and updates the page from the records system. A description file on the agency's website harvested by a harvester, like the US `data.json`.

Criteria. The catalog does not lag behind the system by a quarter. The script does not assign the licence or the "may be published" flag on its own. A person answers errors in the rows.

Decision. A script creates and updates pages from the system's fields. A person fills in the licence, the publication flag and the owner's name in the system before launch. The form stays for datasets that are not in the system.

Why. A manual form for two hundred datasets falls behind in the very first quarter. A file for a harvester covers the task when there is no portal of your own or the national catalog fetches the file from the agency's site. Here a portal with departmental roles is already in place, so the script writes into it directly: a harvester does not maintain these roles. A script without licence and owner fields will publish empty pages faster than people would. Such a script loses on the "a person answers" criterion.

Consequences. The records system gains three fields it may not have had: licence, publication flag, owner. A broken script is silent, so it gets the same deadline monitoring as a dataset. A delay is visible to the editor.

When to review. The national portal changes its profile, and the script starts serving a file it will not accept. The records system is replaced. Most new datasets stop passing through the system, and the form becomes the main path again.

How to write your own record is in the chapter "Justifying a Decision".

One dataset: a bus timetable

The inventory finds the timetable in the transport system. In the queue it ranks above an album of bus stop photos: people already ask for it, and the regional portal may be waiting for it. Preparation: a CSV with columns from the dictionary, no merged cells, a licence the lawyer agrees to. Description: title, the operator as publisher, validity period, the licence as a link, the date of the rows, the name of whoever will answer for an error in a route. Publication: the page is in search, the file opens, the line is in the common dump. Updates: when a route changes, the file and the date are changed, and the page address stays. Feedback: a "timetable error" report goes to the operator. The same W3C recommendation walks through bus stops in a similar example: the frequency is written both in words and in a field a program reads.

If there are twenty such datasets and two people edit them, a spreadsheet and a static site are enough. If there are two hundred and twelve agencies edit them, a platform with roles is needed. The threshold is in the chapter "When You Don't Need a Portal".

How to tell the process is alive

This can be checked from outside, without convening the editorial team.

- A dataset picked at random has a name or team and an address on its page, not just the whole portal's mailbox.
- The file date is no older than the promised frequency. An empty frequency means no update has been set for this dataset.
- A draft cannot be found by search.
- The address of a removed dataset opens and explains what became of it.
- There is somewhere to request a missing dataset. On a public authority's portal this request is visible as a queue or a public discussion.
- The inventory itself exists as a document (a table or a file like `data.json`), not only as the editor's memory.

The same list, shorter, is in the checklists.

Sources

- Data on the Web Best Practices, W3C, 31 January 2017: best practice 21 (data is up to date, the frequency is named, a person is responsible for the release), best practice 22 (if data is not available, it says how to get it), best practices 29 and 30 (feedback can be left and is visible to later readers); the bus stop example with a frequency field
- OMB, memorandum M-25-05, 15 January 2025: an inventory of all assets except national security systems; a new asset within 90 days; reconciliation at least once a year; a `data.json` file in the DCAT-US 3.0 schema; deadline 30 September 2026; the open data plan describes how to ask for an asset to be moved up the queue
- GSA, DCAT-US v3.0 Implementation Guide, version 1.1 of 9 September 2026: a contact is mandatory; removing a dataset comes with an explanation; for geodata, DCAT links to an existing ISO description rather than rewriting it
- GSA, How to get your Open Data on Data.gov: the national catalog harvests the description file, the agency's files stay with the agency
- guides.data.gouv.fr: publishing a dataset (in French): with regular updates the page goes through the API; for rare ones a form is enough
- Check on 14 October 2026: `www.gsa.gov/data.json` redirects to `open.gsa.gov/data.json`, 344 records, DCAT-US 3.0 schema, a contact on all 344, access levels 310 / 33 / 1, a licence on 320, quarterly frequency on 121 and for 112 of them `modified` no later than 2020, empty frequency on 89, 20 links to `www.asap.gsa.gov` (the host does not open), the inventory's own record modified on 5 October 2026; `eac.gov/data.json`: 11 records, DCAT-US 1.1 schema
- Requirements Checklist, good practices, metadata completeness, licences, access and roles

Metadata Standards

A metadata standard is a common language of description. A catalog serves its records so that another portal, a search engine or a program can understand them. Your own set of fields is understood only by those who have read your instructions.

The main conclusion of this chapter: take the language the recipient already reads, and often two languages at once. A federation reads its country's DCAT profile, a search engine reads schema.org markup on the page, DOI registration reads the DataCite schema. These languages describe the same dataset and do not replace one another. A draft of the next version and "JSON loosely based on DCAT" are no good for exchange.

How a platform serves the file and which profile it enables by default is in the chapter on export. Which fields to fill in so that the dataset can be used is in the chapter on metadata completeness. Here: which language for which catalog, and how to recognise it from a file.

Describing a dataset versus structuring data

Standards around data come in two kinds, and they are often confused.

A **metadata standard** describes a dataset from the outside: what it is, who the publisher is, what period it covers, where to download it, on what terms. A catalog, an aggregator or a search engine reads it. DCAT, DataCite, ISO 19115 and schema.org are metadata standards.

A **data standard** describes what is inside the file: which columns or fields, what they mean, how values are encoded. A program that processes the data itself reads it. OCDS for procurement, GTFS for timetables and Darwin Core for species observations are data standards. They are covered in the chapter on what is published.

A catalog always needs the first kind when someone outside harvests its descriptions. It needs the second when the sector has one: a procurement dataset in OCDS is immediately understood by contract analysis programs, while a home-grown CSV about contracts has to be parsed anew by everyone. In DCAT, the link between a record and a file standard is recorded with the `dct:conformsTo` property. A column dictionary is a separate task, covered in the section on the data dictionary.

Some standards do both jobs. SDMX describes both the indicator and the structure of the series. DDI describes both the survey and its variables. Croissant describes both a machine learning dataset and its files with fields. STAC describes both an image collection and individual files.

If you do not know the right standard, look it up in a standards directory: FAIRsharing for science, the Data Standards Directory for government data, the data.gov list for the US.

Four layers that get mixed up

When people say "we support DCAT", four different things get mixed up. Until they are separated, you cannot tell whether a dump will pass validation.

Layer	Which question it answers	Example
Vocabulary	what a field means	DCAT, schema.org, DataCite, ISO 19115
Profile	which fields are mandatory and which list a value comes from	DCAT-AP, DCAT-US, GeoDCAT-AP
Controlled vocabulary	which exact value may be written	data theme, licences, language codes
Serialisation and protocol	in which file the answer arrives and how to ask for it	JSON-LD, Turtle, XML; OAI-PMH, CSW

The vocabulary sets the meaning. DCAT says that a catalog has datasets, a dataset has distributions, and a distribution has a link to a file. A profile narrows the vocabulary for a country or sector: it keeps the needed fields, marks them mandatory or recommended and ties them to controlled vocabularies. DCAT-AP and DCAT-US are profiles of the same DCAT, not two different languages. A national profile, such as the Italian DCAT-AP_IT or the Swiss DCAT-AP-CH, sits on top of DCAT-AP and adds its own fields.

A serialisation is a file. The same DCAT description can be served as Turtle text, XML or JSON-LD. For a person there is no difference; for a program there is: it asks for the right form with an `Accept` header or takes an address with the right extension. A protocol is a way to query a catalog: OAI-PMH, CSW, OGC API Records. A single CSW can answer with both an ISO 19115 record and a DCAT description, if the software supports it. More on ways of serving in the chapter on export.

The practical consequence follows. A portal can honestly serve DCAT and still fail the validator of the profile a harvester expects. On 13 October 2026 the data.gov.gr dump was in the DCAT-AP 3 profile, and the DCAT-AP 3.0.1 validator found two errors: the file format was written as the word "CSV" rather than as a link to a controlled vocabulary, and the checksum had no algorithm specified. The analysis is in the chapter on export.

What to take as the basis

Catalog	What to serve outwards	What to rely on	Whom this does not cover
Open data portal	the country's profile and schema.org on the page	DCAT-AP 3.0.1 in the EU; DCAT-US 3.0 in the US; in Russia, the "passport" from methodological recommendations 4.0	a search engine does not read DCAT alone, a harvester does not read page markup alone
Geoportal	ISO 19115; for an open data portal, GeoDCAT-AP as well	ISO 19115, in the EU INSPIRE and GeoDCAT-AP 3.1.0; image files: STAC	a layer record does not replace a description of files, STAC does not replace the country's infrastructure
Research repository	DataCite for DOIs, Dublin Core over OAI-PMH, schema.org on the page	DataCite 4.7; to transfer a folder of files, RO-Crate	a DOI is not registered with a DCAT record
Indicator catalog	SDMX for partners; StatDCAT-AP if tables must reach an open data portal	SDMX 3.1, in practice often still 2.1; tables in the browser: JSON-stat	SDMX does not make a dataset findable in Google Dataset Search
Microdata catalog	DDI per survey and schema.org on the page	DDI Codebook 2.5; the new 2.6 since 15 April 2026	DDI does not replace access conditions or DOIs
ML catalog	a card for people and Croissant for programs	Croissant 1.1 of 29 January 2026; platforms in 2026 still serve 1.0	Croissant does not describe an indicator and does not replace a licence
Vocabulary registry	SKOS; for data elements, ISO/IEC 11179	SKOS; ISO/IEC 11179-3:2023	a glossary does not replace a dataset catalog

Two languages at once is normal work, not a fallback. A catalog's internal form can be richer than the profile: your own editorial fields stay with you. What goes outwards is the profile the harvester named.

DCAT and profiles

DCAT is the W3C vocabulary for data catalogs. The first recommendation came out on 16 January 2014, the second on 4 February 2020, the third on 22 August 2024. Version 3 describes a catalog, a dataset, a distribution, a data service, a dataset series and a catalog record. It introduced series of separately published datasets and version properties, including a file checksum. How these parts look on a live record is analysed in the chapter on catalog anatomy.

DCAT itself declares almost nothing mandatory. Obligation appears in the profile. That is why "we have DCAT" does not tell you whether data.europa.eu or data.gov will accept the file.

Profile	For whom	Status as of October 2026
DCAT-AP 3.0.1	European portals and data.europa.eu	a SEMIC recommendation of 27 October 2025. The 3.0.2 draft of 17 June 2026 has not yet become a recommendation
National profiles on top of DCAT-AP	a single country's portal	follow the version the national harvester checks. Examples: DCAT-AP_IT and DCAT-AP-CH
DCAT-US 3.0	US federal agencies, the `data.json` file	published 1 May 2026. data.gov continues to harvest version 1.1 while agencies migrate
DCAT-AP HVD	high-value datasets in the EU	release 3.0.0, October 2024, on top of DCAT-AP
GeoDCAT-AP 3.1.0	translating an ISO 19115 record into DCAT	a SEMIC recommendation of 16 February 2026, aligned with DCAT-AP 3.0.1
StatDCAT-AP	statistical tables on an open data portal	the current version is 1.0.1 (2019). Version 3.0.0 has been a working draft since 10 June 2026
HealthDCAT-AP	health data catalogs for the EHDS	a European Commission draft; release 7, May 2026. The number has already shifted within the same year, so take the latest release
mobilityDCAT-AP 3.0.0	national access points for transport data	a NAPCORE recommendation on top of DCAT-AP 3.0.0, not yet on top of 3.0.1

A sector profile is not run as a second catalog. It is produced by translating the description the sector already uses. That is what GeoDCAT-AP is for: the geoportal keeps an ISO 19115 record, and the open data portal reads it as DCAT. Likewise, StatDCAT-AP exists so that a statistical office's tables appear among open data without replacing SDMX.

In DCAT-AP 3.0.1 the mandatory dataset fields are the title and description, while the licence lives on the distribution and is not named as mandatory. For people, the licence, update date and period matter more than this minimum. The full table of mandatory and recommended fields is in the chapter on metadata completeness.

Data spaces publish their catalog in DCAT and record usage conditions in ODRL: who may take the data and for what. The description remains ordinary DCAT even when the file itself is closed. How exchange works is in the chapter on other catalog types.

A passport instead of DCAT

In Russia, open data descriptions are defined not by DCAT but by methodological recommendations version 4.0. They were approved by minutes No. 49pr of the presidium of the Government Commission on Digital Development of 12 December 2024. The unit of description is the dataset "passport": identifier, title, description, owner, responsible person and their contacts, a link to the file, format, publication, modification and currency dates, keywords, links to data versions and structures.

The machine-readable passport for parsing someone else's site is one file for the whole list: `meta.csv`, `meta.xml` or `meta.json`. The CSV requirements name four columns: `property`, `title`, `value` and `format`. The version of the recommendations is recorded in the `standardversion` field. The data file and the structure file sit next to the passport. On government websites built on Gosweb, each dataset has three CSVs: the passport, the structure and the data. How such sections work is in the chapter on open data portals.

The sample in the same section has fallen behind the requirements. The reference `meta.csv` contains only the `property` and `value` columns and links to the old address of standard 3.0, `http://opendata.gosmonitor.ru/standard/3.0`, although the cover says version 4.0. Separately, in the section on RDFa, the recommendations advise marking up HTML with the schema.org or Dublin Core vocabulary. Page markup and the passport are not brought together into one profile in the document.

The practical conclusion. The passport is read by the Russian portal and its page parser. data.europa.eu, data.gov and Google Dataset Search do not read it. If datasets only need to be found inside that system, the passport is enough. If someone else must find them, schema.org on the page is added to the passport, along with the DCAT profile a specific harvester has named.

The other languages

schema.org/Dataset is markup in a dataset's HTML page, usually as a JSON-LD block. Search engines read it. Google asks for a name and a description of 50 to 5,000 characters, and calls the licence, author, period and territory recommended. Markup describes one dataset at a time and does not replace a file of the whole catalog. A page that a script draws in the browser is often empty for a robot. How to check this is in the chapter on discoverability.

ISO 19115 describes spatial data: extent, coordinate system, quality, constraints. A record is written to a file per ISO 19139 or ISO 19115-3: in XML this shows in the `gmd:` and `mdb:` prefixes. INSPIRE says how a European public authority must serve such a record. Satellite images and other spatial files are described by STAC: a collection has thousands of files, each with its own date and territory. More in the chapter on geoportals.

Dublin Core is a short set of fields: title, creator, date, subject, rights. Research repositories serve it over OAI-PMH, and almost every archive understands it. **DataCite** is more detailed and is needed to register a DOI. The current schema is 4.7 of 3 March 2026. The identifier, creator, title, publisher, year and resource type are mandatory. A licence is optional in the schema: its absence does not prevent getting a DOI, but it does prevent a person from knowing whether the files can be used. How the DOI itself works is in the chapter on identifiers.

DDI describes a survey: the questionnaire, the sample, the variables, the files. **SDMX** describes an indicator, its dimensions and the series. Both are needed within their sector. Outwards, to an open data portal and to a search engine, a translation is made from them anyway: StatDCAT-AP and schema.org. The analysis is in the chapters on microdata and on indicators.

Croissant describes a machine learning dataset on top of schema.org: files, fields, a split into training and test parts. Version 1.1 of 29 January 2026 added provenance using the PROV-O model, usage conditions (DUO and ODRL) and links to external vocabularies. In a file the version shows as `dct:conformsTo` with the value `http://mlcommons.org/croissant/1.1`. Platforms in 2026 still serve 1.0. A card for people is a separate text alongside. More in the chapter on ML catalogs.

SKOS describes a vocabulary: a concept has an address, labels, a code and "broader/narrower" links. **ISO/IEC 11179** describes a data element in a registry: a definition, permitted values, an owner. These are the languages of metadata catalogs, not dataset portals. The analysis is in the chapter on metadata catalogs.

Packaging and combining existing standards

Several specifications often end up on the same list as catalog standards, although they answer a different question: how to package files, or which existing standards to combine.

RO-Crate packages a study's files together with their description. Next to the data sits `ro-crate-metadata.json`, a JSON-LD file based on schema.org. The current recommendation is 1.3 of 22 June 2026. It is used to transfer and deposit a research object: the data, the programs, the article on how they were obtained. A national catalog is not harvested with this file. Dataverse has an RO-Crate export as a separate community exporter, not as a portal exchange format.

CDIF (Cross-Domain Interoperability Framework) is CODATA's guidance on which existing standards to combine when data crosses a discipline boundary. Version 1.0 came out in May 2024, version 1.1 in June 2026. For discovery, CDIF recommends schema.org and DCAT, not a new catalog vocabulary. Each discipline's own standards remain: DataCite, DDI, ISO 19115.

Data Package and **BagIt** describe a folder of files. In the first, a `datapackage.json` sits alongside with a list of tables and a column dictionary: this is how a dataset is published in Git. In the second, a checksum manifest: this is how an archive checks that a package is intact. OCFL sets out how a repository lays out such packages on its own disk. None of the three tells a portal harvester how to walk a catalog. On publishing as a folder, see the chapter on Git repositories; on archival packaging, the chapter on archiving.

CodeMeta works like schema.org for a dataset, but for software: authors, version, a link to the repository. It is placed next to the data if code is published too. It does not become a dataset catalog.

Controlled vocabularies

Free text in a "theme" field does not join up across portals. "Transport", "Transportation" and "transp." are three different themes to a program. The value is taken from a shared list and written as its address or code.

For open data in Europe these are the thirteen data themes of the EU Publications Office: agriculture, economy and finance, education, energy, environment, government and public sector, health, international issues, justice, regions and cities, population and society, science and technology, transport. For geodata, the nineteen ISO 19115 topic categories: biota, boundaries, elevation, inland waters and others. You can have your own classification. Outwards it is mapped onto the shared list, as EU national portals do: their own headings inside, the data theme in DCAT-AP.

A licence is written as a link, not as the phrase "open licence". In DCAT-AP the link points to the EU Publications Office licence vocabulary. In a repository and in a `CITATION.cff` file the same licence is recognised by its SPDX code, e.g. `CC-BY-4.0`. A Russian reader will understand the words "standard terms of use", but an international catalog's program will not: there is no such code in its list. More in the chapter on licences.

The language is written as a code from the EU language vocabulary or ISO 639, and the territory as a country and region code or a link to a recognisable object such as GeoNames, not as "our region". A code survives harvesting better than a name: when moved from catalog to catalog, a city name is lost more often than an identifier. In a research description the publisher is written as an organisation identifier from the ROR registry, and the author as an ORCID identifier. Then one organisation and one person do not split into different spellings of the name.

Your own controlled vocabulary is justified if there is no shared one. Then it is published separately, each value gets a persistent address, and it is described in SKOS. A list in a form field's hint does not count as a controlled vocabulary: it cannot be referenced from another catalog.

How to choose

Situation	Decision
Descriptions are harvested by a national portal or data.europa.eu	the profile and version their validator checks. A neighbouring version will not do
Datasets must be findable by search	schema.org on every dataset page, even if DCAT already exists
A DOI is needed	the DataCite schema. A DCAT record does not register a DOI
A map layer must also appear on an open data portal	ISO 19115 inside, GeoDCAT-AP outwards
Only your own staff see the catalog	no external standard is needed
A Russian authority publishes to the federal portal and nowhere else	the recommendations 4.0 passport
The same authority wants its datasets found from outside	keep the passport and add schema.org; DCAT when a harvester names a profile
The discipline has its own schema	keep it inside, serve a common profile outwards
You want your own fields	in the catalog's form, yes. In a file for someone else's harvester, only a known profile
A dozen datasets with no harvester	schema.org and one file with all descriptions. Full DCAT-AP is overkill

In brief, this is what a justification looks like using the decision record template. The record on which CKAN extension to enable is already in the chapter on export. Here the fork is different: the national passport or "an international standard".

- **Context.** A ministry publishes about two hundred datasets. The federal portal and Gosweb sites expect a passport under recommendations 4.0. Staff ask to "do it like in Europe, in DCAT". There is no specific harvester that would name a DCAT-AP version. It would be good to see the datasets in dataset search.
- **Options.** Replace the passport with DCAT-AP. Keep only the passport. Keep the passport for the Russian portal, mark up pages with schema.org and not write a home-grown JSON.
- **Criteria.** Who actually harvests the descriptions, whether the dataset gets into search, how much it costs to maintain two descriptions of one dataset.
- **Decision.** Keep the passport. Add schema.org to the dataset page. Enable DCAT-AP when a specific harvester names a profile version. Do not make a home-grown "almost DCAT" format.
- **Why.** The Russian portal reads the passport. A search engine reads the page markup. Nobody would validate DCAT-AP without a harvester, and a home-grown JSON would have to be explained separately to every new recipient. Both descriptions come from one record: the second is not typed by hand.
- **When to review.** If the federal portal starts harvesting DCAT, if a dataset has to get into data.europa.eu or a sector aggregator, or if DCAT-AP 3.0.2 becomes a recommendation and the draft stops being accepted.

How to recognise a standard from a file

The commands for checking that a dump opens and passes a validator are collected in the chapter on export. Here: how to tell from the contents which language you are looking at. Look at the start of the file, not at the name of the "download metadata" button.

What is visible in the file	Which standard it is
`property` and `value` columns, a `standardversion` field	the passport from the Russian methodological recommendations
`conformsTo` with the address of DCAT-AP, DCAT-US or another profile version	a DCAT profile; the number in the address matters more than the word "DCAT" in the instructions
`@type` with the value Dataset and the schema.org vocabulary in `@context`	schema.org markup. If `conformsTo` points to `mlcommons.org/croissant`, it is Croissant
A `ro-crate-metadata.json` file	RO-Crate
XML with a `gmd:` or `mdb:` prefix	an ISO 19139 or ISO 19115-3 record
The `datacite.org/schema/kernel-4` namespace	the DataCite schema
`datapackage.json` next to tables	Data Package, a dictionary of files, not a portal catalog

Then two checks. First: the version in the file matches the one the harvester named. DCAT-AP 2 where 3.0.1 is expected, and DCAT-US 1.1 where 3.0 is already accepted, are different profiles. Second: the field values come from a controlled vocabulary. To a validator, the word "CSV" in the format field and a file type address from the vocabulary are not equal, as the data.gov.gr dump showed.

Where people usually go wrong

Calling your own JSON a standard. The fields resemble DCAT, the names are your own, there are no controlled vocabularies. Every harvester parses such a file separately, and it fails at the first profile check.

Enabling the dump and assuming the profile is right. An extension's or a cloud service's default profile is often older than the one the country expects. It is switched on in settings. On data.gov.gr the profile was enabled, and two fields still failed the validator.

Taking a draft as the current version. As of October 2026 the drafts are DCAT-AP 3.0.2, StatDCAT-AP 3.0.0 and HealthDCAT-AP. They are read to understand where the profile is heading. For exchange, serve the version that has recommendation status and that the harvester's validator accepts.

Free text instead of a code. A theme, licence, language and territory written as phrases do not join up with someone else's catalog. A mandatory field filled with a placeholder formally passes and is empty in meaning.

Covering all tasks with one standard. DCAT does not register DOIs, DataCite does not feed data.europa.eu, the passport does not get into dataset search, Croissant does not describe an indicator. Each recipient gets its own language from the same record.

Confusing a dataset description with a file's structure. You cannot publish procurement "in DCAT": DCAT describes the record, OCDS describes the contracts themselves. The record then says, via `dct:conformsTo`, that the file conforms to OCDS.

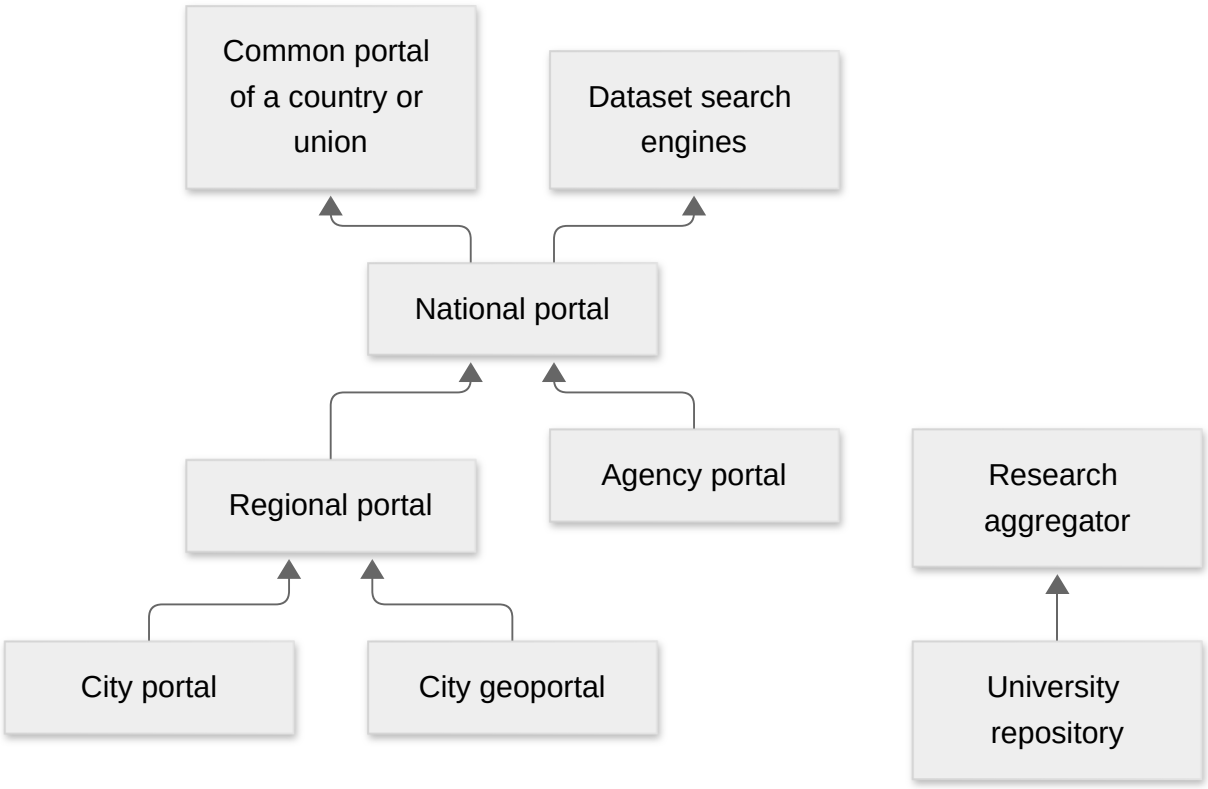
Sources

- DCAT 3, W3C recommendation of 22 August 2024; the previous one is DCAT 2 of 4 February 2020
- DCAT-AP 3.0.1, SEMIC recommendation of 27 October 2025; draft 3.0.2 of 17 June 2026
- DCAT-US 3.0, published 1 May 2026; announcement of 26 May 2026. data.gov accepts version 1.1 during the transition
- GeoDCAT-AP 3.1.0, recommendation of 16 February 2026; DCAT-AP HVD; StatDCAT-AP 3.0.0, working draft of 10 June 2026; HealthDCAT-AP; mobilityDCAT-AP 3.0.0
- DataCite schema 4.7, released 3 March 2026
- Croissant 1.1, published 29 January 2026
- RO-Crate 1.3, recommendation of 22 June 2026
- CDIF, version 1.1, June 2026, text at cdif.codata.org
- Methodological recommendations on publishing open data, version 4.0, annex to minutes No. 49pr of 12 December 2024; the document's page on the Ministry of Finance website (in Russian), published 14 January 2026. The reference `meta.csv` in the text links to `http://opendata.gosmonitor.ru/standard/3.0`
- The EU Publications Office data theme vocabulary and licence vocabulary
- schema.org/Dataset, Dataset markup in Google's documentation

- Standards directories: FAIRsharing, Data Standards Directory, standards on resources.data.gov, checked 13 October 2026
- I. Begtin, Data standards (in Russian), January 2021: the difference between metadata standards and data standards
- Metadata export, metadata completeness, open data portals

Federation and Harvesting

A federation is a chain of catalogs. A city publishes datasets, the region harvests them, the country harvests from the region, and a common portal such as data.europa.eu harvests from the country. Dataset search engines stand to the side and read whoever serves descriptions to machines.



Fetching descriptions on a schedule is called harvesting. What is taken is not the data files but the records: title, licence, address, date. A person or a program downloads the file itself from the source.

What the source does

- Serves all descriptions at one address: a DCAT file, an API, CSW or OAI-PMH. Pagination is fine too, if the order of records is stable and the end of the list can be reached. Search results ten at a time, with an order that changes from query to query, are no good for harvesting.
- Does not put this address behind a captcha or registration.
- Keeps the dataset address persistent. If a dataset is removed, its record says so rather than vanishing without a trace.
- Writes the licence on the dataset. An aggregator has no right to invent it.

The profile must match the one the higher-level portal expects. The higher-level portal silently skips "almost DCAT", and the city concludes that its data "wasn't taken".

What the harvester does

- Visits on a schedule and does not take the whole catalog every ten minutes.
- Stores the source address and the harvest date.
- Does not multiply duplicates: a dataset from the city portal and its copy on the regional one are one record with a link to the original source.
- Removes or flags records whose source no longer opens.

In CKAN the harvesting framework is the ckanext-harvest extension, and RDF harvesting is part of ckanext-dcat. In Piveau, harvesting DCAT is the core's main job: Fraunhofer FOKUS built the platform around harvesting and checking the quality of descriptions, and data.europa.eu itself runs on it. Details on extensions are in the chapter on CKAN.

What it looks like in practice

Europe. The state portal Open.NRW harvests descriptions from agencies and cities in North Rhine-Westphalia. Germany's national portal GovData harvests Open.NRW along with other states: in October 2026 GovData had about 10,600 datasets from Open.NRW. Descriptions flow in the German DCAT-AP.de profile. The pan-European data.europa.eu harvests GovData and other national portals, once a day by default, and itself translates titles and descriptions into all official EU languages.

United States. Every federal agency publishes a `data.json` file in the DCAT-US profile, and the data.gov harvester collects these files. The status of each source is visible on a separate site, harvest.data.gov. After data.gov moved to its own software in 2026, the harvester remained a separate program that writes to a shared database, which the showcase only reads.

EU geodata. Countries serve layer descriptions through CSW catalog services under INSPIRE rules. Until 1 July 2026 a separate INSPIRE geoportal harvested them; it is now closed, and data.europa.eu harvests national geocatalogs, translating ISO records into DCAT via the GeoDCAT-AP profile. The layers themselves stay in national services. More in the chapter on geoportals.

Research. University repositories serve descriptions over OAI-PMH in the format set by the OpenAIRE guidelines, and OpenAIRE collects them into a common graph of research outputs.

Thailand: one platform for all levels. The central government data catalog gdcatalog.go.th runs on CKAN, and in October 2026 it held about 25,000 datasets. Provincial catalogs are not separate sites on different software but subdomains of the central one on the same platform, for example ranong.gdcatalog.go.th. In the central catalog the provinces appear as organisations, and geodata is shown by the geo.gdcatalog.go.th geoportal on TerriaJS. Profiles do not diverge because the source and the harvester are one system. The price: regions do not choose their own platform.

Scotland: search first, then a portal. For a long time Scotland had no national portal. Instead, the government built the find.data.gov.scot search engine: it harvests descriptions from the sites of agencies, councils and other data holders via their APIs, and where there is no API, it parses pages. In October 2026 it had about 26,500 datasets from 70 sources and 922 owners, with a page for each dataset. Only in 2026 did the data.gov.scot portal start working, still in beta and with 290

datasets: it is where those without a place of their own publish, and it serves descriptions in DCAT and schema.org. This order is useful when data is already scattered across a hundred sites: a search engine quickly shows what exists, and the portal is needed for the rest.

Harvesting protocols

Protocol	Who usually serves it	What the harvester gets
DCAT file (RDF, `data.json`)	open data portals	all descriptions in one file
OAI-PMH	research repositories; data.europa.eu prefers it for DCAT-AP too	descriptions page by page, with the option to take only those changed since last time; a file list only if the description format itself contains one
CKAN API	portals on CKAN	descriptions in CKAN format; data.europa.eu accepts it only from legacy systems
CSW	geoportals	layer descriptions in ISO 19115 / 19139
SPARQL	portals on graph databases	an answer to a query over descriptions
schema.org markup and a sitemap	any site	descriptions from pages; this is how dataset search engines read

Being able to take only changes matters for large catalogs: the harvester does not re-download a hundred thousand descriptions every night.

How to get harvested

A higher-level portal usually asks you to fill in a short questionnaire. data.europa.eu's is public: the address of the dump endpoint, the description profile (DCAT-AP, CKAN format or INSPIRE), the protocol, the date format per ISO 8601, whether a login is needed. The request is then sent through the feedback form. Once connected, the portal sees a quality score for its descriptions in MQA. It is useful to go through such a questionnaire before choosing a platform: it shows at once what is missing.

Where the chain breaks

A municipality sets up its own portal that the region cannot read. The region copies records by hand. The national portal sees the region and not the cities. A search engine sees three copies of one budget.

The longer the chain, the more is lost along the way. data.europa.eu is an aggregator of aggregators: it harvests national portals, which themselves harvested cities and agencies. It shows clearly where exactly the chain breaks.

The licence and the source get lost. In August 2024 a dataset on environmentally hazardous facilities in the city of Malmö was on the city portal under a CC0 licence. Via Sweden's national portal dataportal.se, it reached data.europa.eu without a licence and without a link to the city page. By October 2026 the city had moved to a new portal, the CC0 licence now gets through, and the file links lead to Malmö. But the dataset page on the city portal is still not in the record:

Sweden's national portal is recorded as the catalog. And the old record on data.europa.eu, which someone may have cited, disappeared when the city moved: its address returns a 404 error, because the aggregator builds the address from the source's address.

This is not an isolated case. By I. Begtin's count, in 2024 a licence was given for about 150,000–180,000 of 1.8 million datasets on data.europa.eu, i.e. no more than 10%. In October 2026 a licence as a link to a known text is present on about 775,000 datasets, less than half of the 1.73 million the portal showed that month. Things have improved, but more than half of datasets still arrive without a machine-readable licence, and a noticeable share of "licences" are values like "conditions unknown".

The lesson for every link in the chain: pass on the licence and the link to the dataset page at the source (in DCAT, the `dcat:landingPage` property), and do not change the dataset identifier when moving.

One source outweighs a country. In October 2026 data.europa.eu had about 1.73 million datasets, 850,000 of them German. 681,000 come from one catalog, the GDI-DE geocatalog, and almost 450,000 of those are development plans for individual plots (Bebauungsplan), each as a separate dataset, usually as WMS and WFS services. The national portal GovData provides another 169,000. In 2024 the picture was the same: 609,000 of 726,000 German datasets came from GDI-DE. So "Germany has half of Europe's data" says nothing about openness and everything about how a geocatalog slices data. When comparing countries in an aggregator, look at the number of sources and the types of data, not the total count.

Only part of a country is visible. In June 2024 data.europa.eu had 324 datasets from Estonia: the portal harvested only the national geoportal. By October 2026 the national open data portal had been connected too, and there were 8,700 datasets, 8,300 of them from that portal. How much of a country's data reaches the top depends on which catalogs it has registered. You can check this with the "catalog" filter in the higher-level portal's search: if your catalog is not there, neither are its datasets.

The protocol does not transfer files. OAI-PMH transfers only descriptions: the specification explicitly puts the nature of the resource itself outside its scope. The mandatory description format is simple Dublin Core, which has no field for a file list. So research aggregators that harvest repositories over OAI-PMH usually store a link to the dataset page, not to the files. A repository that wants its data to be downloadable straight from search should serve DCAT or an API with a file list alongside OAI-PMH.

A home-grown harvesting format. Russia's data.gov.ru harvested dataset lists that agencies kept in their own schema from the methodological recommendations. No foreign aggregator reads this format, so the chain broke at the very first step outside the country. More in the anti-patterns.

It is more honest not to set up a lower-level portal if the higher-level one already accepts publications from organisations. This is the usual conclusion for a small city: see the municipality example.

Sources

- Metadata Export

- APIs and Bulk Download
- ckanext-harvest, ckanext-dcat
- Piveau and the data.europa.eu project at Fraunhofer FOKUS
- How to get harvested by data.europa.eu: the questionnaire, protocols, daily harvesting, machine translation
- GovData and the DCAT-AP.de profile; the number of Open.NRW datasets: a query to the GovData API, 11 October 2026
- DCAT-US, harvest.data.gov, the data.gov catalog application
- data.europa.eu, INSPIRE datasets move to the European Data Portal, 22 June 2026: the INSPIRE geoportal closing on 1 July 2026; OpenAIRE guidelines
- data.europa.eu: the number of datasets by country, catalog, keyword and licence via the search API; datasets with a licence link via a query to the SPARQL endpoint; the Malmö dataset record and the old record address (404) via the search API; all on 12 October 2026
- Thailand's government data catalog (CKAN 2.9.2, the `package_search` API), the Ranong province catalog, the geoportal, 12 October 2026
- find.data.gov.scot: counters on the home page; data.gov.scot, beta, 12 October 2026
- The OAI-PMH 2.0 specification: the nature of the resource is outside the protocol's scope, the mandatory format is Dublin Core
- I. Begtin, Telegram (in Russian): data.europa.eu as an aggregator of aggregators: Malmö and licences, 4 August 2024; GovData and the GDI-DE geocatalog, 20 August 2024; Estonia in the European portal, 12 June 2024; Thailand, 15 August 2024; find.data.gov.scot, 3 November 2024; why not to harvest OAI-PMH, 2 April 2024

Licences and Access Conditions

A licence answers the question of what you may do with the data. Access conditions answer a different one: who may get the file. These are two fields. A sentence in the site footer fills in neither.

In brief:

- Each open dataset gets one well-known licence: as a link to the text and as the code the harvester expects. The page, the file and the dump all say the same thing.
- For ordinary open data this is CC BY 4.0 or CC0. A home-grown two-page text and the value "not specified" will not be recognised by a harvester.
- If a country's law requires its own document, it is kept for people, and the field for programs gets a well-known licence that does not contradict that document.
- A closed file gets access conditions and a response time. An open licence is not put on it: it promises what the file does not deliver.

The three conditions under which data is called open are in the chapter on open data portals. Who sees the record and who downloads the file is in the chapter on roles and access.

Four labels that get confused

Label	Which question it answers	Where it is written
Licence	what may be done with the data: copy, change, use in work, including commercial work	the dataset field and each file's field, the same link in the dump
Access conditions	who gets the file and what they must do for it	the access field on the record: open, after registration, on request, only in a secure environment
Site rules	how to behave on the portal: do not overload the server, do not pretend to be the owner	a "Terms of use" page, not a dataset field
Attribution and citation in an article	how to name the author	a licence may require attribution; scholarly citation works separately, see the chapter on citation

A common contradiction: the dataset says CC BY, while the site rules say "non-commercial use only". Then it is unclear which text prevails. Keep one.

What counts as an open licence

The Open Definition (version 2.1) calls open what anyone can freely access, use, modify and share. Attribution may be required. Requiring a derived database to remain under the same terms is also allowed. A ban on commercial use, a ban on modification and a "ask for permission" formula are not open data by this definition.

Hence a practical list.

Licence	What it permits	When it is used
CC0 1.0	a waiver of rights as far as the law allows; attribution is not required	reference lists and facts that should flow into other databases without friction
CC BY 4.0	copy and change, including for commercial purposes, with attribution	the usual choice for open data
CC BY-SA 4.0	the same, but derivative works stay under the same terms	when it matters that adaptations stay open, and the dataset is not going to be mixed with other databases
ODbL 1.0	a licence for a database; a derived database is shared under the same terms	databases run as databases. This is how the OpenStreetMap map is published
National open licence	close in meaning to attribution	when it is already customary in the country and the text is on the harvester's list

The version is written explicitly: "4.0", not "CC BY". The CC BY 4.0 text separately covers the sui generis database right recognised in the European Union. The high-value datasets regulation names this version too. Article 4(3) of Regulation (EU) 2023/138 requires such datasets to be made available under CC0, or CC BY 4.0, or an equivalent or more permissive open licence. CC BY-SA and ODbL are stricter: a derived database must keep the same text. They are not suitable for high-value datasets. Two share-alike licences also mix badly with each other: one database cannot carry two such texts at once. If the goal is for the dataset to be taken into other people's databases, use CC BY 4.0 or CC0.

CC BY-NC and CC BY-ND are not on this list. The first prohibits commercial use, the second prohibits modification.

Which one to use

Situation	Licence
A reference list that should be copied into other databases without extra conditions	CC0 1.0
An ordinary open dataset from a public authority, research or a company	CC BY 4.0
A high-value dataset under EU Regulation 2023/138	CC0 or CC BY 4.0, or a national licence no stricter than them
Adaptations must stay open, and nobody plans to mix the database with others	CC BY-SA 4.0 or, for a database, ODbL
The country already has its own open licence, and the harvester recognises it	that licence
The law requires a national text that has no code on the harvester's list	the text for people, and next to it a well-known licence that does not contradict it
The file is closed	access conditions, not an open licence
A program sits next to the data	the program has its own licence, the table its own

National texts that harvesters already know: the Open Government Licence v3.0 in the UK, the Licence Ouverte 2.0 in France, the Data licence Germany — attribution — Version 2.0 in Germany. Each has an SPDX code and a code in the EU Publications Office licence vocabulary. The code table is below.

Standard terms in Russia

Russian portals often refer to the "Standard terms of use of publicly available information posted on the Internet in the form of open data". They were approved by minutes No. 6 of the absentee vote of the Government Commission for Coordination of Open Government of 19 September 2016.

In the text that opened on 14 October 2026, three clauses pull in different directions. Clause 10 asks that data be used for lawful purposes, with a link to the source and without distorting the information. Clause 13 explicitly permits both commercial and non-commercial use. Clause 12 says no contract is needed and no special agreements or licences are required. The document has no international code. The recommendations 4.0 passport has no licence field either: the passport is read by the Russian parser, and a harvester outside the country will not find this field in the passport even if you add it yourself.

The practical conclusion. The page with the standard terms is kept for people. In the field that goes into the dump, a lawyer chooses a well-known licence that does not contradict these clauses. The closest to "a link to the source, and commercial use is permitted" is CC BY 4.0. If you do not want to require a link to the source, CC0 is closer. The document itself does not declare this correspondence: the authority's lawyer must confirm it, not the catalog record.

Where it goes

The licence goes on every dataset. A single licence for the whole catalog is acceptable only when there really is one and it is copied into the dump for every dataset. A harvester does not read the site footer.

In DCAT-AP 3.0.1 the licence is a recommended property of a distribution (a file or service), not of a dataset. The profile does not call it mandatory. The profile deliberately does not attach the licence to the dataset: if the dataset has one text and the file another, it is unclear which prevails. For people and for harvesters the field is needed all the same. Where it sits among other properties is in the chapter on metadata completeness.

In CKAN the licence is stored on the dataset. The ckanext-dcat extension carries it into the DCAT dump. According to the documentation for release 2.4.4, a file inherits the dataset's licence only if the `ckanext.dcat.resource.inherit.license` setting is enabled. By default it is off. A filled-in dataset field may then not reach the file in RDF.

On 14 October 2026, on data.gov.gr, the records of one licence for the dataset `emy-station-116` diverged. In the dataset API the code is `cc-by`, and the link points to an Open Definition page without a version number. The first file already links to the `CC_BY_4_0` concept in the EU vocabulary. In the RDF that opened at `dataset_series/emy-station-116.rdf`, the dataset itself has no licence property: `CC_BY_4_0` sits on the distribution and on the data service. Whoever checks only `license_url` in the API will not see version 4.0. Whoever checks only the dataset field in the DCAT dump will conclude there is no licence, although the file has one. On data.gov.gr, in the sample from the chapter on metadata completeness, 32% of datasets and 99% of files had a licence. Both places and the dump need checking.

The harvester also loses the licence along the way. On data.europa.eu in October 2026 about 40% of dataset records had a link to a known text. How this happens is in the chapter on federations.

Then there is the empty placeholder. The value "licence not specified" is also a value. On data.gouv.fr the built-in score counts it as a filled-in licence. The CKAN code `other-open` ("some other open licence") says the licence is supposedly open and does not say what its text is. A strict check expects a link to a known text. How shares are counted is in the chapters on completeness and on metrics.

Three ways of writing one licence

Programs look for the same licence in different ways. A `CITATION.cff` file and the SPDX list expect a code like `CC-BY-4.0`. CKAN by default expects its own short code without a version. DCAT-AP recommends a link to a concept in the EU vocabulary. Zenodo takes the SPDX list but writes the code in lower case in its API: for the record 10.5281/zenodo.4898406 on 14 October 2026 the field said `cc-by-nc-4.0`.

Licence	SPDX	Default CKAN code	EU vocabulary code
CC0 1.0	`CC0-1.0`	`cc-zero`	`CC0`
CC BY 4.0	`CC-BY-4.0`	`cc-by`	`CC_BY_4_0`
CC BY-SA 4.0	`CC-BY-SA-4.0`	`cc-by-sa`	`CC_BYSA_4_0`
ODbL 1.0	`ODbL-1.0`	`odc-odbl`	`ODC_BL`
Open Government Licence v3.0	`OGL-UK-3.0`	`uk-ogl`	`OGL_3_0`
Licence Ouverte 2.0	`etalab-2.0`	not in the default list	`ETALAB_2_0`
Data licence Germany, attribution, 2.0	`DL-DE-BY-2.0`	not in the default list	`DLDE_BY_2_0`

The EU vocabulary link is built as `http://publications.europa.eu/resource/authority/licence/` plus the code from the table. The ODbL code in this vocabulary is `ODC_BL`. You will not find it by the letters "ODbL".

The CKAN 2.12 core has fifteen licences. For `cc-by` and `cc-zero` the link points to an Open Definition page rather than the Creative Commons text, and there is no version 4.0 in the code. For `uk-ogl` the link does point to the version 3.0 text. `other-open` has no link. Your own list is connected with the `licenses_group_url` setting: it is a JSON file, not a core change. On 14 October 2026 data.gov.gr served exactly those fifteen values. open.canada.ca replaced the list: its national licence has the code `ca-ogl-lgo` and a link to the Open Government Licence — Canada text. In the SPDX list it is written as `OGL-Canada-2.0`.

data.gouv.fr has its own codes too. On 14 October 2026 a mandatory procurement dataset of the Auvergne-Rhône-Alpes region had the code `fr-lo` in the API. In RDF the distribution had a link to a 2014 PDF, `etalab.gouv.fr/wp-content/uploads/2014/05/Licence_Ouverte.pdf`. The same day the address opened and redirected to a PDF named licence-ouverte-2-0. There is no `ETALAB_2_0` code in the link. A harvester that recognises only the EU vocabulary will not match this licence, although the right text is at the link.

Several sets of rights in one package

A table, the program next to it and the photos inside it may carry different rights. The record states the data licence. The program's licence goes into the code's `LICENSE` file. If the dataset contains someone else's rows, the description says so: someone else's licence does not become the licence of the whole dataset just because the file sits in the same folder.

Four places must name the same text: the page, the `CITATION.cff` file, `datapackage.json` and the record on a shared hub. A divergence has already been analysed with Global Jukebox: in the repository tag the `LICENSE.txt` file is CC BY 4.0, while the Zenodo record by DOI is CC BY-NC 4.0. Details are in the chapter "When You Don't Need a Portal".

Platforms add another confusion: the publisher's agreement with the archive and the licence for whoever downloads the data.

In the DSpace 10 submission form, the mandatory "License" step is an agreement with the archive. Its text is in `default.license` and can be specific to a collection. The "CC License" step is separate and optional: it sets the licence of the record itself. A checkbox saying "I agree to deposit the work in the archive" is not the dataset's licence.

In Dataverse a record gets a licence from the installation's list or, if the installation allows it, custom terms. The documentation does not offer both at once. Often, if a person chose nothing, CC0 is filled in: this is an installation setting. The community norms ask people to cite the dataset anyway. The documentation says plainly that these norms do not replace a licence and do not create a legal obligation.

When the file is closed

Restricted access is needed for personal, sensitive and licensed data. The description is still published openly: the dataset can be found and requested. Closing the record together with the file hides the very existence of the data. Levels, platforms that can keep the description open, and the `dct:accessRights` property are analysed in the chapter on roles and access.

An important addition here. An open licence on a closed file is a lie. CC BY promises the file can be copied. If the file is released on request, the record states the access conditions, the request procedure and the response time. "On request" without a deadline is not an access level.

Registration is also a barrier, even if the field says CC BY. A portal that releases data only after an identity check is, by the three conditions of openness, publicly accessible rather than open. This is how the Guangdong provincial portal works, and how Kazakhstan's national portal worked until registration was abolished: examples are in the chapters on open data portals and on anti-patterns.

Personal data is anonymised before publication. Assessing whether a person can be identified from the remaining fields is separate work. If a risk remains, the file is not put into an open catalog.

According to the dataportals-registry export (release 1.27.0 of 15 October 2026), restricted access is marked for about 1,100 catalogs. This is a mark on the whole catalog, not a share of closed datasets inside it.

How visible the licence is from outside

In the same registry export, a licence text or link at the level of the whole catalog could be filled in for about 1% of records (about 420 of 43,000). This means the licence is hard to find even for someone describing catalogs. The model, "one licence for the whole catalog" or "a licence on each dataset", is identified more often, for about 17,000 catalogs, split equally between the two. For another 5,600 the model is explicitly unknown. The remaining records do not belong here: no model is named for them. These three numbers do not add up to 43,000. The licence text and the licence model are different fields. The queries are in the appendix.

The share of licences inside a portal is not counted from this mark. It is counted from the dataset dump, as in the chapters on completeness and metrics.

Example: your own text or a well-known licence

Context. A regional authority publishes about two hundred open datasets and several surveys with personal data. The lawyer has brought two pages of "terms of use" and asks for them to be put on everything. The national portal harvests descriptions in DCAT-AP and recognises the licence by the EU vocabulary. Some datasets are on a list for which a licence no stricter than CC BY 4.0 is expected.

Options. Leave the lawyer's text in the footer. Paste this text into the licence field of every dataset. Put CC BY 4.0 on the open files as an EU vocabulary link, keep the lawyer's page for people if it does not contradict CC BY, and describe the closed surveys with access conditions.

Criteria. The harvester recognises the licence without parsing a paragraph. The text on the record and the text in the dump do not conflict. A closed file does not promise free copying.

Decision. Open datasets: CC BY 4.0, with the link `.../licence/CC_BY_4_0` in the dump and the code `CC-BY-4.0` in `CITATION.cff`. The page with the national text stays, and the lawyer confirms in writing that it does not tighten CC BY. Surveys: an open description and access conditions with a response time, without CC BY.

Why. The footer does not get into the dump. A home-grown paragraph matches no code in the vocabulary, and the national portal will show the dataset without a licence. One label for open and closed files lies about one of the two groups. CC BY-SA drops out: it is stricter than the regulation allows for high-value datasets and hinders mixing databases.

Consequences. On CKAN, licence inheritance by files is switched on, or the file field is filled in separately, and the RDF is checked, not only `license_url`. The core licence list, where CC BY has no version, is replaced by your own file. Surveys get a response time that the old "terms" did not have.

When to review. The country publishes a national licence with an SPDX code that the EU vocabulary already contains. The regulation or the national portal names a different permissible text. A survey is anonymised down to a file that can be released openly.

How to write your own record is in the chapter "Justifying a Decision".

How platforms do it

Platform	Where the licence lives	What matters
CKAN 2.12, core	the dataset's `license_id` field, a list of fifteen values	CC BY has no version, the link goes to Open Definition; your own list is a file set by `licenses_group_url`
ckanext-dcat 2.4.4	carries the licence into DCAT	a file gets the dataset's licence only with `ckanext.dcat.resource.inherit.license`; off by default
udata, core	a mandatory field from its own list	a "not specified" value exists, and the data.gouv.fr record score counts it
Dataverse, core	a licence from the installation's list or custom terms	often CC0 by default; citation norms are not a licence
Zenodo on InvenioRDM	a mandatory field, a list based on SPDX, several allowed, plus custom text	the code is lower case in the API; the default suggestion is CC BY 4.0
DSpace 10, core	agreement with the archive is mandatory; the record's licence is a separate Creative Commons step	the deposit agreement does not by itself go into `dc.rights`
GeoNetwork	legal constraints of the ISO 19115 record	the core has no short code like `cc-by`; a link reaches DCAT if the export profile has mapped this field

Access levels for the same platforms are in the chapter on roles and access. It also says which ones can keep the description open while the file is closed.

Publishing without a portal

Next to the files, one licence is put in four places, with the same text everywhere:

- `CITATION.cff`: the SPDX code; in the book's template it is `CC-BY-4.0`;
- `datapackage.json`: the same code in `licenses[].name` and a link to the text in `licenses[].path`;
- schema.org markup: a link to the text in the `license` property;
- the Zenodo or other hub record: the licence the hub accepts under its own code.

The templates are in the templates folder. GitHub recognises the licence from the `LICENSE` file in the root if the text matches a known one. A file in the root does not replace the field in the hub record: in the Global Jukebox example they diverged.

How to check

Open three random records, not the ones at the top of the list.

- The record has a licence.
- It is a link to a known text or a code from the table above. A paragraph saying "the data may be used" and the value "not specified" do not count.
- In the dump the distribution has the same link. If the field is missing in DCAT, the harvester will lose the licence even if it is written on the site.

- If the dataset and the file have different links, this is deliberate, not because the platform's list does not know the version.
- If the file is closed, the record opens without a login, there is no open licence on it, and it says how to request the file and when an answer will come.

For a CKAN portal, this command shows the code, the dataset link and the licence of the first file. Substitute your own identifier.

```
curl -s "https://data.gov.gr/api/3/action/package_show?id=emy-station-116" \
  | jq '{id:.result.license_id, url:.result.license_url,
         file:(.result.resources[0].license // "empty")}'
```

On 14 October 2026 the answer was: code `cc-by`, link `http://www.opendefinition.org/licenses/cc-by`, and for the file the `CC_BY_4_0` concept. The list the code came from:

```
curl -s "https://data.gov.gr/api/3/action/license_list" \
  | jq '[.result[] | {id, url}]'
```

In RDF the same link is looked for on the distribution. On that day the address `dataset/emy-station-116.rdf` opened `dataset_series/emy-station-116.rdf`, and it contained `licence/CC_BY_4_0`.

Sources

- Open Definition 2.1
- Creative Commons, CC0 1.0, CC BY 4.0 and the licence text, CC BY-SA 4.0
- Open Data Commons Open Database License
- Open Government Licence v3.0, Licence Ouverte 2.0, Data licence Germany — attribution — Version 2.0
- The EU licence vocabulary, a query to the Publications Office SPARQL endpoint on 14 October 2026; SPDX codes
- DCAT-AP 3.0.1, the section on legal information: the licence on the distribution, a recommendation of the EU vocabulary and of CC BY 4.0 for FAIR data
- Regulation (EU) 2023/138, Article 4(3)
- Standard terms of use of publicly available information in the form of open data, minutes No. 6 of 19 September 2016. The text of clauses 10, 12 and 13 was read on 14 October 2026 on a page of a regional office of Rosselkhoznadzor (in Russian), which reproduces the document. The PDF on data.nalog.ru (in Russian) opened the same day; the file properties say 22 September 2016. The document's page on the Ministry of Finance website did not open from this network
- The CKAN 2.12 licence list: ckan/model/license.py. Inheritance in the dump: ckanext-dcat 2.4.4, the key `ckanext.dcat.resource.inherit.license`, default `false`
- Choosing a licence in Dataverse 6.12, Zenodo licences, the DSpace 10 submission form

- Check on 14 October 2026: the data.gov.gr API and RDF (the `emy-station-116` dataset, a `license_list` of 15 values), the data.gouv.fr API (the Auvergne-Rhône-Alpes procurement dataset, code `fr-lo`, a link to a 2014 PDF in RDF), open.canada.ca's `license_list` (code `ca-ogl-lgo`), the Zenodo API for record `4898406`
- The dataportals-registry export, release 1.27.0 of 15 October 2026
- Metadata completeness, roles and access

Discoverability

A dataset is published when it can be found without knowing the portal's address by heart. Otherwise the catalog is an internal drive with a pretty cover.

This chapter is about the catalog as a whole: what the platform must do by itself for hundreds and thousands of pages, where to register the catalog, and how to check that search engines see it. How to make a single page or a repository findable without a portal is in the chapter "Making a Publication Findable".

Search engines

Google Dataset Search and similar systems read schema.org/Dataset markup on the dataset page. Next to it you need a sitemap, through which the search engine walks all dataset pages: according to the dataportals-registry export (release 1.27.0 of 15 October 2026) this is the most common machine address of catalogs, about 7,000 records. Titles and descriptions are written in human language. A dataset called "Export 12" cannot be found.

On a portal, markup and the sitemap are not written by hand: the platform must produce them for every page.

Platform	schema.org markup on the dataset page	Sitemap
CKAN	the ckanext-dcat extension, structured data module	not in the core; third-party ckanext-sitemap extensions, not maintained by the CKAN community, the most recent updated in 2024
Dataverse	in the core	in the core: generated by an administrator command, a large catalog is split into several files
InvenioRDM	in the core	in the core since v13: generated and updated automatically, listed in `robots.txt`
GeoNetwork	in the core: schema.org JSON-LD on the record page; Croissant fields for the ISO 19115-3 schema are in the branch of the upcoming 4.4.13 release, not in 4.4.12	in the core, but the `robots.txt` linking to it sits inside the GeoNetwork address, not at the site root; the sitemap is submitted to search engines by hand
Cloud services	check the dataset page: markup is sometimes trimmed	check the `/sitemap.xml` address

Google advises putting markup on one canonical dataset page, and on copies (for example in search results or on a portal that harvested the dataset from its source) linking to it with the `sameAs` property. This matters for federation: if a regional portal repeats a city's record with full markup and no link to the original, the search engine sees two different datasets.

What a search engine does not see

A search robot reads an HTML page and follows links. Anything that cannot be reached that way does not exist for it, even if a person sees it easily in a browser.

An interface drawn by a script. The page arrives as an empty shell, and a script loads the list of datasets and descriptions from an API. A robot that does not run scripts sees an empty page; so does a web archive. This is how the "Compendia" section of the Rosstat website works: the page HTML contains not a single link to a document, and the list is loaded by a separate request (this is how it looked in the web archive in February 2026). Rosstat documents are still found in search, but mostly because other sites link to them. The dataset page on data.europa.eu is built the same way: the source HTML has neither a description nor markup, and everything is drawn by a script (October 2026). The archive of Accounts Chamber bulletins serves documents through a search API, and robots never reach the PDFs themselves; this case is analysed in the chapter on archiving.

The cure is not to abandon the API but to give every dataset an ordinary page: the server returns HTML straight away with the description, file links and markup, and dataset lists are paged with ordinary links. The API is in addition to pages, not instead of them.

A site closed to foreign addresses. Search engines, web archives, aggregators and AI agents come from cloud data centres in various countries. If a site does not respond to requests from abroad, it drops out of all of them at once. On 12 October 2026 neither rosstat.gov.ru nor ach.gov.ru opened from a foreign server. What this means for a catalog and how to check it is analysed in the anti-pattern "The door is closed to programs" and in the chapter on archiving.

A sitemap the search engine does not know about. A search engine looks for `robots.txt` only at the site root. GeoNetwork puts its `robots.txt` with a sitemap link inside its own address, for example `/geonetwork/srv/api/robots.txt`, and the documentation explicitly advises registering the sitemap with search engines by hand. In October 2026 geodata.nz, geocat.ch and the Estonian geodata catalog had no `robots.txt` at the root. GeoNetwork 4.4 serves schema.org markup on record pages, but without a sitemap a robot may never reach those pages. So GeoNetwork catalogs are found mainly through aggregators (data.europa.eu, which replaced the INSPIRE geoportal from 1 July 2026, and national geocatalogs), not through ordinary search.

Spam has markup too

schema.org/Dataset markup is added by the page owner, and nobody checks it. The Web Data Commons project extracts such markup from the open Common Crawl web index. In the 2024 release, the Dataset type was found on 694,000 pages from about 2,000 sites. By I. Begtin's manual estimate (June 2024), of 1.4 million objects marked up this way, about a third are junk (for example, ordinary reference site articles marked up as datasets), a third are data vendors' pages, and only a third are data that can be downloaded.

The DataCatalog type, used to mark up a whole catalog, is so rare that Web Data Commons did not break it out as a separate subset in either 2023 or 2024. A check of the catalog registry in 2024 found such markup on about one catalog in a thousand.

Large portals mark up differently too. In 2024 data.gov served trimmed markup: the name and keywords. In October 2026, on the new catalog.data.gov, dataset markup includes both the licence and a list of files. data.europa.eu has no markup in its HTML at all.

The practical conclusion: markup by itself does not distinguish your dataset from spam. Fill in what spam usually lacks: the licence, the publisher, a list of files with direct links (`distribution` with `contentUrl`) and a link to the catalog (`includedInDataCatalog`), as in the example in Google's documentation.

Aggregators and registries

A higher-level portal will find you only through a dump you have agreed on. How this works and how to get harvested is analysed in the chapter "Federation and Harvesting".

Independent aggregators and dataset search engines (Dateno, find.data.gov.scot, research aggregators) do not crawl the whole internet either. They keep a list of sources and maintain a harvester for each platform type. Hence two reliable ways to reach them:

- **Publish where harvesting already happens.** A national or agency portal, Zenodo, a domain repository: their harvesters are already set up, and a new dataset reaches aggregators without any effort on your part.
- **Install standard software with a known interface.** CKAN serves its API and DCAT, DKAN a `data.json` file, JKAN a common `data.json`, InvenioRDM a REST API and OAI-PMH. Harvesters for them are already written; it is enough to report the catalog's address. A custom-built portal with its own interface needs a separate harvester, and an aggregator gets to it last or not at all.

How much this matters can be seen in a Russian example. In 2024, of 154 Russian catalogs in the Dateno registry, about 15 were indexed. The reasons given were: government sites are closed to foreign addresses, lists under the methodological recommendations are awkward to harvest, and 40–50 custom-built portals each need their own harvester. Finland, where catalogs run on standard software, was much easier to harvest.

In addition, the catalog is registered where its audience looks for the list of portals itself:

- the national registry or the country's higher-level portal;
- re3data for a research repository: the repository itself submits a request through the site form, and after review a record appears with subjects, conditions and certificates;
- FAIRsharing for a database or a standard;
- the dataportals-registry, which describes catalogs for those looking for a portal rather than a dataset.

Registration does not replace a dump. A registry entry with a broken address is worse than no entry: update it after the portal moves.

Programs

A program needs a stable API or a file with all descriptions, and documentation without a captcha. An `llms.txt` file is a short note for a language model about what is on the site; it is a proposed format, not an adopted standard. MCP is a way to give a program the list of a catalog's queries. Both make sense when such programs already use the catalog. They do not fix the absence of DCAT. More in the chapter on trends.

How to check that the catalog is found

Substitute the portal's address and the address of one dataset page:

```
PORTAL=https://portal.example
PAGE=https://portal.example/dataset/budget-2026

# the sitemap responds and is listed in robots.txt
curl -s "$PORTAL/robots.txt" | grep -i sitemap
curl -s -o /dev/null -w "%{http_code}\n" "$PORTAL/sitemap.xml"

# the dataset page has Dataset markup
curl -s "$PAGE" | grep -o '"@type": *"Dataset"' | head -1
```

Then three steps that a command cannot check:

1. Run the dataset page through Google's Rich Results Test: it will show errors in the Dataset markup.
2. Add the site to Google Search Console and submit the sitemap. It shows how many pages the search engine has indexed and why it skipped the rest.
3. A few weeks later, search for five to ten datasets by title in Google Dataset Search. If copies from aggregators come up rather than your pages, check `sameAs` and that the dataset page opens without a login.

A short order of work

The address, schema.org, the sitemap and a dump in the right profile are already in the requirements checklist. Here they are joined by registering the catalog in a relevant registry, submitting the sitemap to Search Console and a check a month later: the dataset is found by search, the dump opens.

Sources

- Dataset markup in Google's documentation: the canonical dataset page, `sameAs`, the sitemap
- ckanext-dcat: Google Dataset Search; third-party ckanext-sitemap extensions: a GitHub search, 11 October 2026
- The sitemap in Dataverse, schema.org export in Dataverse
- Sitemaps in InvenioRDM, release v13
- GeoNetwork 4.4 and search engines: `robots.txt` inside the catalog address, advice to register the sitemap by hand; RFC 9309: `robots.txt` is read only from the site root; the absence of a root `robots.txt` at geodata.nz, geocat.ch and metadata.geoportaal.ee: a check on 12 October 2026
- Rosstat's "Compendia" section: a web archive snapshot of 13 February 2026 (the list is loaded by a request to `/compendium/getPage`); the unavailability of rosstat.gov.ru and ach.gov.ru from a foreign server and the HTML of a data.europa.eu dataset page: a check on 12 October 2026
- Web Data Commons: schema.org subsets, 2024 release and 2023 release

- Dataset markup on catalog.data.gov: a check of a dataset page on 12 October 2026
- JKAN, DKAN, InvenioRDM
- I. Begtin, Telegram (in Russian): Rosstat and Ajax, 16 November 2022; the Accounts Chamber archive, 30 November 2022; GeoNetwork and search engines, 1 April 2023; the quality of Dataset markup in Common Crawl, 20 June 2024; Dataset and DataCatalog in schema.org, 15 September 2024; how to get into the Dateno index, 22 October 2024; Russian catalogs in Dateno, 15 August 2024; geoblocking and AI agents, 14 July 2025
- Registering a repository in re3data
- llms.txt, Model Context Protocol
- Metadata Export
- The dataportals-registry export, release 1.27.0 of 15 October 2026. A sitemap is marked for about 7,000 catalogs.

PART VIII.

Publishing Without a Portal

Static sites, Git repositories, cloud storage and shared hubs: when they are enough and how to make such a publication findable.

1. When You Don't Need a Portal
2. Static Sites
3. Git Repositories
4. Cloud Storage and Shared Hubs
5. Making a Publication Findable

When You Don't Need a Portal

You need your own portal when an obligation cannot be met without it, or when there are already so many datasets and publishers that a page and a description file no longer cope. Until then, data is published in someone else's catalog, on your own website, in a Git repository or on a shared hub. Both small groups and federal agencies do this: the US national catalog harvests a description file from an agency's website and leaves the data files where they are.

An empty installation promises a catalog that nobody is there to maintain. Descriptions go stale. If registration is open, it collects spam; this happens with CKAN portals, see the chapter on access and roles. When the hosting contract ends, the addresses disappear. A page with a dozen datasets makes no such promise.

What this chapter calls a portal

A catalog is any place where a dataset can be found, understood and obtained. A static site can be one. A portal in this chapter is your own application: a database of descriptions, search, accounts, roles and a server that someone updates for years. A dataset list is a page or a file with links, without dataset pages or search. For a dozen datasets a list is fine. It becomes a dead end when a row has no licence, date or persistent address.

How a catalog is assembled from parts is in the chapter on catalog anatomy. How to turn a list into a findable publication is in the chapter on dataset lists.

Five questions before installing

The answers are written down before choosing a platform. A platform that covers the features is useless if nobody is there to maintain it. A comparison of build approaches is in the chapter "Build, Open Source or SaaS".

1. **Who must see the data.** A higher-level catalog, a search engine, a reader of an article, colleagues inside the organisation. The description file depends on the recipient: DCAT-AP, DCAT-US, schema.org or DataCite.
2. **How many datasets there are and how often they change.** Dozens of datasets and edits a few times a year are covered by a page or a repository. Hundreds of datasets arriving daily from a records system require loading into a portal or a script that updates files by itself. A script is still publishing without a portal.
3. **How many organisations publish and how they edit.** One group gets by with repository permissions and review of changes before merging. A portal is needed when there are many publishers who do not work with Git, or when the flow of requests for a closed file with an open description can no longer be handled by email and manual approval on a shared hub.
4. **Who updates the server in three years.** If the answer is "the page author" or "nobody", it is too early for a portal. A position or a maintenance contract makes a portal possible, but does not cancel the first three questions.

5. **What exactly the higher-level catalog must harvest.** Usually it is one description file in the right profile. A search API is needed by people and programs when there are already hundreds of datasets. A harvester is satisfied with a file.

Answer	Where to publish
One group, dozens of datasets, the higher-level catalog accepts outside organisations	in that catalog, with no showcase of your own
You already have a website, and the higher-level catalog harvests a file	a description file on that website
You need edit history and collaboration	a Git repository and a page built from the same files
The data will be cited in an article, and there is no repository of your own	a general repository with DOIs, such as Zenodo
The file is large	cloud storage or a hub, with the description on your own page
Many publishers outside Git, a flow of requests for closed files, harvesting other catalogs	your own portal, or a standard portal already handed out to public authorities

The last row does not mean "install CKAN tomorrow". A standard portal that a region or provider already runs for municipalities covers the roles without an installation of your own. The analysis for a city is in the municipality example.

Where to publish

Five approaches. They are combined: a repository holds the tables, a website shows them to people, and Zenodo issues a DOI for a release.

Approach	When it is enough	What you will not have	More
Someone else's catalog	few datasets, the catalog already accepts yours	your own domain and your own fields	build approach
A section of your own website	the site exists, there are few datasets	an "add dataset" form and roles for outside publishers	below, using a `data.json` file as an example
A static site	dozens of datasets, someone who edits files	API search and scheduled harvesting of other catalogs, until you write them separately	static sites
A Git repository	history and collaborative editing matter	large files in Git itself, and findability without a page	Git repositories
Cloud and a shared hub	the file is large, a DOI is needed, or the audience is already on the platform	a guarantee that the platform's rules will stay the same	storage and hubs

The minimum without which none of the approaches works: a persistent address, a licence, a date and a contact, schema.org markup. A file with all descriptions is added when there are several datasets or a harvester takes them: from that point the publication becomes a catalog, and the common minimum applies to it. How to build it step by step is in the chapter "Making a Publication Findable". Which capabilities the map considers mandatory, and how to cover them without a portal, is in the capability map.

A file on your own website instead of a portal

data.gov does not store agencies' files. An agency publishes a metadata catalog, usually `data.json` in the DCAT-US profile, at a persistent address on its own website. The harvester takes this file on a schedule, and the records appear in the common catalog. In the GSA instructions the source is a public file, the source type is "document", and the frequency is from daily to monthly. Under the harvester's rules a source is harvested at least once a month.

On 13 October 2026 the list of harvest sources showed 793 records. 135 of them have the "document" type: the harvester takes one file. For 117 the address points to `data.json`. Another 577 belong to the Census Bureau: these are folders of geographic descriptions sliced by boundary type, not separate agencies. For a small agency the conclusion is direct: one file is enough.

A live example is the US Election Assistance Commission. Its website is built on Drupal 10. The commission's data catalog is the file eac.gov/data.json. On 13 October 2026 it contained 11 records in the DCAT-US 1.1 schema, and the data.gov harvester takes it once a week as the `eac-data` source. That is enough for the descriptions to reach the national catalog.

Harvesting also takes a weak description. The records have no `license` field. In the `rights` field, the DCAT-US 1.1 schema expects a short text about access conditions, while some records have the boolean value `true` and some have no field. Some download addresses point to a section of the site rather than a file. Some records are PDF reports. A file instead of a portal covers the harvesting task. The licence, a direct link and separating a dataset from a report are up to the author.

In Europe it is the same technique: data.europa.eu harvests a catalog in DCAT-AP if the publisher has submitted it for harvesting. Your own CKAN is not required for this. How the chain works is in the chapter on federations.

Three places at once

The Global Jukebox research project publishes data on song traditions without a portal of its own. The site theglobaljukebox.org shows the records to people. Tables and code live in the repositories of the theglobaljukebox organisation on GitHub. The release of the Cantometrics dataset that people cite is archived in Zenodo: DOI 10.5281/zenodo.4898406, version v0.1-alpha of 3 June 2021. On 13 October 2026 both the site and the Zenodo record opened.

The combination works as long as the three places have the same description. Here it diverged already in the release that got the DOI. The `LICENSE.txt` file in the v0.1-alpha tag is the CC BY 4.0 licence. The `.zenodo.json` file in the same tag specifies CC BY-NC 4.0, and that is what the Zenodo record shows. Someone who cites the DOI and someone who downloaded the repository get different terms. The licence is chosen once, written in the place considered primary, and copied unchanged into the others.

Example: an agency with twenty datasets

Context. A small agency. About twenty datasets: quarterly dumps and annual reports. A website on an ordinary CMS already exists, run by one webmaster. The national catalog harvests a description file in a known profile (DCAT-US or DCAT-AP) and also accepts organisations inside itself. There is no portal team.

Options. Its own CKAN. A cloud portal. An organisation only on the national catalog, with files staying on the agency's website. A `data.json` file on the existing website, registered for harvesting.

Criteria. Get into the national catalog. Every dataset has a licence and a direct link to the file. The address of the description file does not change when text on the site is edited. Maintenance fits into one webmaster's work. Its own showcase on the existing website matters more than a separate portal: readers look for data where they look for the agency itself.

Decision. A description file on the existing website and a harvesting request. If the national catalog does not harvest external files and only accepts organisations inside itself, publish there and do not install a portal.

Why. Its own CKAN with one webmaster will be left without core updates: it loses on the maintenance criterion. A cloud portal would cover search and dataset pages, but the fee and the address migration do not pay off for twenty datasets. An organisation only on the national catalog also gets harvested and is even simpler to maintain, but the agency loses its own showcase. A file on its own website keeps the showcase where it already is and gives the harvester the profile it expects.

Consequences. The agency depends on the harvesting profile and schedule. An error in the file is visible only after the harvester runs, so the file is checked with a validator before the request. There is no dataset search of its own: with twenty records a page of links is enough.

When to review. More than two hundred datasets, several departments with different rights, the national catalog stops accepting the file, or some files need to be closed while the description stays open and requests exceed what one person can handle.

How to write such a record for yourself is in the chapter "Justifying a Decision".

When you do need your own portal

- Many organisations publish datasets, these people do not work with Git, and each needs its own rights and review before publication.
- The description is open, the file is released on request, and there are so many requests that a shared hub and email cannot cope. This is how microdata catalogs work.
- Other catalogs need to be harvested regularly and copies moderated. A script in a repository, as at opendata.scot, pulls thousands of descriptions, but it does not provide roles.
- Datasets arrive daily from a records system, and a script that updates files is harder to maintain than loading into a portal.
- The country's profile requires a capability that neither a file nor someone else's catalog that already accepts your datasets has. More often what is required is precisely a file in the right profile.

For certain catalog types the threshold is different. A research group with one dataset for an article does not need its own repository. Nor does a city with thirty datasets.

Task	Where it is analysed when you don't need your own catalog
A public authority's open data	open data portals
Maps and layers	geoportals
Data for an article	research repositories
Indicators	indicator catalogs
Microdata	microdata catalogs
Datasets for models	ML catalogs
Reference lists	metadata catalogs

What you pay

Approach	What you get	What you pay
Someone else's catalog	dataset pages, harvesting and search already work	someone else's fields, someone else's schedule, no domain of your own
A section of your website	cheap, the address is in your domain	manual edits; API search does not appear
A static site	almost nothing to break, the site can be taken away as files	edits through files; people without Git will not manage an "add dataset" form
A Git repository	history and collaboration	a size limit: GitHub does not accept files larger than 100 MiB and advises keeping a repository under 1 GB; without a page with markup the dataset is hard to find
Cloud and a hub	large volumes, DOIs, someone else's audience	the platform's rules and limits change, and the platform may close

Prices of other platforms are compared in the chapter on the build approach. The history of closed hubs is in the chapter on cloud storage.

What live lists show

In the dataportals-registry (release 1.27.0 of 15 October 2026), 107 catalogs are recorded as dataset lists: a page, a repository or a storage listing, without dataset pages. 15 have a programming interface. 12 have a sitemap. 95 are recorded as custom-built. The other twelve are pages on ordinary website engines: Drupal, WordPress, TYPO3, Jekyll, SharePoint and Liferay. 7 are defunct or withdrawn.

A small share of interfaces is expected for a list. A list of twenty links does not need a search API. What breaks is something else: a row has no licence, the page address changes when the site is redesigned, there is no schema.org markup, and links die silently. How this was measured on a thematic list and how to fix it is in the chapter on dataset lists. The registry hardly records licences and schema.org markup, so these gaps are not visible in its figures.

How to check your decision

- [ ] The recipient is named: who harvests the descriptions and in which profile.
- [ ] The datasets, publishing organisations and update frequency are counted.

- [ ] The person who will update the publication in three years is named. If it is a portal server, whoever updates the server is named too.
- [ ] Every dataset has a licence, a date and a direct link to the file, not just a section page.
- [ ] The file with all descriptions opens without a login and passes the check of the profile the harvester expects.
- [ ] The event that reopens the decision is recorded.

Commands for checking the file and the markup are in the chapters on metadata export and on findability.

Common mistakes

A portal "like everyone else's". A city or agency installs CKAN because the neighbour did. Two years later the core has not been updated, and there are as many datasets as would fit on one page. First answer the five questions above.

A link to a folder. A shared drive or a bucket without a description page is not a publication. The address is temporary, there is no licence, and a year later nobody can find the folder.

A home-grown description file format. To harvest sites, people invent a CSV table with their own columns. The national catalog's harvester does not read it. The file is made in the profile the higher-level catalog already accepts. How this turned out with the methodological recommendations in Russia is in the chapter on anti-patterns.

Waiting for someone to build a portal. While the national or regional catalog does not accept datasets, a page and a repository are already publishing data. The community in Scotland built opendata.scot this way, without waiting for a government portal. Details are in the chapter on static sites.

Three copies, three licences. A site, a repository and a hub describe one dataset differently. The reader does not know which copy is primary. One is designated primary, and the others are checked against it.

Sources

- GSA, How to get your Open Data on Data.gov: data.gov does not store files, the agency publishes `data.json`; Harvester API: a source is harvested at least once a month; the list of harvest sources: 793 sources, 135 of them of the "document" type, 117 `data.json` addresses and 577 Census Bureau sources, checked 13 October 2026
- eac.gov/data.json: 11 records, DCAT-US 1.1 schema, the `rights` field and download addresses; the commission's site responds with a Drupal 10 header. Checked 13 October 2026. The `eac-data` source in the harvest list is a weekly document. The `rights` field in the DCAT-US 1.1 dataset schema is text up to 255 characters or empty; `license` is an optional link
- Global Jukebox: site, organisation on GitHub, DOI 10.5281/zenodo.4898406 (Zenodo record, CC BY-NC 4.0 licence, version v0.1-alpha of 3 June 2021). In the same repository tag, LICENSE.txt is CC BY 4.0, while .zenodo.json specifies CC BY-NC 4.0. Checked 13 October 2026

- Large files on GitHub: a 100 MiB limit per file and a recommendation to keep a repository under 1 GB
- The dataportals-registry export, release 1.27.0 of 15 October 2026: 107 dataset lists, an interface for 15, a sitemap for 12, custom-built for 95, defunct or withdrawn 7. Queries of 15 October 2026 are in the appendix
- How to get harvested by data.europa.eu

Static Sites

A static site is a set of ready-made pages. A generator builds them from files, most often from a Git repository, and the hosting only serves the result: GitHub Pages, GitLab Pages and others. There is no database and no application running around the clock. An already built site survives a pause in the generator's development, and taking the catalog home means copying the files.

This is enough when a small group publishes the datasets, and the reader and the harvester need a dataset page and one description file. This is how opendata.scot works: on 14 October 2026 its description file held 2,596 datasets. Roles, a queue of requests for a closed file and a search API are a reason to change the publishing approach. A server set up next to the generator does not solve these tasks: the result is a home-made portal that nobody is there to maintain.

The order "repository, markup, DOI, record in someone else's catalog" is in the chapter "Making a Publication Findable". Here: what the site itself is built from, and where ready-made tools stop.

When it is enough

Situation	Where to go
Dozens to a few thousand open datasets, edited as files by one or two people	a static catalog: JKAN or your own generator
Few datasets, the agency's website already exists, the harvester needs one file	a description file on that website. See "When You Don't Need a Portal"
Sustainable Development Goal indicators, people need a showcase	OpenSDG
Satellite images and other spatial files read by a program	a static STAC catalog in storage. See geoportals and cloud
One large table that needs a page and queries	Datasette. This is already a small server, not a folder of files
Many publishers without Git, closed files, a search API	a portal or someone else's catalog

How it is built

The sources live in Git: a file per dataset, a page template, licences. On every change the generator builds HTML, a common description file and `sitemap.xml`. The hosting puts this folder at the site's address. The reader gets ordinary pages. The harvester takes one file instead of clicking through buttons.

Harvesting other catalogs is possible in this setup, but it is a script next to the site, not a generator feature. On a schedule, the script downloads descriptions, puts them as files into the repository, and the generator rebuilds the pages. opendata.scot is made this way. The finished site still crawls nothing and moderates no one: it contains whatever files the script managed to put there.

On GitHub Pages this setup has a ceiling. As of 14 October 2026 the documentation says: a published site is no larger than 1 GB, a build is cut off after 10 minutes, the recommended limit for the source repository is also 1 GB, and the soft bandwidth limit is 100 GB a month. On

opendata.scot the data files live with the publishers, and the repository holds the descriptions, so the site fits within this limit. A catalog that holds the tables themselves will hit a gigabyte before it hits a number of datasets. Other hosting has its own limits; read them for that hosting before publishing.

A page must contain the description text in HTML. A site that draws the dataset page only with a script in the browser is empty for a harvester and a search engine. How to check this is in the chapter on portal discoverability.

What to build with

Tool	When it is used	What it does not ship with
JKAN	open datasets, from dozens to a few thousand	a search API, roles, schema.org; the description file is shorter than the schema it names
An ordinary generator (Hugo, Jekyll, Astro)	few datasets, a site template already exists	a ready catalog: the page, markup and description file are written by you
PortalJS	your own interface on top of files or an existing catalog	roles and a search API in the free build from files
OpenSDG	a showcase of Sustainable Development Goal indicators	a live SDMX service and a general catalog of arbitrary files
A static STAC catalog	spatial files read by a program	an answer to the query "all files for this territory for these dates"
Datasette	one SQLite database	being static: the page is alive only while the process is running

In the dataportals-registry, release 1.27.0 of 15 October 2026, there are 48 open data portals on JKAN, 72 catalogs on OpenSDG and 44 open data portals on Datasette. These are the same numbers as in the chapters on open data portals, indicators and in the JKAN profile.

An ordinary generator needs a page template, a schema.org block from the ready-made file and a `data.json` or `datapackage.json` file alongside. A single dataset in Git is analysed in more detail in the chapter on repositories. A generator is chosen when there are few datasets and JKAN with its schema only gets in the way. When there are dozens of datasets, a ready catalog saves you the template.

JKAN

JKAN is a catalog on Jekyll, MIT licence. A dataset is a Markdown file in the `_datasets` folder. The page address comes from that file's path: editing the title does not change the address, renaming the file does. Alongside, an organisation page and a common `data.json` are built.

The file declares the DCAT-US 1.1 schema with a `conformsTo` line. The template fills a fixed list of fields: title, description, category, contact, licence and file links. A field added to the site schema is visible on the page. It reaches `data.json` only if it is separately written into that file's template. The DCAT-US 1.1 schema also requires keywords, a modification date, a publisher, an identifier, an access level and, for US federal bodies, bureau and programme codes. The template does not output them. It tries to take the publisher and loses it: in the template on the `main` branch, the

variable the publisher is written to and the variable it is read from have different names. The `conformsTo` line is always set nonetheless. This file is not served to a harvester that needs DCAT-AP until the template is reworked. The municipality scenario says the same.

The page template has no schema.org markup. There is RDFa: the page block carries `dcat:Dataset`. Google's documentation says it reads such DCAT on a page too, but the Rich Results Test does not accept the DCAT example. schema.org from the book's file is added to the template: the test checks this markup before indexing. How the fields work is in the chapter on findability without a portal.

Search on the site is an in-browser filter over the `datasets.json` file. This search has no separate API. The common file for harvesters is `data.json`.

There is an editor form: Decap CMS, at the `/editor` page. A person logs in with a GitHub account, and an edit becomes a commit. The form provides no roles and no request queue. On Netlify, login is configured at the hosting. On GitHub Pages, the JKAN instructions call for a separate small login server. That is already a process someone has to keep running: for the sake of a form on free GitHub pages, the static site stops being just files. Those who are happy to edit Markdown do not enable this server.

The numbered release is 2.0.0 of 30 March 2023. As of 14 October 2026 there is no later number. The `main` branch lived on after the release. The sitemap, via the `jekyll-sitemap` plugin, was added on 29 July 2026 and is not in the 2.0.0 tag. The branch's last push was on 7 September 2026. In 2018–2020 there were commits, mostly on styling. In 2021 there were none. At the end of 2022 the built-in editor was replaced with Decap. An already built site does not fall over during such a pause. The template goes stale: a copy from the 2.0.0 tag will not get the sitemap until it is updated.

PortalJS

PortalJS is a toolkit on Next.js, MIT licence. According to the documentation as of 14 October 2026, the free build reads CSV and JSON from project files by default and can export a static site to a folder. At build time the same interface can fetch pages in advance from an already running CKAN. While the build is static, the hosting again serves files, and new datasets appear after the next build. A mode in which the page asks the server every time requires a host with Node, not a folder of files.

Paid hosting is a separate product. Pricing on the same day: 1,188, 3,588 and 14,990 dollars a year. Roles are available from the first paid plan. API search and managed CKAN start from the second, at 3,588 dollars a year. This is a portal with a subscription, not a way to make do with files. A price comparison is in the chapter on the build approach. The "ready catalog with your own site on top" setup is there too, in the section on a ready core.

OpenSDG

OpenSDG is a static site for Sustainable Development Goal indicators, MIT licence. Release 2.4.0 came out on 19 August 2025. Indicator data and texts live in a separate repository, a build program prepares files from them, and Jekyll builds the pages. This is how the UK SDG indicators work: on 14 October 2026 the page code included the `jekyll-open-sdg-plugins` plugin.

SDMX here is a source the build can read, and SDMX-ML files it can put on the site if this is switched on. The site does not run a live SDMX service that Eurostat or the IMF would query. A national office needs PxWeb or .Stat for exchange, and OpenSDG for a public showcase. The choice is set out in the chapter on indicator catalogs.

A static STAC catalog

A static STAC catalog is a tree of JSON files. At the root is `catalog.json`, then collections and items with links to images, rasters and other spatial files. The specification's best practice is direct: these are files on a web server, often in object storage; the catalog can be walked by its links, and it does not answer queries. The file names are `catalog.json` and `collection.json`. Links within the tree are made relative if the catalog must be movable.

As long as a program reads the catalog, no separate catalog server is needed. STAC Browser opens both such a tree and a STAC API. The tree cannot itself run the query "all items in this rectangle for these dates": for that, the files are loaded into an API. A live example of a tree is the public Google Earth Engine catalog, the file `catalog.json`. On 14 October 2026 it was a document of type `Catalog` with 133 child links, 25,841 bytes. There is also an HTML page for the same datasets; the JSON is for programs.

Datasette

Datasette publishes an SQLite database: a web page and a JSON API appear with one command. Apache-2.0 licence. It is a neighbour of static sites, not one itself: the process must be running, otherwise there is no page. Maintenance is simpler than a portal, but it is still maintenance of a process.

There is also Datasette Lite: the same program in the browser, via WebAssembly. The hosting serves the page and the database file, and the reader loads the database locally. This is how you publish a single table that can realistically be opened in full. A catalog of thousands of files is not put into a browser.

Two live catalogs

Two working examples show that a static catalog is not only for a small collection: one is a national portal, the other was built by a community with no government budget.

The Czech national portal

data.gov.cz has two different products at one address. The information part (articles, instructions, standards) is built with Jekyll from the site repository and published via GitHub Pages. The README as of 14 October 2026 says the same. Text edits are proposed with a pull request, as in any open project.

The dataset catalog is a separate application. The README points to it explicitly: the national catalog's interface lives in another repository, LinkedPipes DCAT-AP Viewer. Descriptions sit in a linked data store, and a SPARQL query endpoint is open to it. Linked data is a way of publishing descriptions so that a record and a value each have their own address and can be referenced from another system.

The portal does not store dataset files. On 14 October 2026 SPARQL counted 31,272 datasets from 133 catalogs and 39,624 distinct download addresses. Of these, 16 pointed to data.gov.cz itself. These are the catalog's own materials: lists of schools, libraries and sports grounds, a catalog dump, link quality reports. The other addresses are publishers' sites. The price of this lightness is that the portal will not keep a copy if an agency deletes a file. Why this matters is in the chapter on storage.

The number of datasets changes more than the structure. On 24 February 2023 there were about 142,000, of which about 137,000 came from one cadastral agency. By October 2026 the SPARQL endpoint had 31,272. The catalog is still an index, not an archive.

A catalog built by the community

For a long time Scotland had no common government open data portal, although cities, agencies and universities had their own. Volunteers built opendata.scot: a JKAN site, with descriptions and harvesters on GitHub. Scripts take descriptions from portals on CKAN, ArcGIS and other platforms and put them as files into the repository. From July to 9 October 2026, synchronisation ran roughly once a week. The last such commit was on 9 October, and the file on the site was updated in the same minute.

On that day, 14 October, `data.json` held 2,596 datasets. The file is 5,878,536 bytes and declares the DCAT-US 1.1 schema. Each record has six fields: title, description, category, contact, licence, links. None has a publisher. There are no keywords, modification dates, identifiers or access levels either. Yet the organisation is on the dataset page: the block carries a publisher in RDFa. A harvester that reads only `data.json` will not see it.

The licence in the file is filled in and not always usable. For 1,407 of 2,596 it is a link to the Open Government Licence 3.0. For 462 the field says `No licence`: the cell is not empty, but the data cannot be used under it. 232 datasets have no file link at all. The files themselves live on the websites of cities and agencies. Among the most frequent addresses are portals on ArcGIS Hub.

The dataset page, opened the same day, contains no schema.org. It has `dcat:Dataset`, as in the JKAN template. The sitemap opens. Search on the site reads `datasets.json`, 5,775,288 bytes. There is no separate search API.

The site has an analytics section: which organisations publish data and which platforms their portals run on. Page hosting is GitHub Pages. A catalog server is not needed as long as the description file and pages are satisfactory. Later the Scottish Government built its own data search engine. It is analysed in the chapter on search engines and aggregators.

What to do on the site

- The dataset page address does not change when a typo is fixed. In JKAN, do not rename the file for that.
- The page HTML contains schema.org/Dataset. RDFa with `dcat:Dataset` does not replace it.
- One file with all descriptions opens without a login. A record has a publisher, a licence, a modification date and a direct file link. The `conformsTo` line is checked against the fields.

- The licence field contains a link to a well-known licence or an explicit text of terms. The value `No licence` does not count as a licence.
- There is a `sitemap.xml`.
- The description text is in the HTML, not appearing only after a script runs.

Without these items the site remains a list of links. With them it covers the minimum from the checklist. How to take one publication through to a search engine and someone else's catalog is in the chapter on findability.

When a static site is no longer enough

It is time to change the approach when there are more publishers without a Git account than the team can handle through pull requests. The same applies when a description is open, a file is closed, and requests can no longer be handled by email. A search API is needed when there are so many datasets that a common JSON becomes heavy for every program, or when a harvester accepts only an API under the country's rules. For EU high-value datasets the regulation requires both an API and bulk download: a description file covers the metadata dump but does not replace a data API. More in the chapter on access interfaces.

A build that does not fit within the hosting limit is the same threshold. On GitHub Pages that means a site larger than 1 GB or a build longer than 10 minutes. Beyond that, a portal is taken, or the files are moved to storage and the site keeps the descriptions.

Example: a catalog run by two people

Context. A city library publishes about 80 open datasets: lists of institutions, timetables, annual tables. The files stay on agencies' websites. The regional portal does not accept outside organisations. Two staff members edit, both with GitHub accounts. There are no closed files. The description must be findable in a search engine. If a national harvester appears later, it is given one file.

Options. Its own CKAN. Cloud PortalJS with roles. JKAN. An ordinary generator and a `data.json` file built by a script.

Criteria. Each dataset has its own page, whose address does not change when the title is edited. The description file has a publisher, a licence, a date and a file link. The page has schema.org. Maintenance needs neither an application server nor an annual fee. Both editors manage on their own.

Decision. JKAN, with files hosted on GitHub Pages. Before the first publication, schema.org is added to the template, and the publisher and modification date are added to `data.json`. The Decap form is not switched on: it is easier for two people to edit files than to keep a separate login server. If the harvester, when it appears, expects DCAT-AP, the template is reworked for that profile in advance.

Why. CKAN adds a server for two editors. Cloud PortalJS on the first paid plan gives roles that are not needed here and costs from 1,188 dollars a year; its search API starts from the next plan. An ordinary generator will also build pages, but the dataset page and `data.json` would have to be

written from scratch. JKAN has them. The template, however, is not ready for publication while the file has no publisher: opendata.scot shows this.

Consequences. Search on the site filters JSON in the browser. A program is given the whole `data.json`. The library does not copy data files: a file disappearing from an agency's site shows up as a broken link. A person without a GitHub account cannot add a dataset.

When to review. A third publisher appears who will not work with GitHub. It becomes necessary to hide a file while keeping the description open, and requests exceed what email can handle. The regional portal starts accepting the library as an organisation: the two catalogs are reconciled, and one is kept if duplicates get in the way. The build hits 1 GB or 10 minutes on GitHub Pages.

How to write such a record for yourself is in the chapter "Justifying a Decision".

How to check

The commands below repeat the check of 14 October 2026. The expected answers were taken from the live opendata.scot. After the template is edited, the list of fields will be different.

```
# The description file opens without a login. On that day: 2,596 datasets,
# DCAT-US 1.1 schema, the first record has no publisher key
curl -s "https://opendata.scot/data.json" | python3 -c "import json,sys;
d=json.load(sys.stdin); print(len(d['dataset']), d['conformsTo']); print(sorted(d['dataset']
[0]))"

# The page has RDFa DCAT and no schema.org
curl -s "https://opendata.scot/datasets/aberdeen+city+council-academy+school+catchments/" |
python3 -c "import sys; t=sys.stdin.read(); print('schema.org', 'schema.org' in t);
print('dcat', 'dcat:Dataset' in t)"
```

The first command prints `2596`, the schema address `https://project-open-data.cio.gov/v1.1/schema` and a list of six keys without `publisher`. The second prints `schema.org False` and `dcat True`.

Then open your own site. In `data.json` a record has a publisher, a licence, a modification date and a direct link. The `conformsTo` line matches the profile the harvester expects: it is checked against the record's fields. The page HTML contains `schema.org/Dataset`. The `sitemap.xml` address responds. The dataset file, if you host it, opens via a direct link without a login.

The number of datasets in the Czech national catalog was counted on the same day like this. The answer was `31272`.

```
curl -sG "https://data.gov.cz/sparql" \
  --data-urlencode "query=SELECT (COUNT(DISTINCT ?d) AS ?n) WHERE { ?d a
<http://www.w3.org/ns/dcat#Dataset> }" \
  -H "Accept: application/sparql-results+json"
```

Sources

- JKAN, repository, architecture, installation and the `data.json` template on the `main` branch. Release 2.0.0, 30 March 2023. The sitemap: commit `b4733fb` of 29 July 2026, not in the 2.0.0 tag. The branch's last push: 7 September 2026. Commits by year: a pause in 2021, the editor replaced with Decap at the end of 2022. Checked 14 October 2026
- The DCAT-US 1.1 dataset schema: mandatory fields including publisher, modification date, identifier and access level. The schema address JKAN writes redirects here
- opendata.scot and repository: 2,596 datasets, 5,878,536 bytes in `data.json`, 5,775,288 bytes in `datasets.json`, six fields, publisher empty, 462 `No licence` values, 232 datasets without a file link, a page without schema.org. Synchronisation roughly weekly, the last run on 9 October 2026. Checked 14 October 2026
- The data.gov.cz site repository: Jekyll and GitHub Pages in the README; the dataset catalog is moved out to LinkedPipes DCAT-AP Viewer. SPARQL endpoint: 31,272 datasets, 133 catalogs, 39,624 download addresses, 16 of them on data.gov.cz. Query of 14 October 2026
- I. Begtin, Telegram (in Russian): the Czech open data portal, 24 February 2023: about 142,000 datasets, about 137,000 of them from the cadastral agency; opendata.scot on JKAN, 16 July 2022; Open Data Scotland, built by the community, 13 May 2023
- PortalJS: backends, deployment and pricing: static export, building pages from CKAN, plans at 1,188, 3,588 and 14,990 dollars a year. Checked 14 October 2026. Repository datopian/portaljs, MIT licence
- Open SDG, release 2.4.0 of 19 August 2025, MIT licence. Data sources and SDMX, SDMX output in configuration. sdgdata.gov.uk: the page includes `jekyll-open-sdg-plugins`, checked 14 October 2026
- STAC best practices on static catalogs: files, relative links, the catalog does not answer queries. STAC Browser. The Earth Engine catalog: catalog.json, 133 child links, checked 14 October 2026
- Datasette, repository simonw/datasette, Apache-2.0 licence. Datasette Lite: the database opens in the browser, last push 9 August 2026
- GitHub Pages limits: 1 GB per published site, 10 minutes per build, a recommended 1 GB per source repository, a soft bandwidth limit of 100 GB a month. Checked 14 October 2026
- JKAN, OpenSDG and Datasette installation counts: the dataportals-registry export, release 1.27.0 of 15 October 2026.

Git Repositories

A repository is needed when several people edit a small table and it matters who changed what. Git shows the history by itself. A dataset becomes a publication when a description of the columns, a licence and a page with a persistent address sit alongside it. If the table will be cited in a text, a release is fixed and gets a DOI. A file larger than the platform's limit is not put into Git itself: it is attached to a release or moved to storage, and the repository keeps its address.

The platforms are GitHub, GitLab and others. The figures below were taken from the GitHub.com and GitLab.com documentation on 14 October 2026. On your own server the administrator sets the limits, and they are checked again before publishing a large file.

When a repository is enough

Situation	Where to go
A table of up to a few tens of megabytes, edited by two or three people, row history needed	a repository and a page built from the same files
An annual snapshot will be cited in a report or article	the same repository plus a release with a DOI
The file is larger than the Git limit	cloud storage or a release attachment, with the description staying in the repository. See the chapter on cloud and hubs
Many publishers, and they do not work with Git	a portal or an existing catalog. See "When You Don't Need a Portal"
Only a DOI is needed, and nobody looks at the edit history	a general repository such as Zenodo, without your own Git

This chapter does not assemble the page, schema.org markup and a record in someone else's catalog step by step. The order is in the chapter "Making a Publication Findable".

What to put in the repository

Ready-made files are in the templates. They are copied, and your own names are substituted.

File	Why
`README.md`	what this is, for what period, how it is updated, who is responsible
`LICENSE`	the licence specifically for the data
`datapackage.json`	the list of files and the meaning of the columns
`data/`	the files themselves in open formats: CSV, JSON, Parquet
`CITATION.cff`	how to cite. On GitHub it produces a "Cite this repository" button

One dataset, one package. A package is a folder with a descriptor and files. The folder can be the whole repository or a directory inside it, if there are several datasets and each has its own descriptor. Ten unrelated tables in one repository without a descriptor are the same sheet of links, only in Git.

The descriptor is called `datapackage.json` and sits at the root of the package. This is what the Data Package 2.0 standard requires: the standard's page was updated on 25 August 2026. Another name is allowed inside your own system, but the standard advises against publishing a package under another name. In the book's template the file already has this name, and it has a `$schema` line with the value `https://datapackage.org/profiles/2.0/datapackage.json`. Without this line the descriptor is validated against profile 1.0. The `created` field in profile 2.0 is a date and time per RFC 3339, not just a calendar date. The template writes it that way.

The live Country Codes repository does not follow this rule: on 14 October 2026 the descriptor sat at the root as `datapackage.yml`. The columns are described in it and the CSV file is in place, but the published name is not the one the standard asks of a package. It can be seen as a model of a data dictionary and as a reminder to check the file name.

If a script sits next to the table, the code and the data often have different licences. They are written explicitly: the data's in `LICENSE` and `datapackage.json`, the code's in the script header or in a second file the README links to. A single phrase "licence in the root" leaves the reader guessing what it covers.

The script that builds the table also goes into the repository. Updates can be run automatically: on GitHub this is Actions, on GitLab it is CI. A missed quarter then shows up as a failed run.

Platforms and limits

Git stores text and small tables. A large file bloats the history: it is copied with every clone, even if it is no longer needed. Hence the limits.

	GitHub.com, free plan	GitLab.com, free plan
A file in ordinary Git	a warning if the file is larger than 50 MiB; rejection if larger than 100 MiB. Through the website, no more than 25 MiB. Separately, they advise keeping one object within 1 MB: a table of a few megabytes will go through, but cloning and viewing a change will get heavier	a new file of 100 MiB or more is not accepted. Through the website, no more than 100 MiB
The whole repository	better under 1 GB, strongly advised under 5 GB. They recommend keeping the `.git` folder within 10 GB. One push is no more than 2 GB	10 GiB per project, including Git LFS. One push is no more than 5 GiB. On a paid plan the project limit is 500 GiB
Git LFS	a file up to 2 GB. The month includes 10 GiB of storage and 10 GiB of downloads per account, not per repository. If the quota runs out and the overage budget is zero, LFS use is blocked until the next month	LFS space is part of the same 10 GiB per project. Beyond that the project becomes read-only until the size is brought back within the limit or space is bought
Release attachment	total volume and traffic are not limited. Each file is no larger than the LFS limit for the plan; on the free plan that is 2 GB	the GitLab.com settings table does not name a separate limit for release attachments. A file that does not fit into the project's 10 GiB is not put into this project

MiB and GiB are units from the platforms' documentation: 1 MiB = 1024 × 1024 bytes. The GitLab.com quotas page says 10 GiB, and the same site's settings table says "10 GB" for the same limit, the repository size including LFS.

Git LFS is an add-on to Git. A short pointer stays in the history, and the file itself is stored separately. It is used when a file must come along with a clone and does not fit into ordinary Git. Two caveats. GitHub Pages does not serve LFS files: a page built from the same repository will not show the large file. The download quota on free GitHub is 10 GiB a month for the whole account. A popular dataset can exhaust it, and a stranger will not get the file.

This is how geoBoundaries works: administrative unit boundaries are stored in Git LFS, which shows in the `.gitattributes` file (the `releaseData` and `sourceData` folders). The site and the API are separate repositories of the same project. On 14 October 2026 the `LICENSE` file named the CC BY 4.0 licence, the README next to it said "CC BY 4.0 / ODbL", and the GitHub card marked the licence as Other. Read the text in the file; the badge on the card does not replace it.

For a file that is downloaded whole once a quarter, a release attachment is simpler: on GitHub it has no monthly traffic quota. The repository then keeps the README, the descriptor and the attachment's address. Larger still means cloud storage; see the chapter on cloud and hubs.

A platform may change its rules or become inconvenient in your country. A copy of the release in Zenodo reduces this dependency. On GitHub the integration makes the copy by itself. GitLab.com has no such button: the release archive is uploaded to Zenodo by hand or by a script.

Why repositories are hardly ever found

A repository has no "data here" marker. To the platform it is the same as a code repository. In 2022, a file name search run from a logged-in account found tens of millions of CSV, JSON, XML and SQLite files on GitHub. That does not make it a catalog.

On 14 October 2026 code search without a token answered `401` with the text "Requires authentication". Repository search by topic, meanwhile, is open. A topic is a label the author sets themselves. That day the search counted 19,595 repositories with the `dataset` topic and 10,590 with `open-data` (the `incomplete_results` field was `false`). These are only the repositories whose author set the label.

People and programs tell a data repository apart by indirect signs: the `README.md` text, files at the root, `datapackage.json`. The `dataset` and `open-data` topics are still worth setting: it is free and helps those who search by topic. A page with schema.org markup is needed separately. Without it, a dataset search engine hardly sees the repository.

Release and DOI

A link to a file on the `main` branch opens today's state. Tomorrow the same link will have different rows. A release fixes the state. In practice it is a version tag, for example `v2026.1`, with an archive attached to it. A report cites the release, not the branch. How addresses and DOIs work is in the chapter on identifiers.

On GitHub, a repository is linked to Zenodo in the Zenodo profile. The integration takes releases created after it was switched on and gives each a DOI. A series of releases also has a concept DOI: it opens the latest version. A text cites the DOI of a specific release. Until the integration is set up, the `identifiers` block with an empty DOI is removed from the `CITATION.cff` template, so that

nobody cites a non-existent number. The DOI that Zenodo issued is written into `CITATION.cff` on the branch, for the GitHub button. There is no need to rewrite the release archive itself for this: the number is visible on the Zenodo page anyway, and a new release would become a new version.

`CITATION.cff` for data needs the line `type: dataset`. Without it GitHub will format the citation as a citation of software. The button reads exactly `CITATION.cff`.

The Zenodo record works separately. The integration's documentation sits in the section on software, and the supported `CITATION.cff` example is marked `type: software`. If there is a `.zenodo.json` at the root, Zenodo reads only that and ignores `CITATION.cff` for the record. The help article with this rule was published on 30 January 2026. For a dataset, `.zenodo.json` sets `upload_type` to `dataset` and the same title, authors and licence as in `CITATION.cff`. The licence code is written the way Zenodo knows it: `cc-by-4.0`, not the SPDX string `CC-BY-4.0` from the citation file. The meaning of the licence must match, not the spelling. A divergence between CC BY and CC BY-NC has already broken a publication without a portal; the example is in the chapter "When You Don't Need a Portal".

```
{
  "title": "Dataset title",
  "upload_type": "dataset",
  "license": "cc-by-4.0",
  "version": "2026.1",
  "creators": [
    {"name": "Surname, Given name", "affiliation": "Organisation"}
  ]
}
```

After the first release, open the record and check three things: the type is dataset, the licence matches the `LICENSE` file, and the DOI points to this release. A DOI is needed if the data is cited in text. It is not set up for an internal table.

GitLab has no automatic integration. The tag and archive are made there, the file is uploaded to Zenodo as a separate step, and the same three things are checked.

Formats for those who will use the data

In a repository it is easy to publish several formats of the same list. Whether the data gets used depends on this. Two repositories with a list of countries show the difference. Country Codes serves one CSV and a column dictionary. Country List serves an equivalent list in many languages, straight away in files a program can include without conversion. On 14 October 2026 the `data/ru` folder contained eleven files: CSV, JSON, YAML, XML, HTML, plain text, XLIFF, PHP and three SQL variants (MySQL, PostgreSQL and SQLite). The README lists the same formats. That day the first repository had 981 stars, the second 5,253.

Stars are not a licence assessment. Country List's `LICENSE` file is the MIT text, which speaks about software. Country Codes' descriptor states the ODC-PDDL-1.0 public domain dedication, and its README asks you to check ISO's restrictions on the country codes the list is built from. Someone who needs a file for a database looks at the format. Someone who puts the list into a product reads the licence in the file.

The conclusion is the same as before. A column dictionary is needed, and it is supplemented with the formats in which the data is actually used. First find out who those people are. An analyst expects CSV or Parquet, a developer JSON or SQL. How to choose a format, and why GitHub shows only CSV and TSV of up to 512 KB as a table, is in the chapter on formats.

Build code instead of data

Sometimes what is published is not a table but the program that builds it. This is done when the source publishes Excel, changes files without warning or gives no direct link. A ready table goes stale at the next update; a script can be run again.

In 2022 harvesters for the registers of accredited IT companies and domestic software from the website of Russia's Ministry of Digital Development were published this way. On 14 October 2026 the repository still holds build descriptions, not the datasets themselves: the `itregistry` folder has one file, `datacrafter.yml`, which says where to get the Excel file and how to name the columns. The repository's README is one line: "Datacrafter examples".

The script lives as long as the source lives. If the agency removes the file or changes the address, the code will stop building the table, and you will be left without a copy. So the result is published next to the code: a release with ready files or a copy in Zenodo. How to preserve data that may disappear at the source is in the chapter on archiving.

Collaborative editing

An edit is accepted through a merge request. On GitHub it is called a pull request, on GitLab a merge request: it is a proposal to change files that a second person reviews before it reaches the main branch. In a public repository on free GitHub you can make approval mandatory and assign reviewers with a `CODEOWNERS` file, for example for the `data/` folder. On GitLab.com's free plan, approval does not block a merge. How roles work and how plans differ is in the chapter on access. The commit history is the change log.

A large table cannot be read in the request window. GitHub shows a diff for one file up to 500 KB or 20,000 lines, and the whole request up to 1 MB or 20,000 lines. The check is then done by a program that validates the CSV against the column dictionary:

```
frictionless validate datapackage.json
```

The command is added to an automatic run on every change. On GitHub this is Actions. What exactly such a check catches is in the chapter on metadata completeness.

Versions of a table, not a file

Git compares lines of text. For a table of hundreds of thousands of rows this is not enough: a reviewer wants to see which cells changed, not two 50 MB files.

Dolt is a SQL database, compatible with MySQL, that you clone, branch, merge and push just like a Git repository. **DoltHub** hosts public databases free of charge. An edit goes through a pull request, and the difference is shown by rows and cells. The history can be queried with ordinary SQL: the

`dolt_log` table is the commit log. DoltLab is the same platform for self-hosting.

```
# The last two commits of a public database, without logging in
curl -s "https://www.dolthub.com/api/v1alpha1/dolthub/ip-to-country/master?
q=SELECT%20commit_hash%2C%20committer%2C%20date%20FROM%20dolt_log%20LIMIT%202"
```

On 15 October 2026 the answer began with `"query_execution_status":"Success"`. The branch in this database is called `master`. At the address with `main` the server answered "branch not found".

This is a step between a repository and a portal. Edit history as in Git, but for a table that no longer fits into a CSV of a couple of megabytes. The risk is the same as with any single-operator platform: in 2022 the QRI catalog, with a similar idea, closed down this way. So check in advance that the database can be taken home together with its history with the `dolt clone` command. Other innovations of the same kind are in the chapter on leading catalogs.

Example: a quarterly table for two

Context. Two agency staff edit an expenditure table once a quarter. The file is a CSV of about 2 MB. It must be visible who changed a row. Once a year the table is cited in a report. The national catalog does not harvest this organisation. There is no server of its own and no DataCite agreement.

Options. Its own portal. The file only in Zenodo, without a repository. A repository with a README and no column dictionary. A repository with `datapackage.json`, a licence and a page; the annual release goes to Zenodo.

Criteria. Edit history is visible without a server. The columns have names and types. The annual file the report cites does not change retroactively. Maintenance fits into these two people's work.

Decision. A repository on GitHub or GitLab, with files from the book's templates. The page is built from the same repository. Once a year a release tag is set. On GitHub, Zenodo takes the release. On GitLab, the archive is uploaded to Zenodo by hand. In both cases the record's type is dataset.

Why. A portal adds a server while there are still only two publishers. Zenodo alone does not show who changed a cell in March. A repository without `datapackage.json` leaves the table without the meaning of its fields. The `main` branch is no good for a footnote in a report: next quarter the same link will show different numbers.

Consequences. The DOI points to the copy of the release in Zenodo. A "table as of today" link goes to the page, and the report's link goes to this year's DOI. A file larger than the limit is not put into this repository. The licence in `LICENSE`, in the descriptor and in the Zenodo record is one and the same.

When to review. The file hits the platform's warning (on GitHub, from 50 MiB) or a rejection. The diff in a merge request can no longer be read: then look at Dolt. A third group appears that does not work with Git. The national catalog starts requiring a description file: it is put alongside, and a portal does not yet become mandatory because of it. The annual snapshot is cited more often than it is released.

How to check

The check takes a few minutes and does not require logging in to the platform. The expected answers below were obtained on 14 October 2026.

```
# The file opens without a login: code 200, length 133966 bytes
curl -sI "https://raw.githubusercontent.com/datasets/country-codes/main/data/country-codes.csv"

# The descriptor is in place. In this repository it is YAML; the standard asks for datapackage.json
curl -sI "https://raw.githubusercontent.com/datasets/country-codes/main/datapackage.yml"

# Search over file contents is closed without a token
curl -s "https://api.github.com/search/code?q=filename:datapackage.json"

# Repository topics are visible without a login. Look at the total_count field
curl -s "https://api.github.com/search/repositories?q=topic:dataset&per_page=1"
```

The first two answers are code 200. The third is code 401 with a message that authentication is required. In the fourth, `total_count` was 19,595 that day. The `last_modified` field in the Country Codes descriptor says `2023-09-25`, while the repository's last push as of that day was 1 October 2026. Check both the date in the descriptor and the file date: they have already diverged.

Then open your own repository. The package root has `datapackage.json`, `LICENSE` and `CITATION.cff` with the line `type: dataset` and no empty DOI. The file that will be cited opens at the release tag's address, not only at the branch's address. If Zenodo is switched on, the record's type is dataset and it has the same licence as `LICENSE`.

Sources

- Data Package 2.0: the `datapackage.json` name, the `$schema` property, page updated 25 August 2026. Profile: datapackage.org/profiles/2.0/datapackage.json. Validation: Frictionless, validating data
- About large files on GitHub: 50 MiB, 100 MiB, the advice on 1 GB and 5 GB, release attachments. Repository limits: 10 GB for the `.git` folder, 2 GB push, 1 MB / 100 MB object, the diff in a pull request. Git LFS: file size by plan, LFS does not work with GitHub Pages. LFS quotas: 10 GiB of storage and 10 GiB of downloads on the free plan. Checked 14 October 2026
- GitLab.com storage quotas: 10 GiB per project on the free plan, 500 GiB on a paid plan, including LFS. GitLab.com settings: the table gives the same limit as 10 GB, a 5 GiB push. The free plan file limit: 100 MiB. Checked 14 October 2026
- CITATION.cff on GitHub: `type: dataset`. The Citation File Format specification
- Zenodo: enabling a GitHub repository, CITATION.cff, `.zenodo.json`. Help article: when `.zenodo.json` is present, `CITATION.cff` does not affect the record; published 30 January 2026
- Rendering CSV and TSV on GitHub: 512 KB, comma and tab
- Code search on GitHub: login required. On 14 October 2026 an API request without a token returned 401. Topics: `dataset`, 19,595 repositories; `open-data`, 10,590; `incomplete_results: false`

- Country Codes: `datapackage.yml`, the ODC-PDDL-1.0 licence in the descriptor, the ISO caveat in the README, 981 stars, CSV of 133,966 bytes. Country List: eleven formats in `data/ru`, LICENSE is MIT, 5,253 stars. Checked 14 October 2026. The format comparison was suggested in I. Begtin's post Open data as a data product (in Russian), November 2024
- geoBoundaries and repository: Git LFS in `.gitattributes`, LICENSE is CC BY 4.0, the README says "CC BY 4.0 / ODbL", the GitHub card says Other. Checked 14 October 2026
- datacrafter-examples: on 14 October 2026 `itregistry` contains `datacrafter.yml`, not a dataset
- I. Begtin, Telegram (in Russian): GitHub as a data source, 24 June 2022; GitHub, the largest but poorly structured catalog, 8 April 2024; what GitHub lacks to become a data repository, 11 July 2024; how to tell a data repository apart, 16 September 2025; build code instead of data, 7 July 2022
- This book's templates: README, datapackage.json, CITATION.cff
- DoltHub, What is Dolt?: SQL, MySQL compatibility, cloning, branches and merging; SQL API. Checked 15 October 2026

Cloud Storage and Shared Hubs

A large file can be published without a portal of your own. For this, two tasks are separated: where the bytes live and where the page through which the file is found lives. An open bucket solves the first. A shared hub often solves both, but by its own rules and up to its own limit. A shared folder behind a link solves neither.

You need your own portal when there are many publishers or the higher-level catalog requires roles and harvesting of other descriptions. As long as there is one dataset and one group publishes it, a page, a bucket or a hub is enough. In any of these cases you need a persistent address, a licence and a copy outside the platform people will cite. If a hub is the only place where the dataset lives, its closure equals deletion.

Three places that get confused

A **bucket** is a container in object storage: Amazon S3, Azure Blob Storage, Google Cloud Storage or a similar system on your own server (MinIO, Ceph). Each file has its own key and an HTTP address. Space is almost unlimited. A bucket has no catalog: a list of keys is not a set of dataset pages.

A **shared hub** is someone else's platform where the author puts their dataset: Zenodo, Figshare, Hugging Face, Kaggle, Source Cooperative. A dataset page and search already exist. A data hub means something else: several catalogs of one organisation under a common sign. An ML hub is a kind of shared hub where models sit next to datasets. Zenodo accepts data from any discipline.

A **shared folder** is Google Drive, Dropbox, Yandex Disk and similar drives. The link lives as long as the account of whoever issued it. There is no licence and no persistent page address, and a dataset search engine usually does not see the folder. This is not how data is published.

	Open bucket	Shared hub	Shared folder
What the reader gets	a file at an address	a dataset page and a file	a file, if the link is still alive
Who pays for space and downloads	the bucket owner, unless the dataset is in a sponsorship programme	the platform, as long as the dataset is within its rules	the account owner
DOI	no, it is issued separately	at Zenodo, Figshare, Kaggle, and with a button at Hugging Face	no
Download without login	yes, if the bucket is open for public reading	usually yes; Kaggle serves the file after login	often only after login
If this place closes	the files disappear	the files and the dataset page disappear	it disappears with the account

When a portal itself keeps files in the cloud, the reader does not see it: they click "Download" on the portal page. This is different. A bucket or a hub is enough instead of a portal, but a description page is still needed. How storage works inside a portal is in the chapter on storage.

Open bucket

A bucket suits satellite images, dumps and models that do not fit into Git or a general repository. Next to the files you put what keeps the folder from being a dump:

- `README.md`: what this is, for what period, how it is updated, who is responsible;
- a licence file for the data itself, with the same text as on the page;
- checksums, in a `SHA256SUMS` file or in `datapackage.json`;
- a persistent HTTPS address. A temporary signed link is no good for an article: at the Humanitarian Data Exchange it lives for minutes; see the chapter on storage.

Large files are stored in formats that can be read in parts without downloading them whole. A table in Parquet. An image in Cloud Optimized GeoTIFF: a program fetches the window it needs over the network. A multidimensional array, climate data or model output in Zarr, a set of chunks with the same property. For images, a static STAC catalog is also put in the bucket: JSON files through which a program finds the frame it needs. No separate catalog server is needed for this; see the chapter on static sites.

The link from an article and from schema.org markup points to a page you run. The direct link field holds the file's address in the bucket. How to build the page is in the chapter "Making a Publication Findable".

Who pays for downloads

While a bucket is simply open for reading, the owner pays for storage and outgoing traffic. A popular dataset of hundreds of gigabytes can cost noticeable money precisely because of downloads, not space. This is calculated before the announcement, not after the first bill.

Some providers take this cost on themselves, but on application and for a fixed term.

The **AWS Open Data Sponsorship Program** pays for S3 storage and data transfer for accepted datasets. Participation lasts two years from the start date and then ends unless AWS renews it. If participation ends for reasons other than a breach of terms, you get at least 90 days to take the data away. After that you pay. The terms allow the programme to be changed or closed. If the application does not specify a licence, the data is considered published under CC BY 4.0. The terms were last updated on 11 September 2025. An accepted dataset is described by a YAML file in the Registry of Open Data on AWS repository: it is a list, not a portal with dataset pages. How such a list works is in the chapter on dataset lists. On 14 October 2026 the repository held 1,241 such descriptions.

Google Cloud hosts a short showcase of public datasets itself and pays for their storage. You do not submit your own dataset through a form: the documentation asks you to email the programme. Google's console and command line require login even for someone else's public dataset, while a direct API link does not. In BigQuery Google also pays for storing public datasets, while queries beyond the first terabyte a month are paid by whoever runs them. It is a convenient way to read other people's data inside the cloud, not to publish your own.

Requester Pays mode shifts the download cost onto the reader. For open data it breaks publication: the file is not served without a cloud account. On 14 October 2026, of 1,241 descriptions in the AWS registry, six datasets had at least one bucket marked this way, among them Landsat and NAIP imagery, USGS lidar and some Sentinel-2 buckets. Two more datasets, Common Crawl and New York taxi trips, state in their description that an AWS account is required. Check both the registry field and a live request: the `sentinel-s2-l1c` bucket is marked Requester Pays in the file, but the `GENERAL_QUALITY.xml` file opened without a key the same day. The `usgs-landsat` bucket rejected an anonymous request.

The Copernicus digital elevation model on AWS opened without login the same day. The `readme.html` file in the `copernicus-dem-30m` bucket responded with code 200, and a chunk of an auxiliary file with code 206: the server can serve byte ranges. The dataset's licence is written in the registry file.

Shared hubs

A platform fits when the audience you need is already there, when a DOI is needed and you have no repository of your own and no DataCite agreement, or when the file is larger than is sensible to keep on your own disk. The limits below are as of 14 October 2026. Research repositories with Dryad and Harvard Dataverse pricing are analysed separately, in the chapter on research repositories. Dataset cards, Croissant and empty descriptions of datasets for models are in the chapter on ML catalogs.

Platform	For which dataset	Limit	What it gives	What you will not have
Hugging Face	datasets and models whose readers are already on the hub	public space on a free account on a best-effort basis; private datasets 100 GB; one file no larger than 500 GB	a dataset card, loading from code, Croissant for some datasets, a DOI with a button	a guarantee of free terabytes; old files disappear if the history is squashed
Kaggle	the same, if the dataset fits the limit	200 GB per dataset and the same for all private datasets on an account; no more than 50 files at the root	a dataset card, notebooks, Croissant, a DOI	downloads without login
Zenodo	data for an article and a repository release	50 GB and 100 files per record; once, up to 200 GB on request	a DOI for every version	Croissant; acceptance of a dataset cut into records to get around the limit
Figshare	the same, when Zenodo is too small	20 GB free per account, files up to 20 GB, no more than 500 records	a DOI	free space for hundreds of gigabytes
Source Cooperative	large open arrays, most often geodata	in the platform's documentation, hundreds of terabytes; publishing by application	a page from `README.md`, download without login	DOIs, versions for citation, closed files

Beyond the first gigabytes of public space, Hugging Face asks for a paid plan or a storage grant: the platform says plainly that free space is given on a best-effort basis and that it protects itself from abuse. A file larger than 200 GB it asks you to split into parts; the hard ceiling for one file is 500 GB. The repository history is not an archive: space is freed by squashing history, and old files disappear. It makes sense to switch on a DOI before the dataset starts being cited: after that the record cannot simply be deleted. More in the chapter on identifiers.

The number of records on Zenodo is not limited. One large dataset cannot be cut into many records to get around 50 GB: the rules treat this as circumventing the limit. Search via the API does not return the whole catalog, no further than the 10,000th record of one query, and there is no full dump as a file. How to get the descriptions is in the chapter on access interfaces.

Figshare+ charges a one-off fee for volume beyond the free 20 GB: 1,225 dollars for every 250 GB, with files up to 5 TB. Universities on their own Figshare have different limits, which the repository administrator states. For a lab without such an agreement, 500 free records are the ceiling of a showcase, not a replacement for a university repository.

Source Cooperative as a model

Source Cooperative is a platform run by the non-profit Radiant Earth. It is worth looking at as a model of how to structure things. A dataset here is called a product: a folder in object storage, open for public reading, plus a `README.md` at the root that the site shows as the dataset page. The author decides what to write on the page. The platform advises including a description, a licence separately for the text and for the data, access examples, a contact and a DOI if one already exists. The platform itself does not issue DOIs.

A product can take up hundreds of terabytes. The platform's documentation names RapidAI4EO as an example over 100 TB, and Earth Genome as a publication of more than 60 TB of processed imagery. This cannot go on GitHub. On 14 October 2026 the platform was in beta: anyone can download without registering, and publishing is by application. The site's code is open. Radiant Earth runs the platform, Taylor Geospatial funds its development, and cloud space is contributed by AWS and Azure. New data is no longer put into Azure. Closing a file or charging for downloads is not yet possible: the documentation lists these as future features. A product can be removed from the site's search, but it stays open by direct link.

There are no versions for citation. Buckets keep previous copies of a file, but non-current copies are deleted after 90 days. This protects against accidental overwriting. These copies do not preserve the release cited in an article.

A file's address is persistent and opens without a key. For the data.gov archive published by the Harvard Law School Library Innovation Lab, the description is at `https://data.source.coop/harvard-lil/gov-data/README.md`: on 14 October 2026 it opened without login, and the response type was Markdown text. The collection itself and how it is packaged are analysed in the chapter on archiving.

The weak point of this structure is that each product has its own folder structure and its own code examples. You read the documentation every time. For a dataset of hundreds of gigabytes you would read it anyway, and here at least it sits next to the data.

The **Internet Archive** keeps a copy of a site and a file for a long time, but it is an archive, not a catalog with field search. It preserves large files poorly. How to use it, and why a snapshot of pages is not enough, is in the chapter on archiving.

What to choose when

Situation	Where to go
A table under 100 MB, edit history matters	a Git repository, see the chapter on Git
Data for an article, up to 50 GB, no repository of your own	Zenodo
A dataset for models, readers already on the hub	Hugging Face; Kaggle if the dataset is no larger than 200 GB
Hundreds of gigabytes and terabytes, files read in parts	Source Cooperative or a cloud sponsorship programme; the page on your own site
A DOI is needed for a whole large array, and the hub application was not accepted	Figshare+ or an organisational repository that takes such volume
Many publishers, roles, harvesting other catalogs	your own portal or an existing standard one, see "When You Don't Need a Portal"

Example: an image archive without a portal of its own

Context. A research group. One dataset of derived imagery, about 400 GB, in Cloud Optimized GeoTIFF, updated once a year. Readers open the windows they need from code and do not download the whole archive. The data will be cited in an article. There is no server of its own, no librarian and no DataCite agreement. The national portal harvests agencies' descriptions and will not host such an array.

Options. Its own portal. The whole dataset in Zenodo. Kaggle. Hugging Face on a free account. Figshare+. Its own public bucket without sponsorship. An application to Source Cooperative or to the AWS sponsorship programme, with the page on the group's website.

Criteria. The file opens in parts without login and without a bill to the reader. The article has a DOI for something reproducible. The group can take the data away if the platform closes. The traffic fee does not become a surprise. Maintenance fits into one person's work.

Decision. An application to Source Cooperative: the array is open, it is geodata, and it is read in parts. In parallel, a page on the group's website with schema.org markup and a direct link to the product. A documented sample of up to 50 GB goes into Zenodo, and its description says plainly that it is a sample and that the full array is at the page's address. The sample gets the DOI the article cites.

Why. 400 GB does not fit into Zenodo: the ceiling for a one-off increase is 200 GB, and the rules forbid cutting one dataset into records to get around the limit. Kaggle is limited to 200 GB. Free space at Hugging Face is not promised for such volume, and the platform's audience is models; nobody looks for this imagery there. Figshare+ for 251–500 GB costs 2,450 dollars one-off at October 2026 prices. That is the fallback if a grant requires a DOI for every file. Its own bucket without sponsorship leaves the group paying for every download. Its own portal adds a server and adds neither a file format nor readers.

When to review. The application is not accepted: calculate the traffic and apply to the AWS programme, or pay Figshare+ if a DOI is needed for the whole array. A second group of publishers appears that is not ready to manage files in the cloud. If AWS was chosen, less than six months

remain until the end of sponsorship and there is no renewal. Readers cannot open the file without a cloud account.

What this risks

A platform changes its pricing, limits or rules, and sometimes closes. This has already happened.

- **QRI**, a public data catalog, announced its closure in January 2022. The qri.cloud site stopped accepting datasets, it was due to be switched off on 1 April 2022, and users were asked to take their data in advance.
- **data.world** started as an open cloud catalog. Then the company bet on an enterprise catalog, ServiceNow bought it in 2025, and on 13 July 2026 the open community was closed. Community datasets can no longer be downloaded, and members' data is being deleted. The enterprise catalog kept working.
- **Microsoft** closed the Planetary Computer Hub on 6 June 2024: an environment where researchers computed in notebooks on the project's infrastructure. The data catalog and programming interface remained. In the same discussion Microsoft wrote that it continues to host more than 50 PB. Notebooks and files from personal hub folders had to be taken away by users themselves: on 4 September 2024 the remaining user data was deleted.
- **Zenodo** did not close, but in 2024 it limited the depth of a single search query. For anyone who wants to take all descriptions at once, this is also a change of rules.

Four habits are set up before uploading:

- the licence is written on the dataset page and in a file next to the data, in the same text, not "as is customary on the site";
- the release that is cited has a DOI or another copy outside the platform;
- you yourself run the address of the page you consider persistent, and the hub is named as the download location;
- before uploading, it has been checked that your datasets can be exported in full, not one by one, and that a stranger downloads the file without login.

The description stays open even when the file itself is large and lives in the cloud. If a provider pays for the space, the sponsorship term is written into the decision record: AWS's two years end on a known day, not "some day".

How to check

The check takes a few minutes and does not require a cloud account. The expected answers below were obtained on 14 October 2026.

```
# Open bucket: the description is readable without a key, response 200
curl -sI "https://copernicus-dem-30m.s3.eu-central-1.amazonaws.com/readme.html"

# The file is served in parts: response 206 and a Content-Range header
curl -sI -H "Range: bytes=0-100" \
  "https://copernicus-dem-30m.s3.eu-central-
1.amazonaws.com/Copernicus_DSM_COG_10_N00_00_E006_00_DEM/AUXFILES/Copernicus_DSM_10_N00_00_E
006_00_ACM.kml"
```

```
# Requester Pays: an anonymous request is rejected
curl -s "https://usgs-landsat.s3.us-west-2.amazonaws.com/?list-type=2&max-keys=1"

# The registry mark and live access can diverge: the file opened without a key
curl -sI "https://sentinel-s2-l1c.s3.eu-central-1.amazonaws.com/GENERAL_QUALITY.xml"

# On the hub the description sits next to the data and opens without login
curl -sI "https://data.source.coop/harvard-lil/gov-data/README.md"
```

In the first answer, look for code 200 and the content type. In the second, code 206: this is how formats read in parts work. In the third, the server refuses and says plainly that anonymous requests to a Requester Pays bucket are not accepted. The fourth request returned 200 that day, although the bucket is marked Requester Pays in the registry file: so trust a request without a key, not only a field in someone else's list. The fifth answer is Markdown text a few kilobytes long.

Then open the dataset page in a browser where you are not logged in to the platform. If the file did not download, for open data that is already the answer, even when the page looks good. The licence is checked in three places: the file next to the data, the hub's dataset page, and the page the link from the text points to. A divergence among these three has already broken a publication without a portal; the example is in the chapter "When You Don't Need a Portal".

Sources

- Registry of Open Data on AWS and the descriptions repository; on 14 October 2026 `datasets/` held 1,241 YAML files, and the Git API tree was not truncated. Requester Pays and AccountRequired marks were counted from these files the same day
- AWS Open Data Sponsorship Program, terms (updated 11 September 2025; participation term two years, and at least 90 days to take the data away) and onboarding guidance: the programme covers S3 storage and data transfer
- Cloud Storage public datasets and BigQuery public datasets: Google pays for storage, and you submit your own dataset by email, not by form
- Requester Pays buckets. Checked 14 October 2026: Copernicus DEM opens without a key and serves byte ranges; usgs-landsat rejects an anonymous request; sentinel-s2-l1c is marked Requester Pays in the registry, but the file opened
- Hugging Face storage limits: free public space on a best-effort basis, private datasets 100 GB, files no larger than 500 GB; Kaggle datasets documentation: 200 GB per dataset
- Zenodo size limits, help article published 11 April 2026, checked 14 October 2026
- Figshare: free account limits and Figshare+: 1,225 dollars per 250 GB, checked 14 October 2026
- Source Cooperative: documentation: beta, application, run by Radiant Earth; core concepts: product, `README.md`, RapidAI4EO over 100 TB; infrastructure: open buckets and deletion of previous copies after 90 days. Checked 14 October 2026. The data.gov archive: the harvard-lil/gov-data page and the README.md file
- QRI, Winding Qri Down, January 2022 (web archive copy): dataset intake stopped, the site was due to be switched off on 1 April 2022
- data.world, A fond farewell: the open community closed on 13 July 2026, checked 14 October 2026

- Microsoft, Retiring the Hub on June 6th, 2024: the hub closed on 6 June 2024, the data catalog remained, remaining user files deleted on 4 September 2024
- I. Begtin, Telegram (in Russian): Source Cooperative and data products, 25 August 2026; the closure of QRI, 19 January 2022; data.world as a catalog behind a login, 18 July 2025; the closure of the Planetary Computer Hub, 28 May 2024

Making a Publication Findable

A dataset is found through a page with a persistent address. A person reads the description and licence on it, dataset search reads the markup, a catalog harvester takes a separate file, and a link from a report leads to a release with a DOI. The repository, the Zenodo record and the record in someone else's catalog name the same dataset and point to this page.

Below is the order of work for one medium-sized dataset. How the platforms themselves work is in the chapters on static sites, on Git and on cloud and hubs. How a platform does the same for thousands of pages is in the chapter on catalog discoverability.

Who searches and what they read

One openly available file is not enough: each reader has their own way in.

Who searches	What they open	What is prepared for them
A person with a link	the page	description, licence, file, date, contact
An ordinary search engine	HTML and links from other sites	text on the page and a sitemap
Dataset search, e.g. Google Dataset Search	markup on the page	schema.org of type Dataset that can be run through a test
A harvester of a national or agency catalog	a file in the profile you agreed on	`data.json` or DCAT-AP, with that profile's fields filled in
The author of a report or article	a release that does not change retroactively	a version tag and the DOI of that release

The repository topics `dataset` and `open-data` help those who search for repositories on the platform itself. Dataset search does not read them. Why GitHub works this way is in the chapter on repositories.

One primary address

When there are several places, the dataset still has one primary address: the page you run. The other addresses rely on it.

Markup field	What it points to
`url`	the primary page
`identifier`	the release DOI, once the number is issued
`sameAs`	another page for the same dataset, e.g. the Zenodo record
`includedInDataCatalog`	a catalog in which the record has already appeared

Google asks for full markup on the primary page. If markup is on a copy too, the copy's `sameAs` holds the primary page's address. The same `sameAs` value on two different datasets glues them into one. Search addresses, the `main` branch and file listings are not put in `sameAs`: they are not a dataset page.

The page address is not built from the title. The title gets edited; the path stays: `…/bus-stops`. Old links from articles and the search engine's index remember the path, not today's name.

The `includedInDataCatalog` field is filled in after the harvester has published the record. A catalog name in the markup before that promises a record that does not exist.

What to put on the page

The markup is an `application/ld+json` block in the page HTML. Ready text is in schema-dataset.jsonld. It is inserted into the page template. In a README on github.com, the same text remains a paragraph for people: there is no `script` tag there, and dataset search does not read it.

Google's documentation, checked on 14 October 2026, requires two fields for the Dataset type: `name` and `description`. The description is 50 to 5,000 characters. Longer text is cut off at 5,000. A single short sentence often falls short of 50. The description says what is in the table, for what period and territory, who collected it and how often the file changes.

Google calls the other fields recommended. The test may pass without them, but the dataset cannot be used. The licence, publisher and a direct file link are also asked for because the Dataset type gets put on junk too: see the chapter on catalog discoverability.

Field	What to write
`name`	a title that distinguishes this dataset from its neighbour
`description`	50 to 5,000 characters, the same meaning as the page text
`license`	a link to a specific version, e.g. CC BY 4.0. A link to CC BY without a version number is weaker for Google
`distribution` with `contentUrl`	a direct link to the file and its `encodingFormat`. A link to a section of the site is not a file
`creator`, `publisher`	who collected it and who publishes it
`dateModified`	the file date. It is changed on the day the file changes
`temporalCoverage`, `spatialCoverage`	a period as `2020-01-01/2025-12-31` and a territory. If there is no period or territory, delete those lines from the template
`identifier`	the DOI. The book's template does not have this field: it is added once the number is issued
`sameAs`	the second page for the same dataset

Google says it understands both schema.org and DCAT on a page. The Rich Results Test accepts schema.org. The DCAT example in the same documentation is marked as not supported by the test. So JSON-LD goes on your own page: it can be checked before you give the link to people. The `dcat:Dataset` attributes on a JKAN page are not confirmed by this test. A `data.json` file for a harvester is a separate document; Google does not read it from the page.

A column dictionary for people and programs lives in `datapackage.json`. Google separately calls CSVW markup directly on the page experimental and warns that the test complains about such a block. For a table, a CSV via a direct link and a package descriptor are enough.

The page serves this HTML by itself, without a login and without a captcha. A shell that a script fills in the browser is empty for a robot. What this looks like on large sites is in the chapter on catalog discoverability.

Sitemap

Google asks for a sitemap to find dataset addresses, and for `sameAs` to understand where the copies are. One dataset is one line in the sitemap. In a catalog of dozens of pages without a sitemap, the robot may not reach every one.

A search engine reads `robots.txt` only from the root of the site's host name. This is what RFC 9309 says, and the same limitation is analysed for geoportals in the chapter on catalog discoverability. For the address `https://name.github.io/bus-stops/` the root is `https://name.github.io/robots.txt`. A `robots.txt` file inside the `bus-stops` folder is not covered by this rule. If you do not control the root, no separate directive will come from your folder. Then the sitemap is submitted to Google Search Console using the full `sitemap.xml` address.

GitHub Pages does not write Dataset markup or a sitemap by itself. Both files are put into the site build. Hosting limits and ready-made generators are in the chapter on static sites.

A file for the harvester

One page's markup does not get into the national catalog. A harvester takes a file in the profile it accepts now, not in the newest edition of the schema. data.gov still harvests DCAT-US 1.1, although version 3.0 has been published. Which language is served to whom is in the chapter on standards.

The file is prepared when there is a harvester and you have given it the address. Registering the site in Search Console does not tell the harvester this address. How to submit a catalog for harvesting is in the chapter on federations.

A file that declares a schema and is silent about mandatory fields may be harvested as it is. JKAN's short `data.json` and a small agency's records without a licence are analysed in the chapters on static sites and "When You Don't Need a Portal". If there is no harvester, a separate file is not set up just for the sake of a `conformsTo` line: the page with markup already describes the dataset.

A second page with a DOI

A DOI is needed when a release is cited in a text. It is not set up for an internal table. How to link a GitHub repository to Zenodo, and why `.zenodo.json` overrides `CITATION.cff`, is in the chapter on repositories. Until there is a number, the `identifiers` block is removed from the `CITATION.cff` template.

A Zenodo record of type "dataset" itself becomes a page with Dataset markup. DataCite says that landing pages with the Dataset class get into Google Dataset Search, and content negotiation by DOI does only if the record is in the Findable state and the resource type is Dataset. DataCite has

no control over when Google gets to the page. The "software" type does not get this class: for data, `.zenodo.json` sets `upload_type` to `dataset`.

A live record shows that the Zenodo page and the DOI response are two different documents. On 14 October 2026 the Cantometrics release, DOI 10.5281/zenodo.4898406, served one JSON-LD block in its HTML. An ordinary page request returned code 200. The same address with a `Mozilla/5.0` browser header returned 403: the refusal relates to that header, not to the page being missing.

- The type in the page markup is `https://schema.org/Dataset`. The short word `Dataset` right after the quotation mark is not there.
- The visible description text is "The data repository for the Cantometrics dataset", 48 characters. The string in the markup is longer, 56 characters, only because of paragraph tags. Google's threshold is 50 characters of the description itself.
- `license` points to the CC BY-NC 4.0 text. In the repository at the same tag, the licence file is CC BY 4.0. The divergence is already analysed in the chapter "When You Don't Need a Portal".
- `url` is `https://zenodo.org/records/4898406`. There is no `sameAs` field.
- `contentUrl` points to the release ZIP. A header request for this address returned code 200 and the type `application/octet-stream`. The markup gives the file as `application/zip`.

A JSON-LD request by the same DOI returned a different document that day: type Dataset, a 48-character description, no file link. Its `url` field is the old path `/record/4898406`, which answers with code 301 and redirects to `/records/4898406`. This response calls the GitHub tag address the article the dataset is based on, not a second page for the dataset.

The practical conclusion follows. The DOI and the Zenodo page cover the link from a report and give dataset search one more page. The description text, licence and direct link are kept on your own page and copied unchanged into the Zenodo record. Your markup's `identifier` gets the release DOI. `sameAs` gets the Zenodo record address. Citations use the release DOI, and the concept DOI for all versions is left for those who need the latest.

The order for one dataset

The example is a district bus timetable. The names are invented, and the files are this book's templates. Alongside, you can open the live Country Codes repository: it has a different subject, its descriptor is called `datapackage.yml`, and on 14 October 2026 the repository page HTML had no JSON-LD block.

1. **Repository.** Put in the templates: README, `datapackage.json`, `LICENSE`, the table in `data/`. In `CITATION.cff` keep `type: dataset` and delete the `identifiers` block until there is a DOI. In the README write what timetable this is, for which season, who is responsible and how often the file changes. Set the `dataset` and `open-data` topics: they mark the repository on the platform. What else goes into the package and where Git's limits end is in the chapter on repositories.

2. **Page.** Build a site from this repository or put a page on an existing site. The path `…/bus-stops` does not change along with the title. Insert the block from the markup template into the HTML. Make the description longer than 50 characters, for example: "Stop and departure times for district buses for the season: stop, route and time. The file is updated once a season." The licence is a link to CC BY 4.0. `contentUrl` is a direct link to the CSV. Do not add the `identifier` field yet.
3. **Markup check.** Paste the page address into the Rich Results Test. Fix Dataset errors before giving out the link. The page must open without a login: otherwise the test and the search engine see a placeholder.
4. **Sitemap.** Put a `sitemap.xml` with this page's address. If you control the root of the site's host name, list the sitemap in that root's `robots.txt`. Otherwise submit the full sitemap address in Search Console.
5. **Version and DOI.** If the timetable will be cited, set a tag, for example `v2026.1`, and get a release DOI. Write the number into `CITATION.cff` on the branch and into the markup's `identifier`. Put the Zenodo record address in `sameAs`. The licence in `LICENSE`, in the descriptor, on the page and in the Zenodo record is one and the same. For a table nobody cites, skip this step.
6. **Harvester.** If a regional or national catalog takes a file, put it in the profile the catalog accepts and submit the address. Fill in the fields the profile considers mandatory before applying.
7. **After a few days, and again after a few weeks.** Search for the title in Google Dataset Search. Google says crawling takes a few days, and dropping out of results takes days or weeks. Do not change platforms before then. First recheck the markup test and that the page opens without a login.

Example: the primary address on your own page

Context. Two people already run a quarterly table in a repository. Once a year they make a release and get a DOI. The national catalog does not harvest the organisation. A month later the table cannot be found by title in dataset search. One person suggests giving the repository link, another only the DOI, a third installing a portal.

Options. The repository tab on github.com. Only the Zenodo record. Your own page with markup, the DOI in `identifier` and the Zenodo address in `sameAs`. A portal.

Criteria. The markup can be checked before indexing. The link from the report opens an unchanging release. A person without the DOI opens the description and the file. Maintenance stays with these two people.

Decision. The primary address is your own page. The release DOI is in the report and in the `identifier` field. The Zenodo record is a second page for the same dataset.

Why. The github.com tab of a repository with a column dictionary contains no Dataset: on 14 October 2026 the Country Codes page opened that way. A Zenodo record sets the Dataset type itself, but takes the description and licence from what was sent to it. For Cantometrics the visible

text is shorter than Google's threshold, and the licence on the Zenodo page diverged from the file in the repository. A portal adds a server for one dataset and two publishers.

Consequences. The sitemap and markup are updated together with the file. An empty DOI is not published. Dataset search may first show the Zenodo page: it has the Dataset type too. Once your own page is indexed, the two pages are linked through `identifier` and `sameAs`.

When to review. There are more datasets than the two can mark up. A harvester appears and names the file profile. The description or licence on your own page and in Zenodo diverge.

How to write such a record for yourself is in the chapter "Justifying a Decision". The choice of repository for this table is in the chapter on Git.

How to check

Checking your own page takes a few minutes. Substitute its address.

```
PAGE=https://example.org/bus-stops

# The page opens without a login
curl -sI -o /dev/null -w "%{http_code}\n" -L "$PAGE"

# The HTML has the Dataset type: either the short name or the full schema.org address
curl -sL "$PAGE" | grep -o '"@type"[[:space:]]*:[[:space:]]*"[^"]*Dataset"' | head -1
```

Code 200 and a line with `Dataset` mean the page serves the markup. Then paste the address into the Rich Results Test: Google's documentation requires a description of 50 to 5,000 characters, and the test checks the markup against these rules. There must be no empty `identifier` in the block.

The commands below were taken on 14 October 2026 from other people's publications. They show the gap, not a model.

```
# The repository page opens. It has no JSON-LD block: the output is empty
curl -sL -o /dev/null -w "%{http_code}\n" "https://github.com/datasets/country-codes"
curl -sL "https://github.com/datasets/country-codes" | grep -o 'application/ld+json' | head

# JSON-LD by DOI: type Dataset, the old /record/ path, no distribution
curl -sL -H "Accept: application/ld+json" "https://doi.org/10.5281/zenodo.4898406"

# The old path redirects to /records/
curl -sI -o /dev/null -w "%{http_code}\n" "https://zenodo.org/record/4898406"

# The Zenodo page: code and Dataset type in the HTML
curl -sL -o /dev/null -w "%{http_code}\n" "https://zenodo.org/records/4898406"
curl -sL "https://zenodo.org/records/4898406" | grep -o '"@type"[[:space:]]*:[[:space:]]*"
[^"]*Dataset"' | head -1

# The record API and the file from contentUrl
curl -sL "https://zenodo.org/api/records/4898406"
curl -sI -L -o /dev/null -w "%{http_code}\n" \
  "https://zenodo.org/api/records/4898406/files/theglobaljukebox/cantometrics-v0.1-
alpha.zip/content"
```

That day the Country Codes page returned 200, and a search for `application/ld+json` found nothing. The DOI response was of type Dataset. The `/record/4898406` path returned 301. The `/records/4898406` page returned 200, and the type search found `"@type": "https://schema.org/Dataset"`. The record API returned 200 and type `dataset`. The headers of the file from `contentUrl` returned 200 and type `application/octet-stream`. A request for the same page with a `Mozilla/5.0` header returned 403.

Where the path breaks

- **A repository tab instead of a page.** The file and README are in place, but there is no Dataset type in the HTML. Ordinary search may find the README. Dataset search looks for markup.
- **A description shorter than 50 characters.** The Dataset type is declared, but a mandatory field is not filled in by Google's rules. Tags around a short phrase do not make up for the length of the text.
- **An empty or invented DOI.** A placeholder number from `CITATION.cff` gets into other people's citations. Until there is a number, there is no `identifiers` block and no `identifier` field.
- **One `sameAs` for two datasets.** Search glues different tables into one record.
- **`contentUrl` points to a section of the site.** There is nothing to download. The field holds a file address.
- **The file declares a profile and skips its fields.** The harvester publishes a holey record. Without a harvester, such a file is not set up.
- **Three places with three licences.** A person via the DOI and a person via the repository get different terms. The licence is written once and copied.
- **A captcha, a login or a script-built page.** The markup test and the robot see a placeholder. A month later the index will be empty for the same reason.

How this path differs from a portal

A portal builds the dump, the sitemap and often the markup for every page by itself. Here the page author does it, one dataset at a time. The check is the same as in the chapter on catalog discoverability: the address is open, the markup passes the test, the search engine knows the sitemap, and the harvester has been given the right profile. When there are more datasets than one person can describe, or harvesting runs on a schedule from many organisations, return to the chapter "When You Don't Need a Portal": the threshold is written there in advance.

Sources

- Dataset markup in Google's documentation: `name` and `description` of 50 to 5,000 characters are mandatory; schema.org and DCAT on the page; the Rich Results Test does not accept the DCAT example; `sameAs`, sitemap, `contentUrl`. Checked 14 October 2026
- DataCite: landing pages and Google Dataset Search: the Dataset class on the page, a sitemap or links, the Findable state and the Dataset resource type for content negotiation. DataCite's help still names the old Structured Data Testing Tool. The current test in Google's

documentation is the Rich Results Test

- RFC 9309: `robots.txt` is read from the root of the host name
- Google Search Console, Google Dataset Search
- Country Codes: on 14 October 2026 the page answered with code 200, and there was no `application/ld+json` block. The `datapackage.yml` descriptor name is analysed in the chapter on Git repositories
- Cantometrics, DOI 10.5281/zenodo.4898406, 14 October 2026. The page HTML has JSON-LD of type `https://schema.org/Dataset`, a 48-character visible description, the CC BY-NC 4.0 licence, `contentUrl` to a ZIP, and no `sameAs`. An ordinary page request: 200. A request with a `Mozilla/5.0` header: 403. The record API: 200, type `dataset`. JSON-LD by DOI: no `distribution`, a `url` with the `/record/` path, which answers 301. The headers of the file from `contentUrl`: 200, `application/octet-stream`
- Templates: /templates/README.md, /templates/datapackage.json, /templates/CITATION.cff, /templates/schema-dataset.jsonld
- Metadata export, catalog discoverability, GitHub and Zenodo

PART IX.

Practices

Good practices, anti-patterns, the lifecycle of data catalogs and their archiving.

1. Good Practices
2. Anti-patterns
3. Catalog Lifecycle
4. Archiving and Rescuing Catalogs

Good Practices

A good practice is one without which a dataset cannot be found or cannot be used. Everything else depends on the catalog type. Below are six things for any way of publishing, why each is mandatory and who already does it. The failures that skipping them turns into are collected in the chapter on anti-patterns.

What to do	Why the dataset is not used without it	Where it can already be seen
The file opens without paid software; encoding and delimiter are stated	otherwise every user fixes the table again, in their own way	open formats and a data dictionary, below
Columns have a dictionary	a `sum_2` column is formally open and closed in meaning	schema.data.gouv.fr, Table Schema
The record states a frequency, and it is about the file, not about edits to the description	services are built on the dataset and need to know when to expect new rows	HDX: the portal itself calculates the update deadline
Licence, publisher, a person to answer, period and territory are on the dataset	without them the file can be neither used lawfully nor judged for fitness	the record score on data.gouv.fr
The whole catalog can be taken by a program in the profile the upper portal already reads	a field "in our own way" does not exist for a harvester	paginated DCAT and weekly CSVs on data.gouv.fr
The dataset address does not change, and there is somewhere to report a fault	a link from a report and someone else's program outlive the platform	the data.gov transition in 2026; discussions on data.gouv.fr

In a technical specification these six points are marked as mandatory. A checklist for the other decision points is in the chapter on requirements. The full list by type is in the capability map.

File

The format opens without paid software. CSV, JSON, XML, GeoPackage, Parquet. PDF is kept as a human-readable copy alongside: a table from it gets retyped or OCR'd, and each person does it with new errors. Three questions that decide whether a file is fit for a program are in the chapter on formats.

The table has a data dictionary. Column name, unit of measurement, permitted values, what an empty cell means. The dictionary is put alongside as a separate file or described as is customary in the field: Table Schema, DDI variables, an SDMX structure. The forms, and how the French portal checks a file against a shared schema for the topic, are covered in the chapter on metadata depth.

Encoding, delimiter and decimal separator are stated. In the file record or in the dictionary itself: UTF-8 or another encoding, comma or semicolon, point or comma in numbers. Otherwise half of one publisher's tables open and half do not, and nowhere does it say which is which. This is what the open data section of Rosstat looked like; the analysis is among the anti-patterns.

One dataset is what people search for by name. A year and a district that can be made a column are not given separate records to boost the count. What is worth splitting is what is searched for on its own: a city budget, a time series with a persistent address. The test question: would it be more convenient if these files were merged into one? If so, it is fragmentation. Examples and the boundary are in the same chapter on anti-patterns.

The frequency in the record is about the file, and someone checks it. The update date is the date of the rows, not the day the description was edited. A user builds a calculation on the dataset and needs to know when to expect the next instalment.

On the Humanitarian Data Exchange frequency is a mandatory field from a short list: every day, every week, every year, as well as "live", "as needed" and "never". The portal itself calculates the date by which the next file is expected. Since April 2025 the rule is: the stated frequency is added to the end date of the period. The EM-DAT country profiles dataset is updated weekly. On 14 October 2026 its data period ended that same day, and the next update was due on 21 October. That day 866 datasets had the frequency "every week": the field is not left empty. When the deadline passes, an email goes to the person responsible for the dataset, not to the portal's general inbox. HDX's own authors ask people to choose the nearest regular interval rather than "as needed": otherwise the check does not fire.

Record

Licence, publisher, contact, period and territory are on every dataset, not in the site footer.

- **Licence**: a link to the text of the terms. Without it the file can be downloaded but cannot be used lawfully. Which one to choose is in the chapter on licences.
- **Publisher**: the organisation responsible for the content. All of one agency's datasets are found by it, and a copy is distinguished from the original source.
- **Contact**: a person or team with an address who will answer about an error in the rows. An `info@` inbox without a name does not cover this: there is no one to read the email.
- **Period and territory**: so that fitness can be judged without downloading. "Data for all years for the whole country" in the description does not replace filled-in fields: a program cannot filter on it.

Beyond this list, one profile is needed that your upper portal already reads: DCAT-AP, ISO 19115, DataCite, DDI or SDMX. Not all at once, but the one that matches the type. The upper portal and the search engine read the description by program, and a field with a local name is empty for them. How to choose a description language is in the chapter on standards.

A copy points to the original. A city, a region and a country may show one record, but the copy has a link to the source. Otherwise an error is fixed in three places, and search counts one dataset as three.

data.gouv.fr shows the publisher a record score so that gaps are visible before a user complains. The portal's guide lists what goes into the score: a long enough description, described files, a frequency that is filled in and kept to, an open licence, at least one open format, the territory and its level of detail, the period. Since 2022 the score has been experimental: the weights may change, and checking against a data schema is not yet included. On 14 October 2026 the record of the portal's own catalog passed seven of the nine checks the API returns. File descriptions and temporal coverage were not covered. The licence was there, though (the French open Licence Ouverte 2.0), the frequency was filled in and kept to, and the territory was stated. The score looks at the record, not at the rows inside the file. An empty file with filled-in fields would get a high score here; how to tell one from the other is in the chapter on quality levels.

A catalog that can be harvested

Without a dump of all descriptions, the catalog will not get into a federation. Without a persistent address, the dataset cannot be cited. Without registration, those who search for portals rather than individual datasets will not learn of the catalog.

The dump opens without a login and without a contract. For a national portal one file is already too tight, and that is fine if a program can walk the pages. On 14 October 2026 the data.gouv.fr DCAT dump contained 74,616 datasets, 100 per page, 747 pages in total: the file states the total count and the address of the next page. Alongside, the portal publishes the whole catalog as tables. Its record states a weekly frequency. On the morning of that day there were fresh CSVs: a 276 MB list of datasets and a 1.1 GB list of files, plus organisations, discussions, reuses and harvesters. Any aggregator takes this without a separate agreement. Other approaches (a single file, a protocol, a change feed) are in the chapter on APIs and dumps.

The address survives a platform change. It is designed before the move: the old path either stays or redirects to the new one. When data.gov moved from CKAN to its own software in April 2026, it kept the previous catalog at a separate address for a transition period so that users could reconfigure their programs. How to keep an address is in the chapter on identifiers.

A search API is switched on when there are hundreds of datasets or more. For a dozen, pages and one description file are enough. Descriptions are then readable without registration and without a captcha: a file can be restricted, a record cannot.

There is somewhere to report an error, and the message reaches the data owner. A form on the dataset page, an address in the record or an open tracker. On data.gouv.fr a dataset has a discussion, and reuses (applications and articles that readers have attached to the record themselves) are served by the portal as a separate file in the same weekly dump. Feedback is then visible both to people and to those who take the whole catalog. How the French portal works overall is in the chapter on open data portals.

The catalog is registered where people look for it. A national and a thematic registry, and re3data for a research repository. A news item on your own site does not replace this. Where to register is in the chapter on discoverability.

What the type adds

Six points are enough for an agency page and a city portal. The catalog type adds its own. This is not a second mandatory list "just in case": what is surplus for your type slows down publishing and is of no use to the user.

Type	What to add to the six points	Where it is covered
Open data portal	the national profile, most often DCAT-AP, and a shared schema for topics many publish	standards, federation
Research repository	DOI, versions, a ready citation for an article, serving descriptions via OAI-PMH	research repositories
Geoportal	the layer can be downloaded, or it says plainly that only viewing is available; a description in ISO 19115	geoportals

Type	What to add to the six points	Where it is covered
Indicator catalog	methodology next to the series; SDMX or another bulk export of the table, not just a screen	indicator catalogs
Microdata catalog	access levels, a variable dictionary, a description that is open even when the file is closed	microdata
ML catalog	a Croissant description, a fixed split into training and test parts, licence and data provenance	ML catalogs
Publishing without a portal	the page opens without a login, the licence is in place, schema.org markup with a description of 50 characters or more, a date and a contact	making it findable

A chart on the site does not replace these points. A file or a dump must still be downloadable.

The dataset as a product

The practices above can be fulfilled formally: the record is filled in, the file opens, and yet the dataset is awkward to use. A different view helps: treat the dataset as a product that has users. In companies this is called "data as a product". How a data product differs from a dataset, and which catalogs are built around products, is covered in the chapter on units of publication. For open data the approach is still rare. A check of one dataset looks like this.

1. It is known who uses the dataset and for what task. At least one scenario, even if the file is free.
2. There is an owner who understands both the data and the users. They answer questions and fix the dataset based on feedback. Not "a department", but a person or team with an address.
3. Documentation is published and matches the file: schema, code example, limitations. HDX has a separate record field for limitations.
4. The data can be obtained in more than one way: file, API, slice. The format is chosen for the user. An analyst needs CSV or Parquet, a developer an API or JSON. An example of how this changes demand for essentially the same data is in the chapter on Git repositories.
5. Quality is described and checked continuously, not once before first publication and not only when a user writes in.
6. There are versions and an update schedule, and the schedule is kept. For a research dataset the version is what an article cites. For an operational one, at least the file date and previous releases if anyone may already have cited the old ones.
7. A dashboard, if there is one, sits next to the data.

Point 6 coincides with the frequency from this chapter. The rest is what even a carefully filled-in record does not give. The idea of a "third wave" of open data says the same: publish what is needed for specific tasks; see high-value datasets.

Where these rules come from

The practices were not invented for one portal. They are repeated by three documents that national open data policies and research repositories rely on. In the book they are laid out by type: what is mandatory for a research archive is surplus for an operational portal.

Practice from this chapter	Open Data Charter, 2015	Data on the Web Best Practices, W3C, 2017	FAIR Principles, 2016
Open format and data dictionary	accessible and usable; comparable	formats and vocabularies	interoperable
Frequency and completeness	timely and comprehensive	data quality	—
Licence on the dataset	open by default	licences	reusable
Dump of the whole catalog	accessible	data access and metadata	findable
Persistent address	—	identifiers	findable
Where to report an error	improved governance and citizen engagement	feedback	—

The International Open Data Charter has six principles for governments: open by default; timely and comprehensive; accessible and usable; comparable and interoperable; for improved governance and citizen engagement; for inclusive development and innovation. On 14 October 2026 the charter's site said it had been adopted by more than 29 national and 146 regional and city governments. The charter says what to publish and says almost nothing about what file to use.

Data on the Web Best Practices is a W3C Recommendation of 31 January 2017 with 35 practices: from description and licence to versions, preservation and feedback. It is the list closest to this chapter. It is about data on the web in general, not only government portals.

The FAIR principles are for research data: findable, accessible, interoperable, reusable. FAIR does not mean "open": a dataset with closed files can also comply if the description is open, the identifier is persistent and the access procedure is clear. For a city CSV they provide vocabulary, not a substitute for a licence and a dump.

How to check your catalog

A short pass through the six points. Counting on a sample of 30 datasets and analysing the files is in the chapter on self-audit.

1. Open one file without paid software. If it is a CSV, read the encoding and delimiter in the record instead of guessing.
2. Find the data dictionary. If a column is understandable only to whoever made the export, the dataset is not ready for an outside user.
3. Compare the date in the record with the date inside the file and with the stated frequency. A delay must be visible to the owner before the user.
4. The record has a licence as a link, a publisher, a person's address, a period and a territory.
5. Download the dump of all descriptions without a login. Make sure it is in the profile the upper portal expects.
6. Open the old dataset address if the platform has already been changed, and send a test error report. It must reach the data owner.

This is what a good answer at step five looks like. The number of datasets in the data.gouv.fr paginated dump:

```
curl -s "https://www.data.gouv.fr/api/1/site/catalog.rdf" | grep -o "hydra:totalItems[^<]*"
```

The frequency and update deadline of one dataset on HDX:

```
curl -s "https://data.humdata.org/api/3/action/package_show?id=emdat-country-profiles" \
  | jq '{freq: .result.data_update_frequency, due: .result.due_date, period:
.result.dataset_date}'
```

On 14 October 2026 the first command showed 74,616 datasets, and the second a frequency of `7` (weekly) and a deadline of `2026-10-21`.

Sources

- Data on the Web Best Practices, W3C Recommendation, 31 January 2017
- FAIR Principles; M. D. Wilkinson et al., The FAIR Guiding Principles, Scientific Data, 2016
- International Open Data Charter; the number of governments that adopted it is from the home page, 14 October 2026
- Table Schema
- HDX: record fields, how freshness is calculated since April 2025, which frequency to set; the EM-DAT Country Profiles dataset and a search by frequency `7`: 14 October 2026, 866 datasets
- data.gouv.fr: announcement of the metadata quality score (in French), 10 August 2022; guide to the score (in French); DCAT dump, 74,616 datasets; weekly catalog CSVs: 14 October 2026
- Announcement of the new data.gov catalog, 15 April 2026
- Requirements Checklist, capability map

Anti-patterns

An anti-pattern is a decision that looks like publishing and does not let anyone use the data. Below are those that can be seen in live catalogs. Shares are from the dataportals-registry export (release 1.27.0 of 15 October 2026) unless stated otherwise.

No metadata dump

34% of catalogs have no programmatic interface recorded. A federation harvester, or a program that needs one file of all descriptions, will not take such a catalog. A search engine will open the page if it has schema.org markup and a sitemap.

Among dataset lists, an interface is recorded for about 14% of registry entries. The findability of such a list rests on the licence, a persistent address and markup. A list without its own API is a normal approach for a few dozen datasets; see when you don't need a portal.

An interface is recorded for 21% of marketplace entries, 23% of ML catalogs and 37% of indicator catalogs. For indicators this hurts most: a table builder in the browser has no substitute.

A showcase for a deadline

Datasets were uploaded once and are not updated. The date in the record is the portal's launch day. There is an "Open data" section with no data in it: on the Rosmorrechflot (the Russian federal river and sea transport agency) site in 2026, a page says in words that data is published, while the link leads nowhere because the portal's domain was deleted back in 2022. The Perm city portal was replaced in 2023 by a page with two datasets, opening hours and a phone directory, although it had previously had more than 480.

The only thing worse than an empty date is a false one. data.gov.uz showed in its monitoring section a dataset as updated in December 2022 that contained a single figure for 2020. The update mark was set without changing the data. How to check your catalog for these signs is in the chapter on self-audit.

Fragmentation to boost the count

The number of datasets is a portal's most visible figure, and it is easy to inflate. One indicator is cut into files by year, month and district so that the report says "several thousand". A user needs one dataset with a breakdown, or an API, not a hundred identical names that have to be downloaded one by one and stitched together.

What this looks like in numbers:

- Uzbekistan's open data portal had 6,623 datasets in February 2023, 42.6% of which consisted of a single row, and all the data together weighed 40 MB. By summer 2025 there were 10,412 datasets, 5,207 of them single-row: exactly half. Only 114 datasets had more than a thousand records.

- In autumn 2021 Rosstat's open data section held 1,547 datasets, of which 742 were files of identical structure that could have been published as one dataset. Without fragmentation there would have been about 190.
- Poland's dane.gov.pl claimed almost 18,800 datasets in 2020, more than 7,500 of them monthly files from a single patent office.
- At the end of 2025, 52,000 of the 63,000 datasets on Austria's data.gv.at were municipal budgets cut up by year.

A related trick is to pad the count with someone else's content. Vietnam's data.gov.vn claimed 10,616 datasets in 2022, of which 10,045 were books and articles from a university repository. There were fewer than six hundred datasets.

A dataset can be split if a part is needed on its own. Each municipality's budget is a separate object: people search for it by the town's name. A year of the same budget is not; it is a column. Central banks such as the ECB and the Bank for International Settlements publish each time series as a separate record, and this is justified: a series is the unit of search and is named as such, while the whole collection is served as a single dump. The test question is simple: would the user be better off if these datasets were merged into one? If so, it is fragmentation. How to measure a catalog by substance rather than by number of datasets is in the chapter on metrics.

A dashboard instead of data

The portal shows charts and widgets in a BI system (Power BI, Tableau, Qlik, Superset), and only what is on the screen can be downloaded. Pakistan's national open data portal nodp.gov.pk in 2026 is a set of Superset dashboards with statistics. The IAEA's nuclear reactor system PRIS is built on Power BI: the data is fresh, but it can only be obtained by exporting from a visualisation, with no API. Such sites are also no good for archiving: the web archive does not preserve BI system pages. The statistical BI portal for Moscow and the Moscow agglomeration on Rosstat's site will vanish without a trace if it is switched off.

A dashboard is a good addition to data, not a substitute. The IMF's new data portal also shows Power BI dashboards, but alongside it serves series through an SDMX API. The order is: first the dataset and the dump, then the visualisation.

A file that cannot be read

PDF, a scan, a table with merged cells and a three-row header. The format counts as machine-readable only by its extension. There is no data dictionary.

The extreme case is a document passed off as data. The Russian Interior Ministry's site once published XML files saved from Word as open data: formally XML, in substance the markup of a text document. The same trick is an "RDF" export that is really ordinary XML without vocabularies or links. Such a file adds a format to the portal's statistics and gives a program nothing.

Quieter but no less harmful are files without a description of technical details. In the same Rosstat open data section (2021), half the CSVs used a comma and half a semicolon, about 30% were in windows-1251 encoding and the rest in UTF-8, and nowhere did it say which was which. Some files

with a `.csv` extension turned out to contain a ZIP. Every user guesses again.

Rights that cannot be found

A catalog-level licence could be recorded in the registry for only about 1% of entries. Details and a caveat are in the chapter on licences. In practice this means that even if a licence exists somewhere, it is not visible in the dataset description.

The door is closed to programs

A captcha, mandatory registration to read a description, blocking of automated requests, undocumented limits. Metadata must be readable without a login. A file can be restricted, a record cannot.

There are more inventive locks too. In 2023 Kazakhstan's data.egov.kz dropped mandatory registration but began serving no more than 100 records from a dataset at a time; this was still the case in 2025. The European mass spectrometry catalog Metaspace gives a download link that stops working after half an hour: the file has no persistent address. Since 2024–2026 a new reason has joined the old ones: robots collecting data for model training. In 2026 the US Patent and Trademark Office put the bulk downloads and API of the data.uspto.gov portal behind registration, although it had been one of the most open. Protecting against load is possible, but in such a way that programs with clear intentions get through without manual approval: published limits, a key issued automatically, a full dump as a file that does not have to be pulled through queries.

The same door closes entirely when a site does not respond to requests from abroad. Many Russian government sites are unavailable outside Russian networks. Along with foreign users, Internet Archive, search engines and data aggregators stop seeing them: the catalog drops out of both the archive and search. This is easy to check: open the site through a service or server in another country. More on the consequences is in the chapter on archiving.

A home-grown format instead of a standard

To collect datasets from thousands of sites, people invent their own list format instead of DCAT or schema.org. This happened in Russia: methodological recommendations obliged agencies and municipalities to keep a list of datasets in a CSV file according to their own schema, and data.gov.ru collected these lists. The result was thousands of small lists that were rarely updated, and a format no foreign aggregator or search engine understands. The harvesting standard is chosen from those others already read; see federation and harvesting.

Links break during a move

The platform was changed, the dataset addresses became different, and there are no redirects. Or the file was not in the catalog but on a neighbouring site, and the neighbouring site moved. The rule was formulated long ago: in the W3C note "Cool URIs don't change" (1998), Tim Berners-Lee writes that it is not technology that changes an address but people who did not think about it in

advance. The counter-example is data.gov: when moving off CKAN in 2026, it kept the old catalog at a separate address for a transition period. A persistent address and a migration plan are discussed before the move; see identifiers and the cost of leaving.

The portal was abandoned

868 catalogs in the registry are not active, and 98 have been decommissioned. A typical story: a pilot on a grant, the grant ended, nobody is left to update it. Closing can also be the right thing if the data and addresses were handed over. Silent decline is not.

Decline usually stretches over years. The Russian Ministry of Health's open data section did not update its datasets from 2017 and disappeared completely in spring 2024. The Social Insurance Fund's portal data.fss.ru by 2022 was still "in test mode", with no link from the fund's site and data for 2018. Signs of abandonment are visible long before shutdown: the last news item several years ago, the publication plan removed, file links leading nowhere. See lifecycle and archiving.

One dataset on three portals

A city, a region and a country show the same record and do not say which is the source. Search lies about the number of datasets, and an error has to be fixed in three places. The copy's record must link to the original.

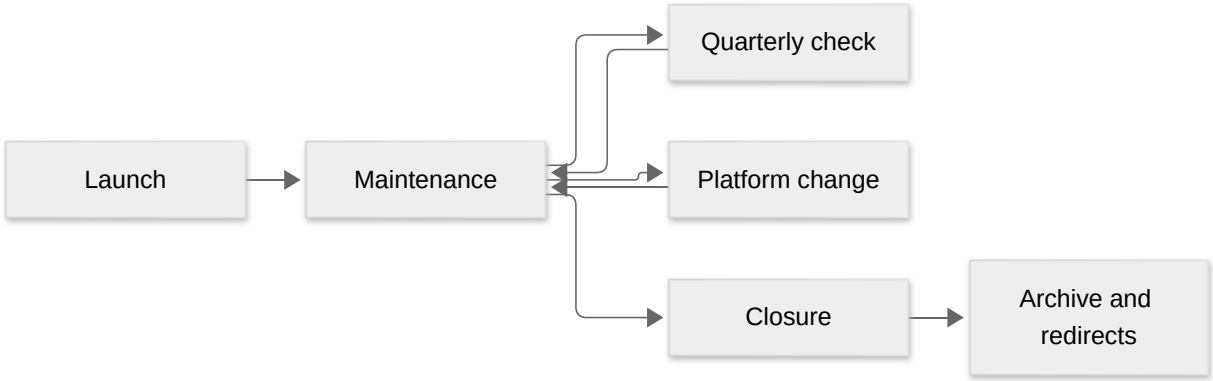
Sources

- The dataportals-registry export, release 1.27.0 of 15 October 2026.
- Data on the Web Best Practices, W3C, 2017: descriptions, licence, persistent addresses and access for programs as mandatory practices
- Cool URIs don't change, W3C, 1998
- Announcement of the new data.gov catalog, 15 April 2026
- I. Begtin, What is wrong with Uzbekistan's open data portal? (in Russian), February 2023: dataset sizes and update monitoring on data.gov.uz
- I. Begtin, Why it is impossible to praise Rosstat (in Russian), January 2022: an analysis of Rosstat's open data section
- I. Begtin, Good and bad data publishing practices (in Russian), November 2024: the Interior Ministry example; catalogs of bad practices OpenDataFail and Bad Data
- I. Begtin, Why data on Russia is so hard to find (in Russian), November 2024: sites unavailable from abroad and a home-grown harvesting format
- I. Begtin, Telegram (in Russian), on fragmentation: dane.gov.pl, 17 December 2020; data.gov.vn, 21 August 2022; data.egov.uz in 2025, 10 July 2025; data.gv.at, 23 December 2025; when slicing is justified, 11 January 2024
- I. Begtin, Telegram (in Russian), on dashboards: PRIS, 11 September 2026; nodp.gov.pk, 15 August 2026; Rosstat's BI portal, 23 May 2025; data.imf.org, 19 April 2025
- I. Begtin, Telegram (in Russian), on closed access: data.egov.kz, 26 April 2023, and in 2025, 29 October 2025; Metaspace, 11 July 2024; data.uspto.gov, 23 June 2026

- I. Begtin, Telegram (in Russian), on abandoned sections: Perm and Primorye, 2 April 2023; the Ministry of Health, 13 November 2024; Rosmorrechflot, 13 May 2026; the Social Insurance Fund, 15 July 2022
- Good Practices

Catalog Lifecycle

A catalog is launched in weeks and maintained for years. Most failures happen not on opening day but when the project has ended, the platform has changed or the only administrator has left.



Launch

Launch is the day an outsider opens a real dataset, downloads the file and sees who will update it next year. The public address is the last step of that day. How long different approaches take to install is in the chapter on deployment.

A pilot is about ten real datasets, and one of them is awkward. An awkward dataset is one whose source has no licence, whose file is large or whose updates are promised daily. The pilot shows who will add the licence, who will notice a broken link and whether the upper portal reads your dump. The vendor's demo datasets do not answer these questions. What to check on a test instance is in the chapter on choosing a platform.

Different people are responsible for the data and for the platform. They are named, by name, before the public address.

This is how New York does it. The nyc.gov/opendata portal is run by the open data team in the Office of Technology and Innovation: strategy, standards, the site itself. Each city agency appoints its own open data coordinator. Through them the agency hands datasets to the portal, updates them at the frequency stated in the record, answers residents' questions and prepares its part of the annual report. If a coordinator leaves, the agency must appoint another and inform the portal team. This is written in the technical standards manual, edition of 12 May 2023. The scheme is alive: the annual report workbook for 2025 on the portal's site was updated on 23 June 2025.

Before the public address:

- [ ] The pilot has run on real datasets, including an awkward one.
- [ ] Two people are named: who updates the rows and who updates the platform.
- [ ] The address scheme is written down so that it survives the next platform. How to keep it is in the chapter on identifiers.
- [ ] A full dump of descriptions and files has been downloaded and opened once.

- [ ] A position and a budget are named for next year. A contractor who leaves on the day of sign-off does not fill these slots. What to put in the contract is in the chapter on build models.

A site can open without a launch. On 15 July 2025 the Russian Ministry of Economic Development reopened data.gov.ru. The previous version had been closed in March 2023 "for transfer to a new platform". The site came back. Through the API it had 5,826 datasets, about 100 MB compressed, and 84% of them were first published in 2019 or earlier. The snapshot of the previous version from 2 February 2022 (26,490 datasets, a 14 GB archive) did not return to the new portal. It is in the National Digital Archive; see below. An analysis by row count and by which agencies are missing from the portal is in the chapter on self-audit. Such an opening becomes a launch when fresh datasets with a named owner appear at the new address.

Maintenance

From then on the catalog lives by checks. The number of datasets in a report does not replace them.

The schedule for links and for a full pass is already set in the chapter on self-audit: once a month a dozen known links are opened, once every six months the whole sample is walked through. It makes sense to add two rarer checks. They are done by the same two people named at launch: rows and records are checked by whoever is responsible for the data; the version, the copies and the bill by whoever is responsible for the platform.

How often	What to do
Once a quarter	The licence on the dataset is filled in. The date in the file matches the promised frequency. The platform version is still supported. Both people from the launch are still in place.
Once a year	Someone tries to restore the copy. The full dump is downloaded again. The expiry date of the licence or contract for a commercial platform is in the calendar.

Links. If the monthly check is skipped for years, you get the UK's data.gov.uk. The portal has been underfunded since 2017. By March 2026 at least a quarter of links led to an error page, and many datasets had not been updated for years by organisations that no longer exist. The platform, though, was kept: the previous catalog works, and a showcase of curated collections was built next to it. How this showcase works is in the section on platform change.

Frequency. The date in the record is the date of the rows. On the Humanitarian Data Exchange the overdue notice goes to the person responsible for the dataset, not to the portal's general inbox. How the deadline is calculated is in the chapter on good practices. For your own catalog the same recipient is enough.

Version. The platform and extensions are upgraded while the version is still in the support window. For CKAN the latest minor version is fully supported, and the previous one gets bug and security fixes. On 14 October 2026 these are 2.12 and 2.11. A release older than the previous one is already outside the window, and the upgrade usually turns into a re-implementation: extensions, schema and the way files are stored change along the way. Other platforms' windows are in the chapter on deployment.

People. A site can still open when nobody is left to answer. In New York, replacing a coordinator is the agency's duty. The other end of the same story can be seen in the registry. According to the dataportals-registry export (release 1.27.0 of 15 October 2026), 868 catalogs are not active and 98 have been decommissioned. The share of non-working catalogs varies widely: about 0.9% among indicator catalogs and about 18% among microdata catalogs. Behind the second figure are mostly abandoned installations: 47 of the 59 non-working microdata catalogs are old NADA catalogs of statistical offices, often on IP addresses and without HTTPS. They were installed under support programmes, and maintenance was not budgeted. Details are in the chapter on microdata catalogs. A catalog without an owner and a maintenance budget dies with the project. The signs visible from outside years before shutdown are collected in the anti-pattern "The portal was abandoned".

The annual copy check is the same idea as in the chapter on deployment: a copy nobody has tried to restore does not count as a copy. The ArcGIS licence date at which the Arkhangelsk Region geoportal stopped shows why this date goes into the calendar in advance. The case itself is among the migration failures.

Platform change

A change of platform is the main cause of dead links. Before the move, a dump of all descriptions and files is made, plus a table of old and new addresses. After the move, the old addresses are redirected. If everything cannot be redirected at once, the old system is kept at a separate address for a transition period. This is what data.gov did in April 2026: the catalog moved from CKAN to its own software, dataset pages stayed at the same addresses, and the old site was kept at catalog-old.data.gov. In the project description, updated on 23 July 2026, this address was still listed as working throughout the autumn. On 14 October 2026 the site no longer opens: the name catalog-old.data.gov still exists but points to an address that does not. The previous CKAN programmatic interface returns an error at the old path. The new interface lives at a different address and asks for a key, so programs that used CKAN had to be rewritten. How to estimate the cost of leaving in advance is described in the chapter on build models.

Moves go in both directions, and neither direction is a mistake in itself:

- **To CKAN.** Since 2019 Australia's data.gov.au ran on a mixed setup: Magda provided search and the showcase, with an outdated CKAN underneath. In June 2025 the portal was consolidated into a single CKAN with Drupal for pages; more than 100,000 datasets and harvesting from a thousand sources were preserved, and the CKAN programmatic interface remained the same. Greece's data.gov.gr moved to CKAN from its own software in 2025–2026 and, by harvesting other catalogs, grew from a hundred datasets to tens of thousands. On 14 October 2026 it is CKAN 2.11.3 with 22,762 datasets.
- **From CKAN.** Besides data.gov, Mongolia's opendata.gov.mn moved to its own engine, and the programmatic interface disappeared along with the old platform. On 14 October 2026 the old CKAN API path opens a site page rather than an interface response.
- **Without a platform change.** In 2026 the UK's data.gov.uk became the National Data Library: a new showcase with curated collections, while the previous catalog was kept as a separate section so there is something to compare against.

Typical failures recur too. When the Slovak portal moved to data.slovensko.sk in 2024, dataset addresses changed and links to them from the European portal stopped working. Malaysia, moving data.gov.my to a new engine, kept the old catalog at archive.data.gov.my. On 14 October 2026 a connection to this address cannot be established, so the temporary archive turned out to be temporary. The Arkhangelsk Region geoportal lost its layers when its ArcGIS licence expired: the catalog kept showing records for layers that no longer existed. Russia's data.gov.ru was closed in 2023 "for transfer to a new platform" before that platform existed, and for two years there was no portal at all.

Hence three rules. The old version is kept until the new one has taken over all addresses. The programmatic interface is as much a part of the address as the dataset page: if it changes, harvesters lose the catalog. The licence for a commercial platform is a date by which the migration plan must be ready.

Closure

Closing a catalog is fine if nobody uses it or there is nothing left to run the platform on. Closing it by switching off the server on the day users find out is bad.

Before shutdown the catalog runs read-only for several months. The site, the files and the programmatic interface open, new edits are stopped, and the front page says why and until what date. A pause without a date is already an abandoned site. data.gov.ru skipped this stage: the portal was switched off "for the duration of the transfer", and for two years the old address offered neither reading nor an explanation of where the datasets had gone.

Before shutdown:

- [ ] The closure is announced in advance, months ahead. Until the announced date the catalog runs read-only.
- [ ] The data is handed over to another catalog or archive, or the old address states plainly that it has been withdrawn from publication.
- [ ] Old addresses redirect. If not everything can be redirected, for at least a year they say where the dataset has moved.
- [ ] A snapshot of the descriptions is saved: a dump through the programmatic interface or a DCAT file, not just site pages.
- [ ] Files that exist only here are saved together with the addresses they were taken from.
- [ ] The site is in Internet Archive or a national web archive, and the archive can reach it, including from abroad.
- [ ] The upper portal and the registries where the catalog was registered have been notified.

The same list is in the checklists.

How to take a snapshot

"A snapshot of the descriptions is saved" is not a screenshot of the home page. Archiving a catalog goes through four steps.

1. **Find everything that belongs to the catalog.** The main site, subdomains, old versions such as `old.` and `archive.`, separate map and API services. Old versions of a site disappear first: they are kept "just in case" and then switched off unnoticed.
2. **Set priorities.** First, what will disappear sooner and exists nowhere else: files that exist only here, services on a platform being shut down.
3. **Collect.** Site pages with a web archiver in WARC format, which Internet Archive and national libraries read. Descriptions as a dump through the API or a DCAT file. Files downloaded as they are, together with the links they were taken from. If data is served only through an API, a separate dump is needed, for example with the APIBackuper tool: a web archiver does not crawl APIs.
4. **Publish or deposit.** WARC files and dumps go into open storage, and their description into a catalog, so the archive can be found. At a minimum, deposit a copy in cold storage and write down where it is.

WARC can be viewed without a server in ReplayWeb.page. Sites with dynamic loading need a browser-based crawler, such as Browsertrix Crawler.

If someone else is closing it

A catalog can be closed by its owners without asking the users. Then the data survives only in other people's archives. Russia's data.gov.ru was closed in March 2023 "for transfer to a new platform". A full snapshot of the previous version (26,490 datasets, 14 GB compressed) survived in the National Digital Archive: it was made on 2 February 2022. The version opened on 15 July 2025 did not bring this data back. How catalogs disappear, why the web archive preserves them poorly and how to take a copy of someone else's catalog is in the chapter "Archiving and Rescuing Catalogs".

Catalogs that simply stopped being maintained are counted in the maintenance section. A separate case is when the product itself is withdrawn from support. Nesstar was declared obsolete in 2015 with version 4.0 and finally withdrawn by 2022. Canada's social survey archive ODESI chose Borealis, built on Dataverse with DDI descriptions and DOIs, for its move, and switched off Nesstar only after the transfer: in October 2026 the old address leads to the new site. The end of someone else's product is met with a migration plan while the old system still opens.

Sources

- The dataportals-registry export, release 1.27.0 of 15 October 2026.
- New York open data coordinators: a coordinator's duties; the annual report workbook for 2025 was updated on 23 June 2025
- Technical Standards Manual, edition of 12 May 2023: who runs the portal, who runs an agency's datasets, replacing a coordinator who leaves
- Announcement of the new data.gov catalog, 15 April 2026; catalog description, updated 23 July 2026: the old address was listed as working throughout the autumn; the new programmatic interface: a different address and a key. catalog-old.data.gov being unreachable and the old Action API error: checked 14 October 2026

- Link Digital, moving data.gov.au to CKAN, 2025; GDS, What's changing on data.gov.uk and why, 25 March 2026: underfunding since 2017, at least a quarter of links lead to an error
- I. Begtin, Telegram (in Russian): data.gov.gr on CKAN, 18 May 2026; opendata.gov.mn, 25 August 2026; data.slovensko.sk, 20 June 2024; data.gov.my, 19 January 2024; the Arkhangelsk Region geoportal, 21 August 2024
- Nesstar: product history, ODESI's move to Borealis
- The WARC format as described by the Library of Congress, ReplayWeb.page, Browsertrix Crawler, APIBackuper: dumping data from an API for an archive
- I. Begtin, How to archive digital resources properly in archiving campaigns (in Russian), November 2021: the four stages of archiving; Web archiving and its limitations (in Russian), February 2025
- I. Begtin, The updated data.gov.ru portal (in Russian), 16 July 2025: 5,826 datasets, about 100 MB, 84% first published in 2019 or earlier; the archive copy of 2 February 2022
- Good practices, self-audit, microdata catalogs
- data.gov.gr: `status_show` and `package_search` on 14 October 2026: CKAN 2.11.3, 22,762 datasets. The opendata.gov.mn API path and the archive.data.gov.my address were checked the same day.
- CKAN release support policy: on 14 October 2026, 2.12 is fully supported and 2.11 gets fixes
- Deployment Speed and Operations
- Persistent Identifiers

Archiving and Rescuing Catalogs

A catalog disappears more often than it seems, and usually without notice. The chapter on lifecycle describes closure through the owner's eyes: what to do before switching off the server. This chapter is about the other side. The owner did not take care of an archive, the decision to close was not theirs, or you use someone else's catalog and are not sure it will live to next year.

The main conclusion: the web archive does not save a catalog by itself. It preserves pages, but descriptions from the programmatic interface, large files and sites closed to foreign addresses are preserved poorly or not at all. A copy of the datasets you need is made in advance and deliberately.

How data disappears

In 2017 the Sunlight Foundation described eight ways information disappears from US government websites. Three matter for catalogs: removal of datasets, removal or redesign of search portals, and moving pages without redirects. In practice it looks like this.

Silent removal. In 2023 Rosimushchestvo (the Russian federal property agency) removed its open data section: 24 datasets, about 1 GB of XML. Links to the section disappeared from every page of the site, and there was no announcement. When it happened could only be established through Internet Archive: the last snapshot of the page was taken in February 2022.

Shutdown by a decision days in advance. The Moscow Region open data portal data.mosreg.ru was decommissioned in March 2019. The order was signed on 25 March, the system was switched off on the 29th, and the order itself was published two months later. Nobody made an archive of the portal.

"Temporarily, for the duration of the move". Russia's data.gov.ru was closed in March 2023 for transfer to a new platform. The new version opened only in summer 2025 and did not bring back the previous data. More in the chapter on lifecycle.

Liquidation of the owner. In a reorganisation the archive passes to the successor; in a liquidation it passes nowhere. The site of the St Petersburg Charter Court was closed together with the court's liquidation in 2021. When Rosturizm (the federal tourism agency) was abolished in 2022, its subdomains, including the open data portal, were put at risk.

Ending publication along with the archive. An agency decides to stop releasing a series and at the same time deletes everything it released before. This is what the Bank of Russia did in 2024 with over-the-counter foreign exchange market statistics, and Rosaviatsiya removed its archive of passenger traffic statistics going back to 2010.

A change of policy. In the US in 2025, data on climate, health and foreign aid was removed from federal sites on a mass scale; the USAID site was switched off together with the agency's data portal. The National Security Archive at George Washington University keeps a chronology of the removals.

Closure of someone else's platform. Commercial and shared hubs close too: in 2024 Microsoft closed the Planetary Computer Hub cloud environment where researchers worked with geodata, and users had to retrieve their notebooks and files in a hurry. The risks of hubs are covered in the chapter on cloud storage.

In every case the first sign is not an announcement but a missing link.

Why the web archive does not save a catalog

Internet Archive and national web archives crawl sites: they open a page, save it and follow the links further. A data catalog is built in a way that makes this approach lose a lot.

- **An interface on top of an API.** The dataset list is loaded by a script from the programmatic interface, and a dataset has no separate page with a link to the file. The crawler sees an empty shell. This is how the only known open data portal of Afghanistan, on GeoNode, was built: it disappeared in December 2022, and all that remained of it in the web archive was the home page. The archive of the Russian Accounts Chamber's bulletins since 1996 also serves documents through a search API, and crawlers do not reach the PDFs themselves.
- **Large files and archives.** Internet Archive rarely saves ZIPs and multi-gigabyte files, and crawls pages with few links to them less often.
- **Closed to foreign addresses.** Since 2022 many Russian government sites do not respond to requests from abroad. In 2023 the sites of the Ministry of Justice and the Interior Ministry were not updated in Internet Archive at all, and the Ministry of Agriculture's site was not saved for about three years. If the archive cannot open a site, there will be no copy, even if the site works.
- **A ban in robots.txt.** The owner can close the site to the archive themselves. The "Map of Russian Science" disappeared this way in 2017: only the home page survived in Internet Archive.
- **One archive for everyone.** Almost the whole world relies on Internet Archive, which has no stable funding. In 2025 it had a months-long outage, and snapshots of many sites were not taken. In 2026 some publishers began blocking its crawler, fearing that companies training models would take their texts through the archive.

So a copy of a catalog is not just a snapshot of pages in WARC format. It is also a dump of descriptions through the API or a DCAT file, and the downloaded files. How the owner does this before closure is described in the section "How to take a snapshot" in the chapter on lifecycle. The same order works for someone else's catalog.

How it is done right

A national archive keeps the state's sites. In the UK, the National Archives' web archive regularly saves the sites of all government bodies. When the statistics office ONS redesigned its site and changed the system for storing series, the addresses changed but the data did not disappear: the time series tool page leads to a copy of the old site in the archive. A user with an old link in an article finds what they cited.

Closure with warning. In March 2017 Microsoft announced the closure of the CodePlex code hosting service: from October the site was read-only, in December it closed, and a migration guide was released for the closure. Nine months were enough for project owners and volunteers to save everything they needed. In the same year the Russian recommendation service Imhonet closed within a day, with no way to retrieve one's materials.

A copy with proof of provenance. In 2024–2025 the Harvard Law School Library Innovation Lab saved everything data.gov linked to: about 311,000 datasets, 17.9 TB. Each dataset is packaged according to the BagIt standard together with metadata from the API and a timestamped digital signature, so anyone can verify that a file has not changed since it was saved. The archive is in the Source Cooperative open cloud storage, it has search, and the code for such archives is open.

Even large services back down. In 2024 Google announced that from 25 August 2025 all goo.gl short links would stop working. ArchiveTeam volunteers began crawling them in advance. A few weeks before the deadline Google changed its decision and kept the links that were still being followed, switching off only inactive ones. A short link is not a persistent identifier: its fate is decided by the service owner, not by whoever put it into a document. More in the chapter on identifiers.

Who rescues when the owner did not

Government and university programmes. Since 2008 the End of Term Web Archive project has saved US federal government sites at every change of president. The 2024–2025 archive was collected by six organisations: Internet Archive, ArchiveTeam, Common Crawl, the Harvard lab, the University of North Texas Libraries and Webrecorder. That is 2.29 PB of compressed WARC files. They are in AWS open storage under a CC0 licence, and complex interactive sites can be viewed separately on GovArchive.us.

The community. ArchiveTeam is volunteers who save closing services with a distributed crawler. In 2025 the Data Rescue Project brought together research organisations that save data specifically, not pages, and keeps an open catalog of what has been rescued. Earlier, in 2017, DataRefuge and Climate Mirror appeared for the same reasons.

In Russia this is done by the National Digital Archive (ruarxive.org), a civic project, not a government service. In 2022 it ran a campaign to archive open data portals: by May it had collected 37 sites, 43 GB compressed; the largest snapshots were the Ministry of Culture's portal and data.gov.ru. In 2025, when the decommissioning of the EMISS statistics system was announced, the archive saved the sites of Rosstat, its regional offices and EMISS itself. Russia has no mandatory deposit of government bodies' sites and data in an archive.

All these archives share one weakness: they can be used only if people know about them. A copy with no record in any catalog is almost as unavailable as a deleted original. So archives are published together with descriptions, in the same format as a data catalog.

How to preserve a catalog you use

1. **Check whether there is an external copy.** Open the dataset address in Internet Archive and see when the last snapshot was and whether the files were saved, not just the page. Also check the site from another country: if it is closed to foreign addresses, there is almost certainly no external copy.
2. **Take the descriptions in full.** For CKAN this is a paginated walk through `/api/3/action/package_search`; for many portals, a single DCAT file or `/data.json`; for research repositories, OAI-PMH. Descriptions weigh little, and without them a file archive turns into a folder of incomprehensible names.
3. **Download the files you need.** Save alongside them the address the file was taken from, the date and a checksum. If data is served only through an API, an ordinary archiver will not take it: a paginated dump is needed, for example with the APIBackuper tool.
4. **Package it with a description.** For a set of files, BagIt or a Frictionless Data Package fits: the package holds both the data and a manifest. For site pages, WARC or its packaged variant WACZ, which opens in ReplayWeb.page without a server.
5. **Decide whether the copy can be published.** An open licence on the datasets allows publishing a copy in open storage and describing it in a catalog. But it covers only what the owner licensed. It does not cover other people's files that the portal merely links to, personal data or the terms of use of the site itself. A copy of a whole catalog may also touch the sui generis right in the database itself: in the EU and Russia it belongs to whoever invested in creating it. These parts are checked separately. If there is no licence, or it prohibits redistribution, the copy is kept privately and reported to an archive or to the owner.

What follows for an owner

If you run a catalog, the simplest thing is not to let it become the subject of a rescue operation:

- do not close the site to archive crawlers or foreign addresses if it holds open data;
- keep a full dump of descriptions and files at a persistent address: both an aggregator and an archive will take it;
- when stopping a series, do not delete what has already been published;
- announce closure in advance, months rather than days, and leave a read-only mode;
- if your country has a national web archive, arrange to deposit a copy, as UK agencies do.

Items for a checklist are in the section "Before switching off a catalog".

Sources

- Sunlight Foundation, Classifying changes to public access to information on US government websites, October 2017 (Internet Archive copy)
- Chronology of data removals in the US, National Security Archive, 30 March 2026
- End of Term 2024 Dataset: composition and size; End of Term Web Archive in the AWS open data registry; GovArchive.us

- Harvard Library Innovation Lab, Announcing the Data.gov Archive, 6 February 2025; archive search; the archive on Source Cooperative
- Data Rescue Project, ArchiveTeam
- Google, updated plans for goo.gl links, 1 August 2025
- UK Government Web Archive; the ONS time series tool with a link to the archived copy
- National Digital Archive (in Russian); code of the 2022 campaign: rudatarchive
- BagIt, RFC 8493, WACZ, ReplayWeb.page, APIBackuper
- I. Begtin, Telegram (in Russian): data.mosreg.ru, 3 June 2019; the Sunlight Foundation on ways information disappears, 9 October 2017; CodePlex and Imhonet, 29 April 2017; what cannot be web-archived, 25 December 2020; the portal archiving campaign, 20 May 2022; the Accounts Chamber archive, 30 November 2022; Rosimushchestvo, 17 February 2023; Afghanistan's portal, 5 July 2023; government sites and Internet Archive, 19 December 2023; ONS and the National Archives, 14 February 2024; archiving Rosstat and EMISS, 19 May 2025; the Internet Archive outage, 24 October 2025
- Catalog lifecycle, persistent identifiers

PART X.

Catalog Quality

Quality levels, metrics, methodologies and indices, and catalog self-audit.

1. Quality Levels
2. Quality Metrics
3. Methodologies and Rankings
4. Catalog Self-Audit

Quality Levels

In a task, "quality" means a level. There are four, and a high score on one leaves the others as they were. Before fixing or reporting, ask which level the assessment touches, and assign the work to whoever owns that level.

The number of datasets, a country index and a record score answer different questions. How to calculate shares is in the chapter on metrics. What each known methodology measures is in the chapter on methodologies. How to walk through your catalog in one go is in the chapter on self-audit. This chapter is about how to split an assessment into levels.

Four levels

Level	Question	What it is usually measured by	Who fixes it	What remains unchecked
Data	can the file or layer be used	format, completeness, agreement with the source, reproducibility of a calculation	the data owner	record completeness, site availability, country index
Descriptions	can the dataset be found and its terms understood	field completeness, conformance to a profile, a live link	the publisher and catalog editor	number of rows in the file, country policy
The catalog as a service	does it open and do machines receive it	availability, API, dump, licences in the dump, persistent addresses	the platform administrator	file contents, country index
The system of catalogs	is there a chain, and do the licence and source travel along it	openness indices, the share of a source's datasets on the upper portal, duplicates	the open data coordinator	a single city file

An empty file with a filled-in record is good descriptions and empty data. A live site without a dump is a good showcase and a weak service. A country with a strong national portal and dead city ones is a strong top and a weak system. The reverse happens too: a city's file and record are in order, but on the upper portal the licence is lost. This is how a dataset from the city of Malmö went in 2024; the chain is analysed in the chapter on federation.

One object, different scores

The portal looks alive, the files are empty. In February 2023 data.gov.uz had 6,623 datasets, more than most portals in the region. The monitoring section showed almost all datasets as updated. At the service level it looked alive. At the data level it did not: 96% of datasets had fewer than 200 rows, 42.6% had a single row, and the whole portal in CSV took up 40 MB. The records, meanwhile, could have been filled in. A description score sees filled-in fields and leaves empty rows unnoticed: they become visible once the files are opened. How to calculate the distribution by rows, and a second example, data.gov.ru in 2025, are in the chapter on self-audit.

There is a badge, but the calculation does not reproduce. Reproducibility is the ability to obtain the stated result again from the published data and the recorded steps. An "open data" badge on a journal article says the file has been published. In 2021 Hardwicke and co-authors checked 25 articles in Psychological Science with such a badge from 2014–2015. Without help from the authors, the checked figures matched in 9 articles out of 25. After the authors'

explanations, in 15 of 25. The main reason for the discrepancy lay in the work itself: the analysis steps were recorded unclearly, although the badge already confirmed the data was published. Whether the calculation matches the article text is checked separately.

A record score looks at the record. data.gouv.fr shows the publisher a metadata quality score: description, files, frequency, licence, open format, territory, period. An empty file with filled-in fields gets a high score. What goes into the score and how it looked on 14 October 2026 is in the chapter on good practices. This is the descriptions level.

What counts as good

A good answer depends on the catalog type. The six practices without which a dataset cannot be found or used are in the chapter on good practices. Below is how "data" and "descriptions" differ between neighbouring types and where a level breaks first.

Type	Data is good when	Descriptions are good when	Where it usually breaks first
Open data portal	the file opens in a program and has rows and a column dictionary	licence, publisher, period and territory are on the dataset	the counter grows through empty and sliced tables
Geoportal	the layer downloads as a file or through a WFS service	the layer has a licence, a date and a description	the municipal map: the layer is visible, there is no download button
Research repository	the calculation reproduces from the files and recorded steps	there is a DOI, authors, a licence and a citation for an article	the badge and DOI are there, the analysis steps are not recorded
Indicator catalog	the series continues and can be taken in full	methodology sits next to the series, codes come from a shared code list	the API responds, but the series has stopped
Microdata catalog	the file can be read using the variable dictionary	the description is open, the access level is named	the file was closed together with the record
ML catalog	the file downloads, the training and test split is fixed	the record has a licence and provenance	the record is filled in, the licence is more permissive than the source data's
Metadata catalog	a vocabulary or code entry is complete and points to the current version	the entry can be found by code and topic	the record is alive, but the catalog it points to is switched off
Page, Git or hub	the file opens without a login, encoding and columns are stated	there is a README, a licence, schema.org markup, a date and a contact	there is a repository but no page with a persistent address

For microdata, a closed file with an open record is a normal level if the access procedure is written down. The link breaks when the description is hidden too.

Failures are analysed in the chapters on open data portals, geoportals, research repositories, indicators, microdata, ML catalogs and metadata catalogs.

Checking on entry

Everything above is measured after publication. It can also be done before: the catalog does not accept a dataset that has not passed a check. Then quality is a condition of entry, not an indicator on a dashboard.

This is how neuroscience data archives work. OpenNeuro accepts only datasets following the BIDS standard: a shared order of folders, file names and descriptions. Upload is possible with validator warnings, but not with errors. In DANDI the publish button becomes active only after all errors in the check of descriptions and files have been fixed. A published version does not change.

Checking on entry works where a field has a shared standard. Otherwise it becomes one portal's list of rules, and publishers get around it with empty fields. For tables on an open data portal the analogue is checking a file against a schema on upload. Schemas for common tables are already collected in the schema.data.gouv.fr catalog. The level such a check holds is data. Fixing the file after it is still the data owner's job, but before publication rather than after a complaint.

Other innovations of the same kind are in the chapter on innovative catalogs.

Where to put someone else's score

Someone else's assessment is first placed on one level. Only then is it decided whether to fix your catalog by it.

Assessment	Which level it touches
Five stars	data: the format of one dataset
FAIR and F-UJI	descriptions of one research record and the access procedure
MQA, the data.gouv.fr score	descriptions of catalogs that already send records upward, and whether links are alive
CoreTrustSeal	the catalog as an archive: procedures and preservation
Open Data Maturity, ODIN, Global Data Barometer	the system of catalogs and country policy
Number of datasets	a record counter
The registry trust score	a rough view from outside: is the site alive, is a licence stated

MQA sees catalogs that already send descriptions to data.europa.eu and averages the records the assessment has managed to count: this may be the catalog's share in search. A country index is built from policy and national portals. Formulas and the limits of each methodology are in the chapter on methodologies.

In what order to fix

From the bottom up, following dependencies. First, files that open and can be read. Then the licence and mandatory fields. Then the dump and persistent addresses: without them descriptions will not reach the upper portal and search engines. A country index moves when the lower levels are fixed in many catalogs at once.

A harvesting deadline from above moves the dump to the front of the queue, for the datasets you actually publish. Filling in records for single-row tables by that deadline means fixing descriptions where the data is broken.

The task is set at the level where things are broken. The record editor is given empty fields. The data owner, the file and its freshness. The platform administrator, the dump, the API and addresses. The country coordinator, the rules by which cities send descriptions, and answers for an index such as Open Data Maturity: that score is collected from policy and national portals. A single CSV stays with the data owner.

Example: an agency was told to raise quality

Context. A regional open data portal. In monitoring, almost all datasets are marked as updated. The country's national index is middling, and a contribution is expected from the region. A user sent in a budget: a single-row table in PDF. The editorial team is two people.

Options. Fill in empty fields in all records. Rewrite the API and dump. Prepare answers for the country index. Open a sample of files, fix the format and content of the datasets people use, then the licence and mandatory fields, then the dump.

Criteria. A person opens the file and uses it without manual fixes. A program takes the descriptions without a login. The work goes to whoever owns the level. Two editors can manage it in a quarter.

Decision. A month goes on the data of the datasets that appear in emails and in download statistics: open format, rows, column dictionary. The next step is the licence and mandatory fields on the same sample, then a dump of all descriptions. The country index is not in the editorial plan.

Why. Filling in all records raises the description score and leaves the PDF in place: it fails the "the file can be used" criterion. A new API over empty files serves the same files faster. Answers for the index are written at country level and do not fix a regional file. A sample with live files resolves the complaint and fits the work of two people.

Consequences. For another quarter the catalog will look more modest than a neighbour with thousands of records. The dump will appear after the files, and until then the federation will not harvest this portal.

When to review. The sample of files opens in a program, and the national portal has named a harvesting date: then the dump becomes the first task. Or there are no more emails about content, but half the records still have empty fields: then the queue moves on to descriptions.

How to write your own decision record is in the chapter "Justifying a Decision".

If there is no portal

The same four levels exist for a page, a repository and a dataset in a shared hub.

- **Data**: the file itself: it opens, the columns are named.
- **Descriptions**: README, licence, `datapackage.json` or schema.org markup.

- **Service**: the page opens without a login, and the address does not change when the title is edited.
- **System**: a record in a catalog people already search: a national portal, Zenodo, re3data. GitHub search does not replace this record.

The order is the same: first a file that can be used, then a description, then a persistent address, then a record one level up. The minimum for a page is in the chapter on findability without a portal. How a short pass differs from a portal audit is in the section "If there is no portal".

How to set the task

The task states the level, the scope and who does it.

A good wording looks like this: "On a sample of 30 datasets from different publishers, the file opens as CSV or GeoPackage without manual fixes. Deadline: one month. Done by the data owner, checked by the editor."

The result of a pass is one page from the chapter on self-audit. The queue on it is already split: what the record editor fixes and what requires a change to the dump or address.

Sources

- T. E. Hardwicke et al., Analytic reproducibility in articles receiving open data badges at the journal Psychological Science: an observational study, Royal Society Open Science, 2021: 25 articles with an open data badge from 2014–2015; without help from the authors the checked figures matched in 9, after the authors' explanations in 15
- 5-star open data
- MQA, MQA methodology
- Open Data Maturity, Global Data Barometer, ODIN
- CoreTrustSeal
- data.gouv.fr: guide to the metadata quality score (in French)
- I. Begtin, What is wrong with Uzbekistan's open data portal? (in Russian), February 2023: dataset sizes on data.gov.uz
- Metrics, methodologies and rankings, self-audit
- Good practices, federation, making a publication findable
- OpenNeuro, user guide: BIDS validation on upload; DANDI, publishing Dandisets: publication after all validation errors are fixed; the schema.data.gouv.fr schema catalog

Quality Metrics

A metric is useful if it shows what to fix. For that, several shares are calculated. Each share has three notes: what is in the denominator, what value counts as filled in, and on what date it was calculated. The number of datasets in the site footer has none of these notes.

What a share is made of

The denominator is the datasets from a named list: the whole dump, every Nth dataset, or an API page with a stated offset. The share is calculated from the number of datasets in that list. Thirty records from different publishers catch a typical gap. How to take them and where to record them is in the chapter on self-audit.

"Filled in" comes in three degrees. The report says which was counted. Otherwise two shares with the same name cannot be put side by side.

Degree	What passed	What may still be empty in meaning
The field is not empty	something is in the licence	the value "not specified": that is a string too
A value from a list	a known licence, a format from a code list	the period "1900–2099", a description copied from the title
The value is true	the period looks like the data, the link opens a file	this is checked by eye on a few records

The first degree gives a high share where an empty field was replaced with a placeholder. The second catches the placeholder if it is on a list of forbidden values. Automated checks hardly cover the third: a profile validator and MQA check that the field is there. On data.gouv.fr the built-in assessment counts the licence "not specified", and on data.gov.gr a placeholder period is on almost every dataset in the sample and goes into the DCAT dump. Both cases are analysed in the chapter on metadata depth.

One API page is not a sample

A CKAN portal serves records in pages. A page often turns out to be one harvesting batch rather than a cross-section of the catalog. On 14 October 2026 the data.gov.gr API showed 22,765 datasets. During the checks the counter moved by one or two records. The command below takes 200 datasets ordered by record creation date and calculates four numbers: how many organisations are on the page, how many have an explicit licence, how many have at least one file from a short list of formats, and how many records changed after 14 October 2025.

An explicit licence here is the second degree: the `license_id` field is not empty and is neither `notspecified` nor `none`. The command counts the code `other-open` ("some other open licence"). A stricter check looks at whether the licence has a link to its text. The short list of formats is CSV, JSON, XML, GeoJSON, GeoPackage, Parquet. The command reads the format label, not the bytes: ZIP and XLSX are not on the list even if there is a table inside. For a geoportal the list is different. When a file is fit for a program at all is in the chapter on formats.

```
PORTAL=https://data.gov.gr
START=0
CUTOFF=2025-10-14
curl -s "$PORTAL/api/3/action/package_search?
rows=200&start=$START&sort=metadata_created%20asc" \
  | jq --arg cut "$CUTOFF" '.result.results as $r | ($r | length) as $n | {
      datasets: $n,
      organizations: ([$r[].organization.name] | unique | length),
      explicit_license: ([$r[] | select((.license_id // "" | ascii_downcase) as $l
                          | $l != "" and $l != "notspecified" and $l != "none")] | length),
      open_format: ([$r[] | select(any(.resources[];
                     (.format // "" | ascii_upcase | gsub("\\."; "")) as $f
                     | ["CSV","JSON","XML","GEOJSON","GPKG","PARQUET"] | index($f)))] |
length),
      card_modified_since_cutoff: ([$r[] | select((.metadata_modified // "") > $cut)] |
length)
    }'
```

`CUTOFF` is the date one year before the day of the check. Substitute your own. Change `START` and run the command several times.

Three runs that day:

Offset	Organisations	Explicit licence	Open format	Record changed within a year
0	75	67	18	200
8000	14	3	57	200
16000	1	0	200	200

All numbers are out of 200. The second page has 14 organisations, and 165 of the 200 datasets came from a single harvest, `data-apdkritis`: the records were created on 14 March 2026. The third page is a single organisation, `keratsini-drapetsona`; all 200 records were created on 22 May 2026, and the harvest source is called `gis-keratsini-drapetsona-wms-only`. The format field made the short list for all 200, while none has an explicit licence. The share "open format 100%" describes this batch, not the portal.

Sorting by identifier mixes publishers. Three pages of 200 at offsets 0, 8000 and 16000: 70, 80 and 76 organisations; an explicit licence for 39, 41 and 36; an open format for 116, 110 and 115. The spread is smaller, but these are still three pages, not a census of 22,765 datasets. Large publishers take up a noticeable share of every page. For a report, take every Nth dataset, as in the self-audit, and write down the selection rule.

The same day the open.canada.ca API showed 48,083 datasets. Three pages by identifier, 200 records each: an explicit licence for all 200, an open format for 88, 83 and 92. On these pages the licence no longer distinguishes the catalog from itself. Look at a share that can still fail.

The date in the command also lies easily. The `metadata_modified` field within the last year is filled in for all 200 records in all three Greek slices, including the batch from one municipality. By this field the portal looks completely fresh. For the `dedo` dataset, `metadata_created` dates the record to 17 November 2025, while the file's `created` field says 15 June 2018. After a portal

migration, the record's birth date is the migration date. Data freshness is judged by the file date or the promised frequency. How a portal itself calculates when the next file is due, using HDX as the example, is in the chapter on good practices.

On a repeat run the page boundary may shift by a record: datasets keep being added to the catalog. The report records the date, the offset and the sort order.

What to count

For a city or agency portal, five numbers are enough. They go on the summary page from the chapter on self-audit.

Number	How to count	When it is already a queue
Explicit licence	the share of datasets where the licence is a link to a known text, not an empty field and not "not specified"	less than half the sample
Live file	the share of datasets where the link opened a file	a noticeable share leads to a site page or does not respond
Open format	the share of datasets with at least one file from a list you have written down	the list is not written down, and the share cannot be repeated
File freshness	the share of datasets whose file date fits the promised frequency	only the record has a date
Full dump	yes or no: all descriptions open without a login	there is no dump. This is not a share but a threshold

The thresholds for "good" and "bad" are set for your own task. A licence on fewer than half of the sample is already a queue for fixing, not a reason to argue about weights.

Alongside these five, keep a separate list that is not mixed into the same share.

What else	What is recorded	What the number does not see
Completeness of other fields	publisher, contact, period, territory: each as its own share with its own degree of "filled in"	copied text, unless it was looked for separately
Interoperability	whether the dump passes the validator for its profile	the truth in fields that are formally present
Use	downloads, API calls, links from articles, together	file quality. A large number of views does not replace it
Content	the share of short files and the total volume	this is another quality level. The procedure is in the substantive check

The unit in the denominator depends on the type. For tables it is a dataset, for an indicator catalog a series, for a geoportal a layer, for microdata a study. What to count by files for each type is collected in the same chapter on self-audit.

What to count with

Fields and links are counted by a program. Whether a description matches a neighbouring record and whether a period is plausible is checked by eye, on the same thirty records.

The addresses below responded on 14 October 2026.

What it checks	Tool
The dump conforms to DCAT-AP	the DCAT-AP validator on the European Commission's testing platform
Layer descriptions and services conform to INSPIRE	INSPIRE Reference Validator: the European Commission closed the central instance on 1 April 2026; the code and test suites are open, and the validator is deployed in-house on ETF
schema.org markup on a dataset page can be read	Schema Markup Validator and Google's Rich Results Test
A research repository record looks FAIR	F-UJI, by the record's identifier or address. The program does not read file contents
Records the European portal has harvested	MQA on data.europa.eu. A catalog that sends nothing there gets no score
File links open	an address check. A 200 response code does not yet mean the link leads to a file

A link check distinguishes four outcomes: the file opened, the server refused, no response, a site page opened. On 14 October, of a dozen links from one open.canada.ca page, nine succeeded, one was refused with 403, and two did not respond within 15 seconds. Of a dozen links from data.gov.gr, all twelve succeeded, and one led to an HTML section of the site. This is an acceptance check, not a share for the catalog. On CKAN an address may be a path from the site root, without `https://`. A check that requires the full site name will count a live link as empty. A soft failure, where the link opens but does not lead to a file, is covered under the term "broken link" in the glossary.

How the MQA formula works

MQA is an example of someone else's formula, not a mandatory score for your own portal. How to read a catalog's score and why it covers only the counted descriptions is in the chapter on methodologies. Here is what one term of it is made of. The text below was checked against the methodology section in a catalog report of 12 October 2026. The methodology page on the site is dated 6 October 2026.

The checks are labelled with the four FAIR words: findability, accessibility, interoperability, reusability. The previous methodology had a fifth dimension, contextuality. The current one does not. The weight comes from whether the field is mandatory in DCAT-AP, not from a separate FAIR exam: a mandatory field is 1, recommended 0.5, optional 0.25. A dataset, a distribution and a data service share the same ceiling, 7.5. A catalog's score is the average on the same scale. The bands on the scale: up to 2.5 is sufficient, 2.5 to 5 good, 5 to 7.5 excellent.

Most checks work the same way: the property is filled in or not. A placeholder passes such a check. A live link is the exception, and it is weaker than it seems. Having an access URL of the correct form weighs 1. The check that the address responds weighs 0.25, even when the field itself is mandatory. Only a 2xx response to a header request counts. The file is not downloaded. A redirect, through which portals often serve a file, does not count.

Conformance to the SHACL profile is not part of the score. It is checked by a separate validator from the table above. The methodology does not check file contents, the legal force of the licence or the truth of the fields.

Your own formula can be shorter: weight by whether a field is mandatory, plus a check that the link opened a file. There is no need to repeat the 7.5 scale. Before putting someone else's score in a report, read the formula. The record score on data.gouv.fr counts the licence field as filled in even when it says "not specified". How that score works is in the chapter on good practices.

What is easy to fake

The number of datasets. It is raised by cutting one series into files. A catalog of 50 honest datasets is more useful than a catalog of 5,000 empty records. If the number is still needed for a report, two others are put beside it: the share of short files and the total volume. How to calculate them is in the substantive check. Where the line runs between slicing and a separate object is in the anti-pattern "Fragmentation to boost the count".

How far the counter diverges from the volume can be seen from a single snapshot. On 2 February 2022 a full dump of data.gov.ru gave 26,490 datasets and 29 GB, mostly uncompressed. 6.8 GB of that was a single snapshot of the address register, not updated since October 2019. 3,839 datasets, 14.5%, weighed less than a kilobyte. The compressed archive of the same snapshot is 14 GB: in other chapters this is the same data, already packed. On 14 February 2022 HDX, with a comparable number of datasets, about 31,000, had a downloaded portion of 6,600 datasets, 21%, that took up 205 GB. The estimate for the whole catalog from this portion is 950–1,000 GB; it is an estimate from an unfinished download. In the same download, data.gov.uk's uncompressed CSVs alone took up 222 GB, and that is not the whole portal. What remained on data.gov.ru after it reopened in 2025 is in the chapter on self-audit.

So an indicator such as "publish N datasets" without a list of datasets measures nothing: such a figure is collected in a day from short tables. A meaningful target is a list of specific datasets, for example high-value datasets, and the share of them published.

Home page views. They do not see those who take the dump. If the main entry point is the API, count that too. How not to turn statistics into an annual figure for its own sake is in the chapter on other capabilities.

A score for a field that is formally present. A validator and MQA check presence. They do not check that the description is not copied from a neighbouring dataset or that the period is not a placeholder. Viewing a few records is added to the automated assessment.

A date moved by a migration or by monitoring. Editing a record and reloading the catalog update `metadata_modified` without updating the rows. A monitoring section may mark as "updated" a dataset consisting of a single old figure. This was the case on data.gov.uz. The analysis is in the self-audit, in the section on where a pass deceives.

Signs of life

A separate task is to understand whether a portal is alive at all. For someone else's catalog you plan to rely on, and for your own if nobody has opened it for a long time, five checks are enough. They are done before a full download of the files.

- the date of the last **new** dataset, and not the date of the portal migration: an old file may have yesterday's date in its record;
- the date of the last news item or changelog entry;
- whether there are series with a promised frequency and whether the file date matches that frequency;
- the share of links that open a file rather than a page;
- whether sections that used to exist have disappeared: the publication plan, usage statistics, data requests.

When the visitor statistics page disappeared from data.gov.ru in spring 2022, and data.mos.ru's publication plan had already been removed in 2020, this was more telling than any ranking. For government portals in 2020–2022 there was a simple historical test: did they publish COVID-19 data while it was needed? Thailand's data.go.th updated such statistics every day; Russia's data.gov.ru did not. Today the same question is asked of a series someone still needs on schedule: a budget, an exchange rate, a weekly operational dataset. A portal on which such a series is updated on the promised day is alive regardless of its dataset counter.

Sources

- MQA methodology: page of 6 October 2026; the methodology section in a catalog report of 12 October 2026: four FAIR dimensions, weights 1 / 0.5 / 0.25, scale 0–7.5, a link check by 2xx code, SHACL not part of the score
- DCAT-AP validator, F-UJI: the addresses opened on 14 October 2026; INSPIRE Reference Validator: JRC announcement of the closure of the central instance, 1 April 2026, ETF and executable test suites for your own installation
- Schema Markup Validator, Dataset markup in Google's documentation
- CKAN 2.12 Action API
- data.gov.gr and open.canada.ca APIs, 14 October 2026: slices of 200 datasets, sorted by record creation date and by identifier
- I. Begtin, Telegram (in Russian): a full dump of data.gov.ru, 2 February 2022; comparing portal volumes, 14 February 2022; the "500 datasets" indicator, 21 March 2025; data.mos.ru running on inertia, 17 January 2022; the status of portals in spring 2022, 5 May 2022; COVID-19 data as a sign of a live portal, 15 December 2021, and data.go.th, 17 July 2022
- Metadata depth, self-audit, methodologies and rankings, anti-patterns

Methodologies and Rankings

A score answers one question. France has 100% in the Open Data Maturity report for 2025. The same national portal had 4.27 out of 7.5 in the data.europa.eu record assessment on 11 October 2026, and the average included 211 descriptions, although in the European portal's search this catalog is listed with 76,548 datasets. Both numbers can be cited. Substituting one for the other cannot: a self-audit of one site also answers its own question, and it is covered in the neighbouring chapter.

The table first. The year is the release that can be cited now.

Methodology	Unit	What it measures	What stays out of view
Five stars (2010)	one file	a ladder of formats: from a licence to linked data	the portal, freshness, live links
FAIR (2016) and F-UJI	one record	fields, identifiers and access conditions	file contents and the open data portal
MQA	records harvested by data.europa.eu	DCAT-AP completeness and link responses	datasets not in the calculation, and rows inside files
A record score on your own portal	one record	the fields the portal itself has decided to check	other catalogs and file contents
CoreTrustSeal	a repository as an archive	procedures, preservation, rights	each file and a city budget
Open Data Maturity (2025)	a European country	policy, portal, quality control and impact, by questionnaire	a specific CSV
OURdata (2025)	the central government of an OECD country	data availability, accessibility and support for reuse	impact and a city site
Global Data Barometer, 2nd ed. (2025)	a country in Africa or Latin America	data for accountability, governance and skills	Europe in this edition
ODIN 2024/25	a statistical office site	coverage and openness of official statistics	the open data portal, unless the office links to it
The registry trust score	a catalog record	a rough filter for thousands of catalogs from outside	whether datasets are updated and whether the file opens

How to read someone else's score

Before putting a number in a report, answer four questions.

1. **The unit.** A file, a record, the records harvested by the upper portal, a country's policy or a statistical office site.
2. **Who filled it in.** A program, the country itself or an expert working from open sources.
3. **Which release.** If questions or weights changed between releases, last year's score is not added to this year's.
4. **How many records are inside the score.** An average over two hundred records and an average over the whole catalog are different statements. Write this number next to the score.

Next, how each family differs. A live analysis of three catalogs in the European federation is in the section on MQA.

One file

Five-star open data (Tim Berners-Lee, 2010) is a ladder for one dataset.

Star	What is already there
1	The file is on the web under an open licence. Any format, even PDF
2	It is a structure a program can read: a table, not a picture of a table
3	The format is open: CSV, JSON or XML, not a file from one paid program
4	Entities inside the data have addresses that can be referred to
5	Those addresses link to other people's data

One exemplary table earns five stars in a dead catalog. The ladder is good for a conversation about format. It does not work as a portal audit.

The Open Data Maturity questionnaire asks a country whether it uses this ladder or a similar model. The score goes to the national team's answer, not to a count of stars across files.

One record

The FAIR principles (2016) say that research data should be findable, accessible, interoperable with other data and reusable. These are requirements for the record and the access conditions. FAIR and "open" are different properties: a dataset with closed files can also comply if the description is open, the identifier persistent and the access procedure clear. For a city CSV the principles provide vocabulary. They do not replace a licence or a DCAT dump.

F-UJI gives an automated score to one record by its identifier or page address. The program looks at fields and identifiers. It does not read file contents. How to run a record through it, and what else to measure quality with, is in the chapter on metrics. Where FAIR is mandatory for a repository is in the chapter on research repositories.

Records harvested by the upper portal

MQA on data.europa.eu assesses the descriptions that portals send to the European catalog. A catalog that sends nothing there gets no score, however tidy it may be.

The catalog scale is 0 to 7.5. A field's weight is taken from whether it is mandatory in DCAT-AP: mandatory is 1, recommended 0.5, optional 0.25. Three parts of a description are checked: the dataset, the distribution and the data service. A catalog's score is the average over the datasets the assessment has counted, plus separate data services that sit in the catalog on their own. A deprecated distribution is not included in the average. A link counts as accessible if a header request gets a 2xx code. The file is not downloaded, and a redirect does not count. The methodology does not check file contents, the legal force of the licence or the truth of the description.

The checks are still labelled with the four FAIR words: findability, accessibility, interoperability, reusability. The weight, however, comes from whether a field is mandatory, not from a separate FAIR exam. The methodology page is dated 6 October 2026. Your own formula can be built on the

same principle: weight by whether a field is mandatory, plus a live link check. More in the chapter on metrics.

A score is read together with the number of descriptions in the calculation. On 14 October 2026, 200 of 215 catalogs in the MQA cache had a score. The calculation dates in this cache are 23–24 September and 11 October. The average catalog score is 4.75 out of 7.5, the median 4.91. That is an average of catalogs, not the quality of all European datasets.

Three catalogs in the MQA snapshot of 11 October 2026. The number of datasets in data.europa.eu search was taken on 14 October.

Catalog	Score	Descriptions in the calculation	Datasets in search
data.gov.uk	2.66	9,505	59,041
data.gouv.fr	4.27	211	76,548
dati.gov.it	6.69	61,693	61,748

Among the British records included in the score, not a single distribution had a direct file link. Nor a licence from the European code list. The licence field is filled in for 25.9% of distributions. An access URL is present for 99.9%: it is a mandatory field and a heavy one. The address itself responded for 35.4%. The dataset theme is filled in for 3.5%, the contact for 0.9%, period and territory for none. Hence 2.66 with almost ten thousand descriptions in the calculation. In the European portal's search the same catalog has 59,041 datasets, so this score also covers a share of the harvest. The UK does not take part in Open Data Maturity: that is a different scale and a different set of countries.

The Italian 6.69 rests on an almost complete harvest: 61,693 descriptions out of 61,748 in search were included in the calculation. Every distribution has a licence and a direct link. This is the top of the record scale. It still says nothing about the rows inside the files.

The French 4.27 was calculated from 211 descriptions. Alongside in search are 76,548 datasets from the same catalog, the "open platform for French public data", that is, data.gouv.fr. Citing 4.27 as an assessment of those 76,000 is not possible. First check how many records went into the average.

Commands for your own catalog, if it is already in data.europa.eu. The British identifier is `data-gov-uk`, the French `plateforme-ouverte-des-donnees-publiques-francaises`, the Italian `dati-gov-it`.

```
curl -s "https://data.europa.eu/api/mqa/cache/catalogues/data-gov-uk" \
  | jq '{score: .result.score, assessed: .result.counts.value}'
curl -s "https://data.europa.eu/api/hub/search/catalogues/data-gov-uk" \
  | jq '{listed: .result.count, home: .result.homepage}'
```

The first shows the score and the number of descriptions in the calculation. The second, how many datasets the European portal holds in search. If the second number is much larger than the first, the score describes a piece of the harvest.

In the Open Data Maturity report for 2025, five MQA metrics appear alongside the questionnaire as a pilot, one catalog per country. They are not included in the quarter of the country score. Nor are automated checks of the portal's home page: mobile layout, speed, security and accessibility. The "quality" dimension in that report is answers about whether the country monitors metadata. That is the next section.

A score on your own portal

A portal can show a publisher a record score before anyone above harvests it. On data.gouv.fr it includes the length of the description, file descriptions, frequency and keeping to it, an open licence, an open format, territory and period. The score has been experimental since 2022. How it works and what was covered on 14 October 2026 for the record of the catalog itself is in the chapter on good practices. This is a third number alongside the country questionnaire and MQA: the publisher sees it on their own site.

Archive

CoreTrustSeal certifies a repository as a place that can be trusted with data for a long time: procedures, preservation, rights. A city budget portal does not need an archive certificate. How many repositories hold it and what requirements the questionnaire includes is in the chapter on research repositories. In the re3data registry the certificate mark sits next to the record. It is a signal for someone choosing where to put data for years.

Country

Indices compare policy and the availability of data in a country. The score goes to the country, not to a city site. One limitation is shared: a high score leaves open the question of whether a specific file opens.

Open Data Maturity is an annual assessment since 2015; the eleventh edition came out in December 2025. Participation is voluntary. Since 2018 the score has consisted of four equal parts: policy, portal, quality and impact. In 2025 the questions were not changed; the last restructuring of the questionnaire was in 2024.

It is a survey of national representatives. Most questions are a choice from ready answers plus a link or an explanation. The data.europa.eu team checks that the explanation relates to the question. Since 2024 the questionnaire has been pre-filled with last year's answers, and what the country has changed and the new questions are examined in detail. A repeated "yes" is not re-checked against files.

In 2025, 36 countries took part: the 27 EU countries, three EFTA countries (Iceland, Norway, Switzerland) and six candidates (Albania, Bosnia and Herzegovina, Montenegro, North Macedonia, Serbia, Ukraine). The average score of all participants is 81%, one point above 2024. The EU-27 average is 86%, three points higher. Within the EU-27, policy (93%) and portal (85%) score highest, followed by quality (83%) and impact (82%). 21 of the 36 countries are above 80%. France scored 100% across the questionnaire's topics, Lithuania and Poland 98%. Among EFTA countries Norway

is highest (92%), among candidates Ukraine (97%). The biggest rises were Albania (23 points), Malta (19) and Germany (11). One country's score fell. Montenegro and North Macedonia have no comparison with 2024: they were assessed for the first time.

A jump of 23 points in a year is about the answers and examples a country was able to show. The "quality" dimension asks whether there is a procedure for updating metadata, whether licences and DCAT-AP are monitored, and whether the share of data that passes a chosen model such as five stars is known. It is not the same as an MQA score.

The answer "this EU rule does not apply to us" to questions about EU legislation gives full marks. This is how comparisons with EFTA countries and candidates work: their high percentage includes questions that were removed for them.

OURdata by the OECD measures the openness of government data in OECD countries. The 2025 edition was published on 16 February 2026 and is the latest as of October 2026. The data was collected by surveying central government bodies in 2025 and describes policy from 1 January 2023 to 31 December 2024. Three equal parts: data availability, accessibility and government support for reuse. The scale is 0 to 1. The index does not measure the impact of open data. The average score of OECD countries rose from 0.48 in the 2023 edition to 0.53. France (0.96) and Korea (0.95) are highest. There is no 2025 data for Denmark, Hungary and the US. The unit is central government; a city site is not part of the score.

Global Data Barometer looks at data needed for accountability: procurement, companies, land, political integrity, public finance, plus data governance and skills. The second edition came out on 29 May 2025. It covers 43 countries in Africa and Latin America and the Caribbean, 27 core indicators, and the period from 31 August 2022 to 1 September 2024. The first edition, May 2022, covered 109 countries and the period from May 2019 to May 2021. Scores from the two editions are not combined: the number of indicators, the weights and the counting rules changed. A European country has no score in the second edition. A figure from the first edition is a snapshot of 2019–2021.

ODIN (Open Data Watch) assesses official statistics. The 2024/25 edition is the seventh; it covers 197 countries, and the ranking page was updated on 21 July 2025. It looks at the national statistical office site and the official pages it links to. Coverage means whether there is data on 22 topics, with breakdowns, for a run of years and by territory. Openness means a machine-readable and open format, a description, a way to download a selection or everything at once, and an open licence. The average country score rose by 12.5% compared with 2022. Average openness is 59 out of 100: data that does not exist is not assessed for openness, and for data that does, the format and licence are often still closed.

Uzbekistan scored 83 in this edition: coverage 78, openness 87. That is the statistical office site. The open data portal data.gov.uz in February 2023 consisted 96% of short files; the analysis is in the chapter on self-audit. Two sites, two scores.

Old indices are useful as a history of the method. The World Wide Web Foundation's **Open Data Barometer**: the fourth and last full edition came out in May 2017 (period July 2015 to June 2016), and an edition for 30 governments in September 2018 (period July 2016 to September 2017). There have been no global editions since. Global Data Barometer grew out of this work and keeps its own

count. Open Knowledge's **Global Open Data Index**: the last edition is 2016/17, published in June 2017, 94 places. There was no 2018 edition; the site is archived. These scores cannot be cited as current.

A rough assessment in the catalog registry

The dataportals-registry calculates an optional trust score from 0 to 100. It is not a certificate. The score consists of five parts: the owner type (science above business, central government above local), the catalog type (a repository slightly above a portal; aggregators and marketplaces are marked down), the licence, a re3data record and certificate, and whether the site and API are alive. An empty licence description lowers the score. The total is clipped to the range 0 to 100.

The owner type weighs more than the rest: science gets 40 points, a community 5. The "federal government" type is not named in the score table, so such a catalog gets the same 10 points as an empty owner field. A low score for a federal platform may be a hole in the formula rather than a verdict on the catalog.

The score rewards filling in the catalog record. It does not see whether datasets are updated or whether the file opens. It is too rough for a self-audit of your own portal. For a first selection from thousands of catalogs from outside, it is a reasonable filter. The formula is in the registry documentation, linked below.

What to take for yourself

Situation	What to open
You are fixing a city or agency portal	a self-audit: files, records, dump. A country index is not applied to a single CSV
data.europa.eu already harvests you	MQA, and next to it the number of descriptions in the calculation. If it is far from the number of datasets in search, the score describes a piece
You run a research archive	FAIR for the record and, when the archive is trusted with data for years, CoreTrustSeal
You are responsible for a European country's open data	Open Data Maturity, to compare policy and portal with neighbours. File quality is still checked at home
You compare countries' official statistics	ODIN. That is the statistical office site, not necessarily the national portal
You are choosing someone else's catalog from thousands	the registry trust score as a first filter, then the signs of life from the chapter on metrics

A city portal does not raise a country index on its own. A country index does not open a city file. The task is set at the unit where things are broken.

Sources

- 5-star open data, Tim Berners-Lee, 2010; FAIR Principles; M. D. Wilkinson et al., The FAIR Guiding Principles, Scientific Data, 2016; F-UJI
- CoreTrustSeal; the number of certified repositories and the 2026–2028 requirements are in the chapter on research repositories

- MQA, methodology as of 6 October 2026; the catalog cache and data.europa.eu search on 14 October 2026, score calculation dates in the cache 23–24 September and 11 October. Catalogs: `data-gov-uk` (2.66 of 7.5 from 9,505 descriptions, 59,041 in search), `plateforme-ouverte-des-donnees-publiques-francaises` (4.27 from 211 descriptions, 76,548 in search), `dati-gov-it` (6.69 from 61,693 descriptions, 61,748 in search). Average catalog score 4.75 of 7.5, median 4.91, 200 of 215 have a score
- Open Data Maturity, 2025 report: 36 countries, EU-27 86%, all participants 81%, France 100%; 2025 methodology: four dimensions at 25% each, a pre-filled survey, pilot MQA metrics not part of the score
- OECD, Digital Government Index and OURdata Index: 2025 Results and Key Findings, 16 February 2026: OECD average 0.53, no data for Denmark, Hungary and the US; previous edition OURdata Index 2023, 22 December 2023, average 0.48
- Global Data Barometer, second edition, 29 May 2025: 43 countries; first edition, May 2022: 109 countries; comparing the editions
- ODIN 2024/25: 197 countries, average score up 12.5% on 2022, average openness 59; ranking, updated 21 July 2025: Uzbekistan 83 (coverage 78, openness 87)
- Open Data Barometer: 4th edition, May 2017; leaders edition, September 2018. Global Open Data Index: archived, last edition 2016/17, June 2017, 94 places
- Trust score in the registry
- Metrics, self-audit, quality levels, good practices

Catalog Self-Audit

A self-audit is one pass through a catalog, after which it is clear what to fix in the coming month. The result fits on a page: when it was checked and against which list, a few shares, five records with typical errors and two queues. One is for the record editor, the other for whoever is responsible for the platform.

A country ranking and an archive certificate answer other questions. Their place is covered in the chapters on quality levels and on methodologies. This chapter gives the order for your own portal and for someone else's that you plan to rely on.

A full pass is done every six months and on the day the platform or address changes: old links break immediately. Once a month it is enough to open a dozen known links. Quarterly and annual platform checks are in the chapter on lifecycle. Two people look. Rows and records: whoever is responsible for the data. The dump, addresses and version: whoever is responsible for the platform.

Two scales

A sample of 30 datasets catches a typical record error: no licence, a placeholder period, a file that does not open. The share of short files, the total volume and the year of publication are calculated from a dump of the files, preferably all of them. Below are two portals where the records looked alive, and counting the files showed one-line notes.

Scale	What it answers	When
30 records from different publishers	a typical description error and a broken link	every full pass
All files or every Nth, several hundred	distribution by size, total volume, file year, gaps among the main publishers	when the dataset counter has become the main number in reports, at portal acceptance and after a sharp rise in that counter

A share from thirty is recorded as a share of the sample: "of 30 datasets, 20 have no licence". If the same gap is in most of the thirty, fix the rule for filling in records, not those thirty records.

How to take thirty

They are taken from the full dump, not from the first page of the site and not from a "recently updated" block. Listings start with what the editors have just fixed. A convenient method is to sort records by identifier and take every Nth, where N is the number of datasets divided by 30. For a catalog of 6,000 records that is every two-hundredth.

Publishers must vary. If nine-tenths of the catalog is one agency, thirty records will tell you about that agency. Then a few datasets from the long tail are added to every Nth, and the "scope" field says so.

If there is no dump, the same thirty are collected by browsing the general list on the site. The first line of the result says "no full dump". The other metrics are still calculated without it, but the catalog is no longer fit for federation.

How to count licence, date, period and open format, and which validators to run the profile through, is in the chapter on metrics. Profile errors (DCAT-AP, INSPIRE, DataCite) are kept in a separate list from empty fields: a validator checks that a field is present and is silent if neighbouring records contain the same text.

Steps

1. Find the catalog in the relevant registry, if it is there, and check its address and type. A wrong record confuses those looking for the portal. Record both the number of datasets on the site and the number in the dump or API: they often differ.
2. Open the dump of all descriptions and the page of one dataset without a login. Registration, a captcha or an "I am not a robot" check mean there is no machine access. Commands for the sitemap, `data.json`, API, OAI-PMH and CSW are in the live portal checklist.
3. Take 30 datasets as above and count licence, date, period and whether the file opens. Compare the date in the record with the date inside the file. A field such as CKAN's `metadata_modified` is the date the record was edited.
4. Run the dump through the validator for your profile. The list of tools is in the chapter on metrics.
5. If the question is about volume, count the files as in the next section.
6. Write down the five to ten main publishers for the topic or country and mark whether their data is present. An empty cell against the statistical office or the treasury matters more than a hundred small datasets.
7. Check the result against the requirements checklist. An item you yourself marked as mandatory and have not met goes to the front of the queue.
8. Set aside five records from different publishers, with links. Five errors from one agency speak about that agency, not about the catalog.

Someone else's catalog is walked through the same way. The queue is then shorter: take it, go to the original source, or postpone. Before downloading files it makes sense to go through the signs of life: a dead portal is not downloaded in full.

Substantive check

The four questions are the same for any catalog. What changes is the unit being counted.

Type	What to count	Where to find the commands
Tables, an open data portal	rows, compressed volume, file year	this section
Indicator catalog	series length and whether there is a dump of the whole database	indicator catalogs
Geoportal	the layer serves features, not just a record	geoportals

Type	What to count	Where to find the commands
Research repository	there is a file behind the record page	research repositories
Microdata	a study has variables and access conditions filled in	microdata
ML catalog	the file opens without a login, the training licence is named	ML catalogs

For tables the questions are these.

1. **Size.** Sort the files by number of data rows: empty or one row, up to 100, 100 to 10,000, more. If most are in the first group, the catalog consists of short notes, or one series has been cut into pieces. Where the line runs is in the anti-pattern "Fragmentation to boost the count".
2. **Volume.** Add up the file sizes and note whether they are compressed. Compare like with like, and with one known source on the same topic: a statistical database, an agency register, a city portal. A country's catalog that is smaller than one of its information systems does not cover the topic.
3. **Age.** The year is taken from the file or from the dataset's first-publication field, not from the record edit date. A new portal in which almost everything was first published many years ago has carried over the old.
4. **Coverage.** The publisher table from step 6. It is filled in from organisation names in the dump, not from memory.

For a folder of CSVs the first count is done like this. `wc` also counts the header row, so a file with one data row gives 2. The threshold "no more than 2" catches exactly the empty and single-row files. At the 100 and 10,000 thresholds a single header row does not change the group.

```
for f in data/*.csv; do wc -l < "$f"; done \
  | awk '{ n = $1
           if (n <= 2) a++
           else if (n <= 100) b++
           else if (n <= 10000) c++
           else d++ }
         END { print "empty or 1 row:", a+0,
                     " up to 100:", b+0,
                     " up to 10,000:", c+0,
                     " more:", d+0 }'
```

`+0` is needed so that an empty group prints as zero. A file with a CSV extension that opens as a site page is not included in this count: it is a broken link. The total size of the folder as it sits on disk:

```
du -ch data | tail -n 1
```

If you publish ZIP or gzip, the result records the archive size, not the unpacked folder. For XLSX and JSON, rows are counted with the program used to open the file. The thresholds are the same.

data.gov.uz, February 2023 and July 2025

In February 2023 data.gov.uz had 6,623 datasets, more than most portals in the region. The formats were machine-readable, the fields had descriptions, and the monitoring section marked 6,120 datasets as updated. A CSV dump showed something else: 2,823 datasets (42.6%) consist of a single row, 6,381 (96%) are shorter than 200 rows, and the whole CSV takes up 40 MB. The largest file is 60,984 rows and 4.4 MB in CSV. The dataset "Eggs produced, total for Khorezm Region" is a single figure for 2020; in monitoring it was listed as updated on 26 December 2022.

Alongside, in the same weeks, data.gov.kg had 646 datasets and about 570 MB of files, mostly Excel. Ten times fewer datasets, more volume. The formats differ, so this is a comparison of order of magnitude, not of exact bytes.

In July 2025 a repeat count (the post gives the address as data.egov.uz): 10,412 datasets, 5,207 single-row, half; more than 1,000 records for 114, just over 1%. The uncompressed JSON dump is 426 MB; for 2023 the same post gives 284 MB of JSON. This figure is not put next to the 40 MB of CSV: the format is different. The number of datasets grew; the slicing rule stayed.

data.gov.ru, July 2025

On 15 July 2025 the portal was reopened. The previous version had been closed in 2023 "for transfer to a new platform". Through the API there were 5,826 datasets; on the site pages, fewer. The compressed ZIP is about 100 MB, about 1 GB uncompressed. Of these 5,826 datasets, 4,694 (80%) are shorter than 100 rows, 586 (10%) have 100 to 500, 350 (6%) 500 to 10,000, and 62 (about 1%) 10,000 rows or more. About 3% could not be downloaded. By the first-publication field, 84% of datasets date from 2019 or earlier. The mark "updated in 2025" does not change the rows in the file: the post still suggested opening such datasets by hand.

The snapshot of 2 February 2022 is 26,490 datasets and 29 GB, mostly uncompressed. The packed archive of that snapshot is 14 GB; the 2025 analysis rounded it to "more than 24,000 datasets". It did not return to the new portal. For scale, from the same July 2025 analysis: the EMISS dump is 6,905 indicators and 4.36 GB of compressed CSV; a snapshot of data.mos.ru in November 2022 is 988 datasets and 3 GB compressed. No data from Rosstat or the financial agencies was found on the new portal.

Both analyses counted all files. A sample of descriptions on such a portal does not reveal this failure.

What to write in the result

One page. It can be handed to an editor without retelling. A field that was not checked is filled in with the words "not counted", not a dash: a dash is later read as "all fine".

Field	What to record
Date and scope	when it was checked; all datasets or a sample, from which list, whether files are compressed
Licence	the share of the sample where the licence is a link
Live link	the share of the sample where the file opened
Full dump	yes or no; whether it opens without a login; whether the count matched the site

Field	What to record
Content	the share of empty and single-row files, the total volume marked "compressed" or "as stored"
Five records	typical errors, with links
Queue for the month	what the record editor and data owners fix
Platform queue	what requires a change of extension, dump or address

This is what a completed page looks like for the February 2023 pass. Licence and the share of broken links were not counted in that analysis, and the table shows it.

Field	data.gov.uz, February 2023
Date and scope	February 2023, all 6,623 datasets, uncompressed CSV
Licence	not counted
Live link	the catalog was downloaded in full; the share of link errors was not counted
Full dump	existed, since the files could be retrieved
Content	42.6% a single row; 96% fewer than 200 rows; 40 MB of CSV
Five records	"Eggs produced…" for Khorezm Region: a single figure for 2020, updated in monitoring on 26 December 2022
Queue for the month	stop cutting the indicator by region; monitoring must look at the rows of the file
Platform queue	the file named RDF is the same XML, without a vocabulary

A finding of "the platform cannot do what is marked mandatory" goes into a decision record. The next pass checks the failed lines: has the single-row problem gone, does the dump open.

Where a pass deceives

- **Your own monitoring.** The "updated" section on data.gov.uz dated a record containing a single old figure. Freshness is checked by the file.
- **The first page and "recently updated".** They hold datasets that have just been touched.
- **The record date.** Editing a description updates `metadata_modified` and does not update the rows.
- **The number of datasets in the result.** It is raised by slicing. The share of short files and the volume are needed alongside.
- **Compressed next to uncompressed.** The 29 GB data.gov.ru dump and the 14 GB archive of the same snapshot are one catalog. So are 40 MB of CSV and 284 MB of JSON.
- **Five records from one publisher.** That is a diagnosis of the publisher.
- **A green validator.** The profile passes while the file is empty. Content is checked separately.
- **A share without the word "sample".** Thirty records are not a census of a catalog with tens of thousands of entries.

If agents read the catalog

If the catalog has an MCP server switched on, or AI agents rely on it, one more step is added to the pass. It is taken from the Open Knowledge Foundation pilots of 2026: users' frequent questions become tests.

1. Write down the ten questions that most often arrive by email or at support. For example: "how many schools are in the district", "where is last year's budget spending".
2. For each, record the correct answer, the dataset and a link to the record. Take the number from the file, not from memory.
3. Ask the same questions to an agent connected to the catalog's server.
4. Mark three outcomes: the right number with a link to the dataset, the right number without a link, wrong or invented.

A wrong answer most often points to an empty description, not to the model. In the OKFN pilot the model attributed "climate factors" to Uruguay's energy balance, which are not in the dataset. What helped was field definitions, units, period and known limitations in the description. A line "Agent answers: correct with a link N of 10" is added to the result of the pass. The test is repeated after every server update and after a major data revision. How the server works is in the chapter on AI.

If there is no portal

A page, a Git repository and a dataset in a shared hub are checked against the minimum for publishing without a portal: the page opens without a login, the licence is in place, schema.org markup passes the test, its description is at least 50 characters long, the date and contact are filled in. There is nothing to check against a catalog registry and no profile validator to run. If the dataset must get into a regional or national portal, a file in the profile that portal harvests is added to the minimum.

The result page is the same. In fields for which there is no catalog, write "no portal". The queue is the unmet items of the minimum.

Sources

- Quality metrics, quality levels, methodologies and rankings
- Requirements checklist, live portal checklist
- Anti-patterns, lifecycle
- Open Knowledge Foundation, MCP for Open Data Portals: Trust Depends on Understanding the Data, 10 June 2026: partners' frequent questions as tests, "climate factors" in an answer about Uruguay's energy balance
- I. Begtin, What is wrong with Uzbekistan's open data portal? (in Russian), 13 February 2023: 6,623 datasets, 42.6% single-row, 40 MB of CSV, monitoring
- I. Begtin, Kyrgyzstan's open data portal, Telegram (in Russian), 11 February 2023: 646 datasets, about 570 MB

- I. Begtin, a repeat count of data.egov.uz, Telegram (in Russian), 10 July 2025: 10,412 datasets, 5,207 single-row, uncompressed JSON 426 MB against 284 MB in 2023
- I. Begtin, a full dump of data.gov.ru, Telegram (in Russian), 2 February 2022: 26,490 datasets, 29 GB uncompressed
- I. Begtin, The updated data.gov.ru portal (in Russian), 16 July 2025: 5,826 datasets, about 100 MB in ZIP, age by first publication, the 2022 archive at 14 GB, EMISS and data.mos.ru

PART XI.

Trends

What is changing in the world of data catalogs.

1. AI, Data Spaces and Regulation
2. What Innovative Catalogs Can Already Do

AI, Data Spaces and Regulation

A catalog that already serves descriptions, files, licences and a full dump is ready for most of the novelties of recent years. An entry point for an assistant program, search by phrase, a record for model training and the exchange of closed data between organisations all sit on that layer. Where the layer is missing, the assistant runs into a captcha, and in a data space there is nothing to agree on.

AI-ready data is a dataset that a program can find, understand and take: an open format, an explicit licence, provenance and a dump. No separate platform is needed for this label.

What is added	Who it helps	What must already work
An `llms.txt` file and an MCP server	a program that searches for datasets in a conversation	a description dump, an API, a licence on the dataset, field definitions and units
Search by phrase	a person who does not know the exact dataset name	filled-in descriptions
A Croissant record	whoever trains a model, and dataset search	files, fields, licence, provenance
A government platform for AI	speech, text and image datasets prepared for training	this is a separate catalog type, with its own access conditions
A data space	organisations that exchange data with different rights	an open description, even when the file is closed
A chat in place of the catalog	nobody who needs to check a number or load it into their own program	the dataset itself, with an address and a licence

Programs as users

A program needs the same four things as a catalog harvester: a stable address, a complete record, an explicit licence, a dump without a login. Dataset search engines have crawled catalogs for a long time. They have been joined by assistants built on large language models: these search for a dataset, read the description and take the file through an API.

An `llms.txt` file is a short note about a site: what is published here and where to look. It is a proposed format, not an adopted standard. An MCP server gives a program a ready list of queries so that it does not guess addresses. Both read what the catalog already serves. Such an entry point serves an empty record faster than the site does.

The first notable entry point at a national portal is the data.gouv.fr experiment. The portal team announced it on 25 February 2026. The public address `https://mcp.data.gouv.fr/mcp` is read-only and needs no key. The server searches for datasets, reads records, for some tables runs a query through the portal's API, and parses an individual file. What enters the conversation is text and a link. A large file is still downloaded through the ordinary dump. The portal itself warns that there are servers around that call themselves official and are not. The code is open, under the MIT licence.

CKAN has no ready extension in its distribution. The Open Knowledge Foundation is running early research: a server in which an answer must link to the source dataset. The description says plainly that the project is at an early stage and the contract is still changing. Independent servers already

sit on the ordinary API of any CKAN portal: where the API is documented, the server is a thin layer on top, not a new database. The same approach is found outside open data: in Colectica Repository 8 an MCP server reads microdata descriptions in DDI. More in the chapter on microdata catalogs.

For the tables themselves, the protocol is narrow. It is designed for a short answer that fits into a conversation. A model does not read a binary file such as Parquet or GeoPackage through it. A good server therefore returns a sample, a query result and the file's address. The full table is taken through the usual bulk dump. How APIs and dumps work is in the chapter on access interfaces.

Programs as users have a downside too. Robots that collect texts and files for training create load, and portals begin to close. In 2026 the US Patent and Trademark Office put the bulk downloads and API of the data.uspto.gov portal behind registration. Many sites put up an "are you a robot" check, and government sites in a number of countries do not respond to requests from abroad, including from the clouds where the assistants themselves run. A person from the same country opens the catalog; a program does not. Load is handled with published limits and a full dump as a file. More in the chapter on anti-patterns.

An MCP server: your own, the platform's, someone else's

The MCP server is no longer a home-made gadget. Platforms provide it and statistical offices install it. The question "should we write our own" has been replaced by "should we switch on the ready one, and on what terms".

Platforms. Huwise, formerly Opendatasoft, has one server for all portals. It is not included in the plan by default. Each request carries the portal address and an API key; without a key the server responds with code 401. The agent has access to catalog search, including by the meaning of a phrase, a record with a schema and a sample of records, and queries against the records themselves. Permissions are the same as the user's, down to an individual column. In the PortalJS cloud each organisation has its own server address, and the server code is open. It searches for datasets and serves a record with a schema and the first rows of a table with a column summary. The full dataset does not enter the conversation. A server "by default" (with no plan, key or separate installation) is still rare among mass-market platforms.

Statistical offices. By Open Data Watch's count, by August 2026 official MCP servers had been launched by five national statistical offices and data agencies: those of the US, the UK, France, India and Singapore. The Statistical Institute of Catalonia became the sixth. The US Census Bureau server is installed in-house: it needs a Census API key, Docker and a local database with part of the descriptions. The Indian server for the eSankhyiki portal is in beta and covers seven statistical products.

Instructions for the agent. The most telling example is the World Bank's Data360 server. It gives the agent mandatory instructions, code lists and search rules. Before requesting data, a tool checks whether there is a series for the country in question and what years and breakdowns it has. The server has no database of its own: every value in an answer comes from the platform itself. Data Commons distributes agent instructions the same way, as resources of its server. This is a new layer of description. It is maintained and checked like a record.

What the pilots showed. In 2026 the Open Knowledge Foundation connected an MCP server to Brazil's transparency portal and Uruguay's open data catalog. The conclusion in the report of 10 June: connecting is easy, understanding meaning is hard. In one answer the model added "climate factors" to Uruguay's energy balance, which are not in the dataset. What helped was field definitions, units, period, known limitations and examples of right and wrong questions. Partners' frequent questions were turned into tests. This is the same work on description that a person needs. It is checked against the program's answers.

What someone else's server threatens. A server that goes to any portal at a supplied address can itself become a way into someone else's network. On 21 September 2026 vulnerability CVE-2026-61612 was published for the independent server ondata/ckan-mcp-server before version 0.4.108. The address check looked at the string with the server name and did not check where that name resolved. This made it possible to reach the cloud's internal address `169.254.169.254`, where its settings are kept. This was the third bypass of the same protection. It is safer to open your own portal with a server bound to one address, read-only, with updates from whoever maintains it.

What to switch on by default

If a platform provides the server, or you install it yourself, it should have six properties.

1. Read-only.
2. Open datasets without a key. A key is needed only for closed ones.
3. The server runs on top of the same public API with the same limits. It has no copy of the database of its own.
4. An answer links to the record and the file. A large file is served as a link to the dump, not as text in the conversation.
5. Alongside it are instructions for the agent: field definitions, units, period, known limitations.
6. Ten frequent questions with verified answers are run after every update to the server and the data. How to fit this into a self-audit is in the chapter on self-audit.

How to check an MCP server

The server responds to an ordinary POST request. The expected answers were obtained on 15 October 2026.

```
# Server name and version, without a key
curl -s -X POST https://mcp.data.gouv.fr/mcp \
  -H 'content-type: application/json' \
  -H 'accept: application/json, text/event-stream' \
  -d '{"jsonrpc":"2.0","id":1,"method":"initialize","params":{"protocolVersion":"2025-06-
18","capabilities":{},"clientInfo":{"name":"check","version":"1"}}}'

# List of tools
curl -s -X POST https://mcp.data.gouv.fr/mcp \
  -H 'content-type: application/json' \
  -H 'accept: application/json, text/event-stream' \
  -H 'mcp-protocol-version: 2025-06-18' \
  -d '{"jsonrpc":"2.0","id":2,"method":"tools/list"}'

# Data360 keeps its agent instructions as resources
```

```
curl -s -X POST https://maimcpext.worldbank.org/ext/data360/mcp \
  -H 'content-type: application/json' \
  -H 'accept: application/json, text/event-stream' \
  -H 'mcp-protocol-version: 2025-06-18' \
  -d '{"jsonrpc":"2.0","id":3,"method":"resources/list"}'
```

data.gouv.fr answered `"name":"data.gouv.fr MCP server","version":"1.28.1"` with ten tools, from `search_datasets` to `query_resource_data` and `get_metrics`. For Data360 the same requests give `"Data360 MCP Server"` version 0.1.0, 16 tools and the resources `data360://system-prompt`, `data360://codelists`, `data360://search-usage` and others. The Huwise server `https://mcp.huwise.com/mcp` answered the first request without a key with code 401.

Next, ask the server three frequent questions about your own data. The answer must name the dataset and give a link to the record. The number in the answer is checked against the file.

When to set up a separate entry point

Context. A regional open data portal on CKAN. The national portal harvests its descriptions. Management asks to "make the catalog AI-ready": an MCP server, search by phrase and, while at it, a "data space". The description dump is incomplete, and some datasets have no licence. One administrator maintains it.

Options. Fix the licences and the dump. Install MCP on top of the current API. Replace the search box with a chat. Join someone else's data space or set up your own.

Criteria. The national portal keeps harvesting descriptions. A program downloads the file without registration. The licence is visible in the record and in the dump. There is someone to document the new entry point. Exchange of closed data between organisations is either already among the portal's tasks or it is not.

Decision. Licences and the dump first. MCP is installed after that, if assistants are already in use and there is someone to maintain the server. If the platform already provides a server, you do not write your own: the ready one is switched on read-only and without a key for open datasets. Search by phrase can be added next to the ordinary search box. No data space is set up.

Why. MCP and search by phrase read what the API already serves. They serve an empty record faster than the site does. A chat in place of the catalog leaves no number that can be cited and loaded into one's own program. A data space is needed when many organisations exchange data and their access differs. A regional portal of open CSVs does not have this task, and a connector of its own with one administrator would be left unmaintained.

Consequences. An assistant connected later will see the same descriptions as the national portal. While licences are empty, a dataset still cannot safely be put into a training set: it is unclear whether that is allowed.

When to review. A steady program user has appeared, and a person who will update the MCP server. The platform has started providing a server itself: then the question is not "should we write our own" but on what terms to switch on the ready one. Or several organisations ask to be given closed data under contract, and there are more requests than email can handle.

How to write such a record for yourself is in the chapter "Justifying a Decision".

AI on the catalog side

A model speeds up the draft of a description. A person still publishes.

The most expensive part of a catalog is the text of its records. In May 2025 the dataportals-registry catalog registry was enriched with the help of AI agents: given a site address, an agent proposed a title, description, topics and geographic coverage. Manual work of the same depth was estimated at six months for two people. People checked the result. In the same way a model can suggest which column is a date and which a region code, and sketch a data dictionary. The draft is checked against the file: a model confidently invents units of measurement too.

Search by phrase finds a dataset, not a value in a table. On the Norwegian portal data.norge.no such search sits next to the ordinary search box and SPARQL queries, not in place of them. It was written about in April 2025. A person asks in ordinary words, and the model picks links to datasets based on their descriptions. The portal itself warns that the programmatic interface for this search is still changing and covers only datasets. People, however, often search for a value: how many, where, when. Such a value is still taken from the file or an indicator API.

Berlin's Geoexplorer is a finished prototype from July 2024 on top of the city geoportal's metadata. It searches for layers by phrase and writes an explanation for a dataset from the results. The code is open under the MIT licence, and the site was still responding in 2026. It is a search showcase, not a replacement for the geoportal and not proof that layer descriptions no longer need to be maintained.

There is also a consequence for data owners. The argument "a machine-readable format is expensive" is weakening: if a table is published in any form at all, a parser can be written quickly. In November 2025 the standard API of Armenia's statistical database did not respond, and more than 900 indicators were collected from its pages. By October 2026 it was already running PxWeb release 2025, and the API worked. Details are in the chapter on indicator catalogs. This is no reason to publish scans. What is collected from pages breaks at the first layout change, and nobody checks parsing errors.

What counts as AI-ready

In December 2025 the Open Data Institute released the second version of its "AI-ready data" framework. It does not introduce a new catalog format. Its four groups are almost entirely covered in other chapters.

Group in the ODI framework	What to do in the catalog	Where it is covered
The dataset itself	open format, clear codes, documented limitations	formats, good practices
Description	a machine-readable record, licence, provenance	standards, licences
Infrastructure	a portal or page, API, versions	access interfaces, identifiers
Governance	a named owner, a contact, a clear access procedure	access and workflows

In November 2024 the same institute checked whether UK open data gets into general-purpose language models. Pages of data.gov.uk were found in the April 2024 Common Crawl dump: 13,556 addresses. These were records with a download link, not the tables themselves. In a recall test the models correctly gave the official figure in 5 cases out of 195, and all five related to one dataset. The models remembered the texts of ordinary government sites better. The conclusion for a catalog is direct. A record with a link makes a dataset findable for a person. A number from that record does not get into a general model's memory by itself. The current value is still served as a file or through an API, not as the answer of a model trained last year.

Government data catalogs for AI

Datasets for training models still live mainly on Hugging Face and Kaggle. Governments are building their own platforms alongside: there the language is different, there are sectoral records and access is stricter. Your own open data portal is not turned into an ML catalog because of this. How the type works is in the chapter on ML catalogs.

- **India, AIKosh.** According to an IndiaAI announcement of 20 August 2026, the platform had more than 15,000 datasets, 300 models and 30 tools. On 26 August 2026, downloads on the records opened after registration, datasets carried a quality rating, and the most common licence in the terms was CC BY-NC-SA: commercial use without separate permission is closed. It is a national repository for training, with its own rules on each dataset.
- **Korea, AI Hub.** The NIA agency's platform publishes training datasets: speech, text, images, including in Korean. Under the platform's rules the description is open without a login, while downloading asks you to confirm your identity and state a purpose. The data is intended for training models. An organisation abroad, and taking a dataset out of the country, require a separate agreement. Selling the dataset itself is also a separate contract, although the rules in principle allow research and development, including commercial.
- **US, NAIRR.** The pilot of the national AI research resource gives researchers access to compute, models and some datasets. It is not a file showcase for everyone. On 1 September 2026 the National Science Foundation announced an operations centre that is to turn the pilot into a permanent service. On that date the pilot site was still accepting applications. More than 800 research projects have gone through the pilot since 2024.
- **EU, AI-on-Demand.** A catalog of descriptions of what has already been published on other platforms, including Zenodo and Hugging Face. On 14 October 2026 the counter on the home page showed 118,025 dataset descriptions. That is the size of an index, not an archive run by the European Commission.

The common requirement for such a platform is one. The record names the training licence, the source and known limitations. The file may be closed behind registration. The description, however, stays open; otherwise the dataset cannot be found and one cannot decide whether to apply.

Data products

An assistant program needs a dataset with a known owner, a schema, example queries and a stable interface. In discussions of catalogs this is called a data product: the dataset is maintained, not uploaded once. A typical open data portal falls short of such a product, so a "semantic layer for AI" on it turns out thin. How a product differs from a dataset is in the chapter on units of publication. A practical list is in the chapter on good practices.

A model instead of data

A government may decide that opening tables is no longer necessary: a chat trained on them will give the answer. A general-purpose model cannot do this. In the UK check of 2024 it hardly remembered data.gov.uk figures, although the portal's pages did get into the training dump. A model's answer cannot be checked against a file, cited, compared with last year or loaded into one's own program. If open data is closed and a chat is left in its place, that is closing the data.

A chat is appropriate next to a catalog: it suggests a record and links to it. A chat does not replace the number, the licence or the file.

Descriptions for model training

A dataset on which models are compared is described with a Croissant record: contents, fields, the split into parts, licence, known limitations. Version 1.1 came out on 29 January 2026. A government open data portal does not have to become Hugging Face because of this. It does have to state the licence and source in a way that prevents the dataset from being quietly moved into a training set. How Croissant works and who already serves it is in the chapters on standards and on ML catalogs.

Regulation (EU) 2024/1689 on artificial intelligence places this work on the model provider, not on a city portal. Since 2 August 2025 a provider of a general-purpose model publishes a sufficiently detailed summary of what the model was trained on. For models already on the market by that date, the deadline is 2 August 2027. A portal helps by having the licence and source in the record and in the dump. The summary is written by whoever trained the model.

Data spaces

European data spaces negotiate access to data between organisations, including data that is not for everyone. This is closer to a catalog with access levels and to a marketplace than to a "download the CSV" portal.

There is one thing in common with an open data portal. The description can be read from outside; otherwise there is nothing to make an agreement about. Ordinary DCAT is enough for this. The file may stay closed. Conditions ("allowed for research, may not be passed on") are written in the ODRL language. Negotiating the agreement and the transfer itself go through the Dataspace Protocol: the Eclipse Foundation approved version 2025-1 on 23 July 2025. How the platforms and the connector work, and examples such as Catena-X, are in the chapter on other catalog types.

Gaia-X in this construction is a trust framework: who is a participant and what requirements they meet. It is not a catalog and not a substitute for a dataset description.

The Data Spaces Support Centre Blueprint version 3.0 (February 2026) sets out what a space needs besides a showcase: a governance scheme, agreements, shared standards. Since 2026 the maturity of such an initiative is measured against the European technical specification CEN/CLC/TS 18331:2026, not against a slogan on a website. The European Commission runs sectoral spaces: health, mobility, industry and others. A city portal does not add itself to this list.

Situation	What to do
A city publishes open files	a tidy portal, or publishing in the regional one
The description can be shown to everyone, the file is issued on request	a catalog with access levels, as for microdata
Several organisations exchange data, and their rights differ	a data space or a sectoral platform with agreements
Data is sold	a marketplace; the description is still open before payment
A chat must answer residents	keep the dataset, licence and dump; the chat links to them

Not every city needs a data space. Most still need a portal in which the description, licence and file are in place.

Rules

Recent laws name the result of publication. They rarely name the platform.

Document	What changes for a catalog
Directive (EU) 2019/1024 on open data	public sector documents are made available in machine-readable form and, where possible, free of charge; countries transpose this into their own laws
Regulation (EU) 2023/138 on high-value datasets, applies from 9 June 2024	six categories (geospatial, earth observation and environment, meteorological, statistics, companies, mobility) free of charge, via API and as bulk download
Regulation (EU) 2022/868, the Data Governance Act, applies from 24 September 2023	an inventory of protected public sector data: the description is open, the file is issued under rules, including in a secure environment
Regulation (EU) 2023/2854, the Data Act, applies from 12 September 2025	access to data from connected devices and exchange between companies; usually adds no new CSVs to a city open data portal
Regulation (EU) 2024/1689 on artificial intelligence	from 2 August 2025 a general-purpose model provider publishes a summary of training data (for models already released, by 2 August 2027); the portal does not write this summary

The six high-value categories and the inventory of protected data are covered in the glossary and in the chapter on access. The European Health Data Space Regulation requires catalogs from health data access bodies from 26 March 2029: this too is a catalog with an open description and a closed file, not a portal of open CSVs.

In the US, OMB memorandum M-25-05 of 15 January 2025 is guidance under the OPEN Government Data Act. Among the purposes of open access it names training artificial intelligence systems, and it requires agencies to inventory their data and publish through the federal catalog. A Congressional Research Service overview treats the memorandum as in force; the same service's report on data.gov of 21 May 2026 discusses it as current guidance.

If a law names a platform, the choice from this book narrows to how to switch on the dump, licences and persistent addresses on that platform. If a law names a result (machine readability, a licence, no charge, an API), the platform is chosen as usual.

What has not changed

A persistent address, a licence on the dataset, a dump of all descriptions and a clear file format are needed by a person, a search engine and an agent alike. A catalog that already does this survives changes of fashion without being rewritten. A showcase catalog does not: neither the federation nor a program sees it.

Before adding a new entry point, check six things.

1. A program takes all descriptions without a login and without a captcha.
2. The licence is on the dataset and gets into the dump.
3. The file downloads via a persistent link, rather than only opening in a chat.
4. Search by phrase, if switched on, returns records. The ordinary search box remains.
5. A chat answer, if there is one, links to the dataset. The dataset can still be downloaded.
6. An MCP server, if switched on, is read-only, and answers frequent questions with verified numbers and a link to the dataset.

Which new catalog capabilities are worth ordering, and which are still only someone else's service, is in the chapter on innovative catalogs.

Sources

- Croissant 1.1, 29 January 2026
- data.gouv.fr, announcement of the MCP experiment (in French), 25 February 2026; guide (in French); code datagouv/datagouv-mcp, MIT licence
- Open Knowledge Foundation, mcp-server: early research, the response contract is still changing; an independent server for the CKAN API: ondata/ckan-mcp-server. Checked 14 October 2026
- Open Knowledge Foundation, MCP for Open Data Portals: Trust Depends on Understanding the Data, 10 June 2026: pilots in Brazil and Uruguay, frequent questions as tests
- NVD, CVE-2026-61612, 21 September 2026: ondata/ckan-mcp-server before 0.4.108, an address check without name resolution, the third bypass of the protection
- Huwise, MCP server: not included in the plan automatically, an API key and portal address in every request; PortalJS, MCP server, 25 November 2025, code datopian/portaljs-mcp-server
- Open Data Watch, Making Official Statistics Work in an AI-Driven World, September 2026: five official MCP servers of national offices by August 2026
- World Bank, Data360 MCP: agent instructions, code lists, a coverage check before requesting data; Data Commons, working with an AI agent
- U.S. Census Bureau, Data API MCP, CC0 licence; PIB, the Indian statistical office's MCP server, 6 February 2026

- Open Data Institute, The UK government as a data provider for AI, November 2024: 5 correct answers out of 195 and 13,556 data.gov.uk pages in the April 2024 Common Crawl; A framework for AI-ready data, version 2, December 2025
- data.norge.no: AI search and API description; data.europa.eu news item, 25 April 2025
- Geoexplorer Berlin, project page (in German; July 2024, status "completed"), code under the MIT licence
- IndiaAI / PIB, AIKosh workshop, 20 August 2026: more than 15,000 datasets, 300 models, 30 tools; I. Begtin, a look at AIKosh records, Telegram (in Russian), 26 August 2026
- AI Hub usage policy (in Korean): identity confirmation, a stated download purpose, a separate agreement for use from abroad
- NAIRR Pilot; NSF, operations centre, 1 September 2026: more than 800 projects since 2024
- AI-on-Demand: on 14 October 2026 the counter showed 118,025 dataset descriptions; Zenodo and Hugging Face appear alongside in the catalog results
- Data Spaces Support Centre, Blueprint 3.0, February 2026; summary; CEN/CLC/TS 18331:2026: a maturity assessment specification, approved in April 2026; European Commission, common European data spaces
- Regulation (EU) 2024/1689 on artificial intelligence: a training data summary from general-purpose model providers from 2 August 2025
- OMB, memorandum M-25-05, 15 January 2025; Congressional Research Service, IN12502 and R48954, 21 May 2026
- I. Begtin, AI and open data. Opinions and doubts (in Russian), March 2026: MCP at portals and the risk of replacing data with a model
- I. Begtin, Telegram (in Russian): AI agents in the catalog registry, 14 May 2025, and the result of the enrichment, 22 May 2025; cheaper data collection, 5 January 2026; geoblocking and AI agents, 14 July 2025; data.uspto.gov, 23 June 2026; MCP for data and binary formats, 3 July 2025
- Directive (EU) 2019/1024, Regulation (EU) 2023/138, Regulation (EU) 2022/868, Regulation (EU) 2023/2854
- Discoverability, marketplaces and API catalogs

What Innovative Catalogs Can Already Do

Innovative catalogs differ not in platform but in shortening the path from a record to an answer. Some turn a file into an API by themselves. Others issue a DOI for a query, keep table versions like code, check a dataset before publication or compute on their side. Almost none of this is yet in the core of CKAN, ArcGIS Hub, Huwise, Socrata, Dataverse, DSpace or GeoNetwork. Some can be installed next to your portal: the code is open. Some are for now available only as someone else's service. That determines what from this chapter to put in a technical specification.

Below are nine such capabilities. For each there are one or two catalogs where it is most visible, and answers to three questions: how to replicate it, when it is not needed, and how to check it with a request.

The catalogs were selected by four criteria:

- the capability is not in the core of mass-market platforms: something switched on in settings is no longer an innovation;
- it works publicly and can be checked with a request without a login;
- the idea can be replicated: there is open code, a standard or at least a documented convention;
- its status is clear. Beta, a single operator or a promise to keep for five years are part of the description, not a footnote.

Capability	Where it is most visible	How to replicate it	When it is not needed
An API for every tabular file	data.gouv.fr, Hugging Face, Datasette	the open api-tabular service, Datasette; in CKAN, DataStore with a loader	there are few files, they are small and are downloaded in full
Identifiers for objects inside the data	Overture Maps, OpenSanctions	identifier rules in a register, crosswalk tables	the dataset is not linked to anything
A DOI for a query	GBIF; more simply, Our World in Data, CELLxGENE Census	a DataCite DOI for a saved query; an address convention	the data is not cited
Table versions as in Git	DoltHub, OpenNeuro	Dolt and DoltLab, DataLad	one person edits the table once a year
Standard-based checks on entry	OpenNeuro, DANDI	a schema and a validator on upload	the field has no shared standard
Processing next to the data	Copernicus Data Space Ecosystem, Climate Data Store, ERDDAP	ERDDAP, an openEO server	the data fits in an ordinary file
Combining many sources	Data Commons, HDX HAPI, OpenSanctions	Data Commons open code, yente	a single organisation's catalog
Tiered access	microdata.no, Redivis	in-house development or someone else's service	the data is fully open
Readiness for AI agents	data.gouv.fr, the World Bank's Data360	an MCP server provided by the platform	there are no licences and no full dump

Where the series begins: Source Cooperative

What is new in Source Cooperative is not the storage but the fact that the catalog hardly stands between the reader and the data. A product is a folder in object storage with a README description. Files in cloud-native formats are read in parts directly over the network. In the

dataportals-registry, the platform has a STAC interface recorded: a protocol for describing spatial files. Its strengths and weaknesses (beta, publishing by application, no DOIs or versions) are covered in the chapter on cloud storage and hubs. The next two sections develop the same idea. In Overture Maps, objects inside the data have persistent identifiers. On Hugging Face a query can be run against a file straight away.

An API for every tabular file

Usually, between "download the file" and "get an API" lies the publisher's work: load the table into a database, describe the columns, maintain the loader. Innovative catalogs do this themselves.

data.gouv.fr. The French portal's tabular API (still in beta) picks up every tabular file by itself within a few minutes of publication. The size limit is 100 MB for CSV, 50 MB for XLS and Parquet, 12.5 MB for XLSX. Each file has three addresses. The profile shows column types and recognised formats (for example, an INSEE commune code or a latitude), frequent values and the number of missing values. The data is served page by page, with filters and sorting. A Swagger description of the queries is built for each file separately. The limit is 100 requests per second. The service page lists 31,768 datasets linked to it. The service code is open under the MIT licence and can be installed next to your own portal.

Hugging Face. The platform itself converts datasets in supported formats to Parquet and builds a viewer, search, filters and statistics on them. The SQL console runs entirely in the browser on DuckDB. A query can be shared as a link, and the result exported to Parquet or CSV. This shifts the boundary the book draws in the section on preview: a preview cannot be cited, while a link to a query is already reproducible. It still does not become a persistent link. The dataset may be updated, and the same query will return different rows.

Datasette is the same principle on a small scale. Every page has a JSON version, filters are written in the address, and there is full-text search and arbitrary SQL. The registry has 45 entries on Datasette, 44 of them active. How to publish a small dataset on it is in the chapter on static sites.

How to replicate it. In CKAN the same result comes from DataStore and the XLoader extension, but the row limit and the loader remain your responsibility. More in the chapters on access interfaces and on storage.

When it is not needed. If there are a dozen files and they are downloaded in full, an API adds nothing. Limitations must be stated honestly. Large files remain just files. A beta may change addresses. An API over someone else's CSV inherits all of that CSV's errors: wrong encoding, extra rows above the header, text in a numeric column.

Identifiers for objects inside the data

The book covers a dataset DOI in detail. But an identifier can also belong to an individual row. This changes how data is linked together.

Overture Maps. Every building, road, place and administrative unit in Overture's monthly release has a persistent identifier in the GERS system. Alongside are kept an identifier register with dates of first and last appearance, a change log for each object and crosswalk tables with the identifiers

of the original sources. Overture explains why this is needed. A traffic data provider can publish only the traffic, without the roads themselves, and an indicator provider only the indicators, without boundaries. They are joined by identifier, without matching geometries.

There is an honest lesson here too. In release 2025-06-25.0 Overture replaced its old identifiers, which encoded an H3 grid cell and a theme, with random UUIDs. The old identifiers can no longer be found. An identifier with meaning built into it costs more than it seems: the meaning changes, while links to the old number remain.

OpenSanctions shows the same idea for people and companies. An entity is assembled from many lists, and after deduplication it has one canonical identifier. If it matched a Wikidata entry, the Wikidata number becomes the identifier.

How to replicate it. Register rules: an identifier by rules, not a row number; records are not deleted but given a status; a crosswalk table with old numbers. They are already listed in the chapter on identifiers.

When it is not needed. A dataset nobody links to others (an annual report, a one-off survey) can make do with a dataset identifier.

A DOI for a query

A DOI is usually issued for a dataset. Innovative catalogs issue it for a query.

GBIF. Every occurrence download has a DOI. The download page keeps the list of source datasets and all the selection conditions. On 15 October 2026 the iNaturalist observations dataset was part of 2,623,241 downloads, each with its own DOI. If a researcher then filtered the data, combined it from several downloads or computed in the cloud, they register a derived dataset with a list of sources and also get a DOI. GBIF's monthly snapshots in the Microsoft, Amazon and Google clouds have DOIs. GBIF tracks where these DOIs are cited. Citing a download is covered in the chapter on citation. What is new here is not that a DOI exists, but that the query becomes the object of publication.

Alongside are two simpler solutions.

Our World in Data lets you append `.csv`, `.metadata.json` or `.zip` to the address of any chart and get the data, a description and a README. The description has a ready source line for citation. This is an address convention; it can be adopted without any software.

CELLxGENE Census publishes single-cell sequencing data as one object from which slices are taken by cell and gene. Long-term support releases are promised every six months, each kept for at least five years. There is a link generator for any slice. A promise to keep a release for a named period is a good model for the chapter on archiving. The status, however, is checked: on 15 October 2026 the release file named 2025-11-08 as the stable release and 2025-11-17 as the latest.

How to replicate it. A DOI for a saved query is issued through DataCite in the same way as for a dataset. What is needed is the saved query text, a date and a list of sources. The general scheme is described in the Research Data Alliance recommendations on citing dynamic data.

When it is not needed. If the data is not cited in published work, an access date and an archive of past dumps are enough.

Table versions as in Git

Dolt is an SQL database compatible with MySQL. It can be cloned, branched, merged and pushed just like a Git repository. **DoltHub** hosts public databases for free. A data change goes through a pull request with review: the reviewer sees the difference by rows and cells, not by file. DoltLab is the same platform for self-hosting. It is an option between a Git repository and a portal: edit history as in Git, but for a table that no longer fits in a 2 MB CSV. More in the chapter on Git repositories.

OpenNeuro is the same idea in a research archive. Every dataset snapshot is a version with its own Git hash and DOI. A dataset can be taken from S3, via DataLad or directly from Git. DataLad is a program that keeps large datasets as Git repositories: history and description in Git, the files themselves separately. The registry has 10 installations on DataLad.

When it is not needed and what it risks. A table edited once a year by one person is easier to keep in ordinary Git. A platform with a single operator may close: in 2022 the QRI catalog, built on the same idea, closed this way; the story is in the chapter on cloud storage and hubs. So the decision record needs an answer to how to retrieve the database and its edit history.

Standard-based checks on entry

The book's part on quality measures it after publication. Neuroscience data archives check it before publication.

OpenNeuro accepts only datasets following the BIDS standard, the shared order of folders, file names and descriptions for neuroimaging. On upload a validator checks the dataset. Upload is possible with warnings, but not with errors. The terms are strict: images are anonymised, and the licence is CC0 immediately or 36 months after the first snapshot.

DANDI is a neurophysiology archive on the NWB standard. The publish button becomes active only after all errors in the check of the dataset description and files have been fixed. A published version does not change and is cited by number. A limitation is stated honestly: datasets with Zarr files cannot yet be published. The registry has three DANDI installations.

Here quality is a condition of entry, not an indicator on a dashboard. It works where a field has a shared standard. For a city portal the analogue is checking a table on upload against a schema from the schema.data.gouv.fr catalog. How to fit this into quality levels is in the chapter on quality levels.

When it is not needed. If there is no shared standard, a check on entry turns into one portal's list of rules. Publishers get around it with empty fields.

Processing next to the data

Here the catalog serves not a file but the needed piece of data or the result of a calculation.

Copernicus Data Space Ecosystem offers S3, STAC, openEO and Sentinel Hub for the same satellite data. openEO is an open protocol by which a user sends the server a chain of operations on imagery: cut out an area, calculate a vegetation index, average by month. The server computes, the user receives the result. There is a federation with partner platforms and a separate access point without it. The limits of the free tier are stated plainly: two synchronous requests and two batch jobs at a time, tested on areas up to 100×100 km at 10 m resolution. The registry has 12 entries with openEO in the address, three of them for Copernicus Data Space Ecosystem.

Climate Data Store takes a simpler approach, and one aimed at people. The user assembles a selection in a form on the dataset page, and a "Show API request code" button produces ready Python request code. The terms of use are accepted manually on the dataset page, once.

ERDDAP is a reminder that an innovation need not be new. The whole request, including a selection by space and time in ordinary units, fits in the address. The response format is chosen on the fly: CSV, JSON, NetCDF, MATLAB, KML, PNG. There is a subscription to dataset changes. The registry has 175 entries on ERDDAP, 174 of them active. That is more than any other solution in this chapter.

When it is not needed and what it risks. If the data fits in an ordinary file, server-side computation is not needed. Computation is the most expensive part and is the first to be closed: in 2024 Microsoft closed the Planetary Computer Hub, while the catalog remained. So the catalog and computation are described in requirements as separate items. Losing the second must not mean losing the first.

Combining many sources

An ordinary catalog knows files. These catalogs know variables, places and entities, and answer a question across all sources at once.

Data Commons gathers statistics from many organisations into a knowledge graph: one database where every variable and every place has its own identifier and the links between them are recorded explicitly. Data is served through REST, SDMX 3.0, BigQuery, embeddable components and an MCP server. "Your own Data Commons" can be deployed from the open code.

HDX HAPI is especially useful to a reader with CKAN. The humanitarian portal HDX runs on CKAN, and the CKAN API serves only descriptions. HAPI adds on top indicator values from many sources, brought into a single form: population, food security, funding. They are updated daily; the numbers themselves are not altered. It is a layer on top of an existing portal without changing the platform.

OpenSanctions stores each value as a separate statement: from which source, which property, when it was first and last seen, and with which canonical identifier the entity was merged after deduplication. The statements table is exported every night. This is provenance at the level of a value, not a dataset. The yente search and matching service is open under MIT. The data itself is free only for non-commercial use.

Alongside are DBnomics, already described in the book (one API to many statistics providers, with the original codes preserved), and IPUMS (an extract builder and a single API for all microdata collections). More in the chapters on indicator catalogs and on microdata catalogs. In **Aleph** a user

uploads their own list of people or companies, and the system matches it against hundreds of datasets they have access to. A worrying sign: the last commit to Aleph's main repository was on 20 February 2026.

When it is not needed. A single organisation's catalog has nothing to combine. Combining is needed by an aggregator and by whoever answers questions across many sources at once.

Tiered access

microdata.no is the strongest example. Staff and students of approved universities, research organisations and ministries work with Norwegian register data without an application and without a project described in advance. They do not see records. The data does not leave secure storage, and results are anonymised on the fly.

Redivis divides access to a dataset into five tiers: the dataset is invisible, the description is visible, the variables and summary statistics are visible, a sample is available, all data is available. Each tier can be opened to everyone, to specific people or after conditions are met. The registry has 13 entries on Redivis and 27 catalogs with a GA4GH DRS interface, a standard for serving biomedical data files according to access rights.

Tiered access is a catalog function. The description and statistics are open to everyone, the data is issued by tier. For a reader with sensitive data this gives an option between "open everything" and "show nothing". How access levels work on mass-market platforms is in the chapters on access and roles and on microdata catalogs.

When it is not needed. If the data is fully open, tiers only hide the description behind an extra step.

Readiness for AI agents

An AI agent is a program built on a large language model that itself searches for a dataset, reads the description and takes the data. An MCP server gives such an agent a ready list of queries to the catalog. This is covered in detail in the chapter on AI and data spaces. Here is what already counts as cutting-edge in this area.

Platforms themselves now provide the server, but almost nobody yet switches it on by default. Huwise has a server, but it is not included in the plan, and every request needs an API key. Each organisation in the PortalJS cloud has its own server address, and its code is open. data.gouv.fr runs its own server read-only and without a key. The best example of design is the World Bank's Data360 server. Besides tools, it gives the agent mandatory instructions, code lists and a check of whether a series exists for the country in question, before data is requested. Data Commons distributes agent instructions the same way, as resources of its server. This is a new layer of description: instructions on the data for a program, which also need to be maintained.

When it is not needed. While datasets have no licences and no full dump, the server just serves empty records faster. You install it yourself only after that. If the platform provides the server, it is switched on read-only, without a key for open datasets and on top of the same public API.

What not to single out as an innovation

A chat in place of the catalog and search by phrase are already covered in the chapter on AI, along with the "model instead of data" risk. data.europa.eu's machine translation of records into the 24 EU languages is in the chapter on other capabilities. The open code of the Malaysian statistics showcase OpenDOSM and data.gov.my is in the chapter on indicator catalogs. Small static "data as code" portals on Datadex are represented in the registry by three entries, none of which has an API recorded. This is still an experiment.

How to check

The commands require no login. The expected answers were obtained on 15 October 2026.

```
# data.gouv.fr: profile, rows and Swagger description for one CSV.
# RID is the file identifier from the dataset record
RID=571a3021-0e10-471c-98b8-72bcab5daca8
curl -s "https://tabular-api.data.gouv.fr/api/resources/$RID/profile/"
curl -s "https://tabular-api.data.gouv.fr/api/resources/$RID/data/?
page_size=2&commune__exact=Cessy"
curl -s -o /dev/null -w "%{http_code}\n" "https://tabular-
api.data.gouv.fr/api/resources/$RID/swagger/"

# Hugging Face: ready Parquet files and what is enabled for the dataset
curl -s "https://datasets-server.huggingface.co/parquet?dataset=nyu-mll/glue"
curl -s "https://datasets-server.huggingface.co/is-valid?dataset=nyu-mll/glue"

# Overture Maps: release catalog in STAC
curl -s "https://stac.overturemaps.org/catalog.json"

# GBIF: how many DOI-bearing downloads include the iNaturalist dataset
curl -s "https://api.gbif.org/v1/occurrence/download/dataset/50c9509d-22c7-4a22-a47d-
8c48425ef4a7?limit=1"

# Our World in Data: a chart's data and description via the address convention
curl -s -o /dev/null -w "%{http_code} %{content_type}\n"
"https://ourworldindata.org/grapher/life-expectancy.csv"
curl -s "https://ourworldindata.org/grapher/life-expectancy.metadata.json"

# CELLxGENE Census: releases and the long-term support flag
curl -s "https://census.cellxgene.cziscience.com/cellxgene-census/v1/release.json"

# DoltHub: SQL against a database's edit history
curl -s "https://www.dolthub.com/api/v1alpha1/dolthub/ip-to-country/master?
q=SELECT%20commit_hash%2C%20committer%2C%20date%20FROM%20dolt_log%20LIMIT%202"

# OpenNeuro: a dataset's latest snapshot
curl -s https://openneuro.org/crn/graphql -H 'content-type: application/json' \
  -d '{"query":"{ dataset(id:\"ds000001\"){ latestSnapshot { tag created } } }"}'

# DANDI: dataset versions and the draft validation result
curl -s "https://api.dandiarchive.org/api/dandisets/000003/versions/?page_size=3"

# Copernicus Data Space Ecosystem: openEO server
curl -s "https://openeo.dataspace.copernicus.eu/openeo/1.2/"

# HDX HAPI: an application key is issued immediately, without registration
curl -s "https://hapi.humdata.org/api/v2/encode_app_identifier?
application=book&email=book%40example.org"
```

```
# OpenSanctions: when the main dataset was updated
curl -s "https://data.opensanctions.org/datasets/latest/index.json"
```

What the answers contained:

- data.gouv.fr: the profile recognised latitude and longitude in the `lat` and `lon` columns (`latitude_wgs`, `longitude_wgs`), the filter returned the row for the commune of Cessy, and the Swagger description answered with code 200. The file is taken from a dataset on the availability of hearing-aid specialists by commune. If the dataset is withdrawn, substitute the identifier of any CSV up to 100 MB;
- Hugging Face: Parquet file addresses in the `refs/convert/parquet` branch; the dataset has the viewer, search, filters and statistics enabled;
- Overture: a release catalog on STAC 1.1.0, with releases including 2026-08-19.0 and 2026-09-23.0;
- GBIF: `count` is 2,623,241, and the first download in the response has its own DOI, licence and selection conditions;
- Our World in Data: a CSV file with code 200, and a source line in the description: "Riley (2005); Zijdeman et al. (2015); HMD (2025); UN WPP (2024)";
- CELLxGENE Census: `"stable": "2025-11-08"`, `"latest": "2025-11-17"`, with long-term support releases flagged `"lts": true` and `"do_not_delete": true`;
- DoltHub: `"query_execution_status":"Success"`. This database's branch is called `master`; an address with `main` returns "branch not found";
- OpenNeuro: `ds000001` has snapshot `1.0.0` of 14 May 2020;
- DANDI: `count` is 5, and the first version in the list is the `draft` with status `Valid`;
- openEO: `api_version` is 1.2.0;
- HDX HAPI: an `encoded_app_identifier` string, which is passed in the `app_identifier` parameter of subsequent requests;
- OpenSanctions: the `default` dataset has the update date `2026-10-15T12:54:35`.

How to check an MCP server is in the chapter on AI. The registry figures were obtained with queries from the appendix "Using the Registry".

What follows

The nine capabilities are not a list of what is mandatory. They are a list of what can be asked for if there is a task. For it to work as a selection tool, each capability in a technical specification should be marked "mandatory", "desirable" or "not needed", with a sentence on where the need comes from. Ready wordings are in the section "New capabilities" of the requirements checklist.

Three things from this chapter suit almost any catalog:

- an API for tabular files, if the files are small and people really filter them;
- a schema check on upload, if a schema already exists;
- a read-only MCP server, if the platform provides it.

The rest depends on the task: a DOI for a query for those whose data is cited; access tiers for an owner of sensitive data; server-side processing for those whose data does not fit in a file.

Sources

- dataportals-registry, release 1.27.0 of 15 October 2026: ERDDAP 175 entries (174 active), Datasette 45 (44), Redivis 13, DataLad 10, DANDI 3, Datadex 3 (none with an API recorded), 27 catalogs with GA4GH DRS, 12 entries with openEO in the address
- data.gouv.fr, API tabulaire (beta) (in French): size limits, profile, 100 requests per second, 31,768 datasets; code datagouv/api-tabular, MIT licence
- Hugging Face, Dataset viewer and SQL Console
- Datasette, JSON API
- Overture Maps, GERS: identifier register, change log, crosswalk tables, the identifier change in release 2025-06-25.0
- GBIF, citation guidelines: download DOIs, derived datasets, cloud snapshots
- Our World in Data, Chart API
- CZ CELLxGENE, Census: long-term support releases, link generator
- Research Data Alliance, Precise, Actionable Reference to Dynamic Data, 2025: citing dynamic data
- DoltHub, What is Dolt?, SQL API
- OpenNeuro, user guide: BIDS, snapshots, CC0 after 36 months; BIDS
- DANDI, publishing Dandisets: validation before publication, an immutable version, the Zarr limitation
- Copernicus Data Space Ecosystem, openEO: federation and free-tier limits
- Climate Data Store, API setup
- ERDDAP, information
- Data Commons, working with an AI agent
- HDX, Humanitarian API
- OpenSanctions, statement model; yente, MIT licence
- DBnomics; IPUMS, API Program
- Aleph, cross-referencing; repository: last commit on 20 February 2026
- microdata.no; Redivis, access levels
- Huwise, MCP server; PortalJS, MCP server; World Bank, Data360 MCP
- Cloud storage and hubs, AI and data spaces, requirements checklist

Appendices

Glossary, using the registry, comparison tables, checklists and bibliography.

1. Glossary
2. Using the Registry
3. Platform Comparison Tables
4. Checklists
5. Bibliography and Links
6. Stable Addresses for Links

Glossary

Administrative data: data that arises as a by-product of an agency's work: licence registers, contracts, inspections, payments. Coverage is usually complete, but the fields were designed for record-keeping, not analysis.

Agent instructions: a description of data for an AI agent: field definitions, units, period, limitations, code lists, search rules. They are maintained and checked like a record. The World Bank's Data360 server serves them as MCP resources.

AI agent: a program built on a large language model that itself searches for a dataset, reads the description and takes the data through an API or an MCP server.

AI-ready data: a dataset that a program can find, understand and take: an open format, an explicit licence, provenance and a dump. A Croissant record and an MCP server complement this once the format and licence are already in place. The Open Data Institute framework (version 2, December 2025) breaks the same down into dataset properties, description, infrastructure and governance.

Anonymisation: processing data about people after which a specific person cannot be identified from the remaining fields. Removing the name is not enough: a person is often given away by a combination of age, address and occupation.

API: an entry point through which a program asks a catalog a question and gets an answer. The answer most often comes in JSON.

API catalog (developer portal): a catalog of an organisation's or country's programmatic interfaces: documentation, example requests, often key issuance and a test console. It describes services, not datasets; a good API catalog links an interface to the datasets it serves.

API gateway: a program that sits in front of an organisation's interfaces: it checks keys, counts requests and limits load. It usually comes with a developer portal, which becomes the API catalog. Examples are Azure API Management, WSO2 API Manager, Gravitee.

ArcGIS REST: the programmatic interface of Esri's ArcGIS servers and cloud. Not a standard, but because ArcGIS is so widespread, GIS programs and many harvesters read it on a par with OGC standards.

ARK: a persistent identifier of the form `ark:12148/…`, issued by libraries, archives and museums. An organisation number is free, and ARKs can be resolved on your own server or through the shared resolver n2t.net.

AsyncAPI: a format for describing interfaces that do not answer requests but send a stream of messages: transit delay events, sensor readings. It does for streams what OpenAPI does for ordinary requests.

Authentication and authorisation: authentication checks who the user is: a password, a second factor, an organisational account or a government login system. Authorisation checks what they may do: a role, rights to a dataset, an approved access request. Single sign-on moves only the first

to another system; the catalog still grants the rights itself.

Automatic table API: the catalog itself parses a published tabular file and serves an API for it: rows, filters, a column profile. The publisher uploads nothing. An example is the data.gouv.fr tabular API. Such an API inherits the errors of the source file. More in the chapter on innovative catalogs.

BagIt: a standard for packaging files for archival storage and transfer: the data sits in a folder next to a manifest of checksums, so the package shows whether all files are in place and unchanged.

Benchmark: a set of tasks and data on which different machine learning models are compared under the same rules.

BibTeX, RIS, CSL: formats in which a catalog serves a dataset citation to a reference manager. BibTeX is for LaTeX; it has no separate type for data, which biblatex added (`@dataset`). RIS is for Zotero, EndNote and the like; its data type is `DATA`. CSL is the styles by which a citation is formatted to a journal's requirements or a national standard such as GOST, plus the CSL-JSON format for transferring the citation itself.

BIDS: a standard for organising neuroimaging data: the order of folders, file names and descriptions. OpenNeuro checks datasets against it on upload.

Broken (dead) link: a link to a file or page that no longer leads to the data: the server answers "not found", the domain does not exist, the certificate is not accepted. A "soft" failure is more dangerous: the link opens, but leads to the site's home page or to a page instead of a file, and a program that looks only at the response code counts it as working.

Bulk download: the ability to download the whole database or the whole dataset as files, without querying one indicator or record at a time.

Catalog capability: something a catalog can do for a person, a program or an owner: serve descriptions in DCAT, issue a DOI, close a file while keeping the description open. The same capability may be in the platform core, in an extension, or absent.

Catalog of catalogs: a list of data portals and repositories rather than datasets. It helps understand where to look for data on a topic or country. Examples are re3data, dataportals.org and the dataportals-registry.

Catalog record: information about how a dataset got into the catalog: when it was added and when the description was edited. It differs from information about the data itself: the record edit date is not the data update date.

Catalog requirement: a function chosen for your catalog, recorded together with the source of the need, the level of obligation (mandatory, desirable, not needed), the rationale, the method of verification and the condition for review. A good requirement says what is needed, not how to do it: "the national portal harvests descriptions in DCAT-AP", not "install ckanext-dcat".

Catalog self-audit: one pass through your own or someone else's catalog: a description dump, a sample of records, a file count and a page with what to fix in the coming month. It is a check of a specific site. A country ranking and an archive certificate answer other questions.

Catalog type: the class of a catalog by what is published in it and what gets its own page: a dataset, a map layer, a dataset version with a DOI, an indicator, a survey and so on. It sets the description standard and the range of suitable platforms.

CDIF: CODATA recommendations on which existing standards to combine when data crosses the boundary of a scientific discipline. For discovery it recommends schema.org and DCAT rather than a new catalog vocabulary. Version 1.1 came out in June 2026.

Change feed: a list of the most recently added and changed catalog records in RSS or Atom format. Harvesting programs subscribe to it. In CKAN the feed holds the last 20 records by default, so a full harvest also needs a "modified since date" query.

Change log: a record of who changed what in a dataset description and when, with the ability to view the previous state. In CKAN it is the `activity` module with diffs between versions; in research repositories dataset versions play this role.

Check on entry: the catalog does not publish a dataset that has not passed a check against a standard or schema. Examples are BIDS at OpenNeuro, NWB at DANDI. More in the chapter on quality levels.

Checksum: a short string computed from a file's contents by an algorithm such as MD5 or SHA-256. It changes if even one byte changes, so it is used to check that a downloaded file is intact and is the same file that was published. In DCAT 3 it is recorded with the `spdx:checksum` property.

Citation meta tags: lines in the head of an HTML page (`citation_title`, `citation_author`, `citation_doi` and others, as well as Dublin Core fields) by which Zotero and Google Scholar understand what they are looking at and save a citation with one click. A version and the "dataset" type cannot be conveyed by these meta tags.

CITATION.cff: a text file in the root of a repository saying how to cite the data or software. GitHub shows a "Cite this repository" button based on it. Zenodo takes the authors and title from it when archiving a GitHub release if there is no `.zenodo.json` alongside; if that file exists, Zenodo reads only it. For data, `CITATION.cff` needs the line `type: dataset`. The record type in Zenodo is set separately, with the `upload_type` field.

Cloud Optimized GeoTIFF (COG): a GeoTIFF file organised so that a program reads the needed fragment of an image over the network without downloading the whole file.

Cloud service (SaaS): a platform that is not installed on your own server but rented from a provider by subscription. Updates and hosting are on the provider's side, but both the data and the capabilities depend on the contract.

Code list (classification, reference data): a list of codes and names: regions, economic activities, currencies. Other datasets refer to it, so a code list's version and date matter.

Codebook: a description of a survey file: variables, question texts, answer codes and their labels, often with frequencies. Without it a microdata file is columns of meaningless numbers. The machine-readable form is DDI Codebook.

Content negotiation: a way to get different representations from one address: a browser at `https://doi.org/...` gets the dataset page, while a program that asked for BibTeX or RIS in the `Accept` header gets a ready citation from DataCite.

Controlled vocabulary: an agreed list of terms or codes from which a field value is chosen instead of free text: topics, languages, file types, countries. A thesaurus adds synonyms and broader/narrower relations. A good vocabulary gives each term a persistent address and is published in SKOS.

Coordinate reference system: the rule by which coordinate numbers turn into a point on the Earth. Each layer has its own; it is given as a code from the EPSG registry, for example EPSG:4326 for WGS 84. A layer without a coordinate system code cannot be reliably combined with others.

CoreTrustSeal: a mark showing that a repository can preserve data for the long term, not just show it on a site.

CRIS: a current research information system for an organisation's research: publications, projects, grants, staff and also datasets. An example is Elsevier Pure. Datasets in it often serve as a university's hidden data catalog.

Croissant: a machine-readable description of a machine learning dataset on top of schema.org: which files, which fields, how the dataset is split into training and test parts. Developed by MLCommons. Version 1.1 (January 2026) added data provenance and terms of use; the RAI extension describes biases, limitations and the dataset's intended use. In 2026 platforms still serve descriptions according to version 1.0.

Crossref: a DOI registration agency for articles, books, preprints and other publications, operating under US law. DataCite is usually chosen for datasets, but InvenioRDM and Dataverse can also register DOIs through Crossref.

CSV on the Web (CSVW): a 2015 W3C Recommendation: a data dictionary for a CSV table in a separate JSON file, with column names, types, units and links to other tables. Google reads such a dictionary from a dataset page's markup in experimental mode and warns that the Rich Results Test complains about this block.

CSW: the protocol by which a program queries a geoportal's layer catalog.

Darwin Core: a standard for describing observations and specimens of living organisms: which species, where and when. GBIF collects data in it.

Data aggregator: a service that collects metadata from many catalogs and offers a single search over it. An aggregator has its own page for each collected dataset, which ordinary search engines see; a dataset search engine creates no such page and sends the user straight to the source.

Data as a product (data product): an approach in which a dataset has an owner, known users, an update schedule, documentation and quality control, like any product; access is usually provided in several ways at once: API, dump, slices. The opposite is a file uploaded once to tick a box.

Data catalog: a service that describes datasets with metadata and helps people find, understand and obtain them.

Data citation: a reference to a dataset in an article, report or another dataset, formatted like a reference to a publication: author, year, title, version, publisher, identifier. It differs from a licence's attribution requirement: the licence sets a legal condition, while the citation gives scholarly credit and makes it possible to find exactly the data on which a conclusion was based.

Data cube: an indicator table with all combinations of breakdowns: years × regions × sex and so on. The needed slice is cut from the cube instead of downloading separate files by year. A term from statistical standards (SDMX, RDF Data Cube).

Data dictionary: a description of a table's columns: name, meaning, unit of measurement, permitted values. Without it a file is formally open and unclear in substance.

Data element: a description of a single field shared by many systems: the concept (object and property, for example "person — sex"), permitted values, definition, steward, version and status. Registries under ISO/IEC 11179 describe fields this way.

Data hub: several catalogs of different types under one banner with a common search, such as the dataset portal, indicator catalog and microdata catalog of one international organisation. Not a separate type: each catalog in a hub is assessed by its own type.

Data inventory: a list of what an organisation already produces or holds, including what cannot be published. From this list, what to publish is then selected and a queue recorded. An inventory is not a catalog itself.

Data management plan: a document accompanying a grant application: what data will arise in the project, in what formats, where it will be stored, and in which repository and on what terms it will be published. Horizon Europe, NIH and many other funders require one.

Data marketplace: a platform where data is sold or issued under contract: a file, a subscription, access to an interface or to a table in the cloud. It differs from an open data portal by having a contract and terms for each issue. Government marketplaces are often called data exchanges.

Data owner: the person or unit responsible for a dataset's rows: whether they are correct, when to update them and who answers about an error. The publisher in a record is an organisation. The owner may be a specific person within it.

Data paper: a peer-reviewed article that describes a dataset and links to it in a repository, without research hypotheses or conclusions.

Data service: a programmatic interface through which a dataset's data is obtained in parts: table rows with a filter, features on a map. In DCAT since version 2 it is a separate entity with its own address, referred to by a distribution.

Data space: an agreement between organisations to exchange data, including data that is not for everyone. The dataset description can be read, while the file may stay closed.

DataCite: a non-profit organisation that registers DOIs for data, and the description schema of the same name that is filled in at registration.

Dataset: a published collection of data with a common description.

Dataset card: a human-readable description of a machine learning dataset: contents, how it was collected and labelled, known biases, limitations and what the dataset is not suitable for. On Hugging Face it is a `README.md` file with a metadata section at the top. A similar description of a model is called a model card.

Dataset licence: the terms on which a specific dataset may be used. It is a separate text from the rules of the whole site and from the conditions of access to a file. An open licence allows copying, changing and using the data, including for commercial purposes; it may require attribution of the source but not prohibit commercial use. Common ones are CC BY 4.0, CC0 (a waiver of rights as far as the law allows) and ODbL (for databases: a derived database is shared on the same terms). The same licence is written in different ways: the SPDX code `CC-BY-4.0`, the CKAN code `cc-by`, the EU vocabulary link to the concept `CC_BY_4_0`. The phrase "open licence" and the value "not specified" are empty to a program.

Dataset list: a page, table or README file with links to datasets or to other catalogs, without search or records. A normal way to publish a dozen datasets, provided each row has a licence, a date and a persistent address.

Dataset search engine: a service that collects dataset descriptions from many catalogs or from site pages and offers one search over them. It publishes no datasets of its own; files stay at the source. It may be general (Google Dataset Search, Dateno), scholarly (OpenAIRE, DataCite Commons), sectoral (NIAID Discovery Portal) or national (find.data.gov.scot). If each collected dataset has its own page for ordinary search engines, it is called an aggregator.

Dataset series: datasets of the same kind released in turn: budget execution for each month, a census every ten years. A separate class for series appeared in DCAT 3 (2024); most portals do not know it and create a separate record for each release.

Dataset splits: dividing a machine learning dataset into a training part (the model learns on it), a validation part (settings are tuned on it) and a test part (the result is measured on it). The split is fixed in advance; otherwise results by different authors cannot be compared.

Dataspace Protocol: the exchange protocol for data spaces maintained by the Eclipse Foundation (version 2025-1). A participant publishes a dataset catalog in DCAT with conditions in ODRL, and the protocol describes contract negotiation and data transfer.

DCAT: the W3C vocabulary for describing catalogs, datasets, distributions and data services. Version 3 is a Recommendation of 22 August 2024: it introduced dataset series and version properties.

DCAT-AP: the DCAT profile for European portals: which set of description fields they exchange with one another. The current recommendation is 3.0.1 of 27 October 2025. The 3.0.2 draft of 17 June 2026 has not yet become a recommendation.

DCAT-US: the DCAT profile for US federal agencies. An agency publishes a `data.json` file listing its datasets, and data.gov harvests it. Version 3.0 was published on 1 May 2026 and is built on DCAT 3. data.gov accepts version 1.1 for a transition period.

DDI: a standard for describing surveys and microdata: who asked, whom, how the sample was built and what the variables mean. A family of several formats: DDI Codebook describes a single study (main version 2.5, new 2.6 in 2026), DDI Lifecycle the whole cycle with questionnaires and waves (3.3), DDI-CDI data from any field (1.0, 2025). Since 2026 the common part has been the ISO/PAS 25955 standard.

Decision record: a short record of one decision point when choosing a catalog: context, options, criteria, decision, rationale, consequences, and the event after which the choice is reopened. Written for the platform, a mandatory extension and dropping a feature. A superseded record is not deleted but marked as superseded.

Delta file: a file containing only what has changed since the last full dump: new, changed and deleted records. It avoids downloading a large database in full at every update.

Derived dataset: a DOI-bearing record for data that a user has filtered or assembled from several downloads, with a list of source datasets. This is how GBIF works.

Dimension (breakdown): an attribute by which an indicator's values are divided: year, region, sex, age, industry. The permitted values of a dimension come from a code list.

Distribution: a specific form of a dataset: a file, a link, an API, a service. A DCAT term. In CKAN it is called a resource, in Dataverse and Zenodo a file, in STAC an asset. DCAT distinguishes a direct link to a file (downloadURL) from an access address (accessURL), a page or service through which the data can be obtained.

DOI: a persistent identifier for a digital object of the form `10.5281/zenodo.16546069`: an organisation prefix and a suffix the organisation makes up itself. The link `https://doi.org/…` redirects to the landing page. For data, DOIs are usually registered with DataCite, for articles with Crossref; technically it is a Handle with the prefix `10.`. A DOI cannot be deleted. A concept (parent) DOI unites all versions of a dataset and leads to the latest.

Dolt: an SQL database compatible with MySQL, with versions as in Git: clone, branches, merge, cell-level diffs. DoltHub hosts public databases. More in the chapter on Git repositories.

Domain data standard: a shared format for one topic that describes not a file in general but a subject: a contract (OCDS), a transit schedule (GTFS), a species occurrence (Darwin Core). Programs and aggregators understand data in such a format straight away.

Dublin Core: a short set of description fields: title, author, date, subject. Research repositories often serve it.

Embargo: a period during which a dataset's description is open while the data itself is closed; after the date the data opens. Usually in research repositories until an article is published.

Extension (plugin): a separate module with its own author, releases and compatibility with a core version. It is installed next to a platform to gain a capability not in the distribution. For example, ckanext-dcat adds DCAT export to CKAN. A setting or a module from the same distribution that is merely switched on is not an extension: it is released together with the core.

External tool: a separate program that the catalog opens to view or process a file. The program's code is not part of the catalog: the catalog stores its address and passes it the file identifier. Dataverse previewers work this way.

FAIR: the principles of findability, accessibility, interoperability and reusability of data (2016). FAIR does not mean "open": a dataset with closed files can also comply if it has an open description, a persistent identifier and a clear access procedure.

FAIR Data Point: a small service through which an organisation tells, in DCAT, what data it has. The data itself stays with the owner.

FAIR Signposting: a way to tell a program the essentials about a dataset without parsing HTML: the dataset page lists, in a response header or in a separate file (a linkset), links to the description, files, licence and author identifiers. It is in the core of Dataverse, InvenioRDM and DSpace.

Five-star open data: Tim Berners-Lee's ladder (2010) for one dataset: from a file under an open licence to a machine-readable table, an open format, addresses inside the data and links to other people's data. The ladder speaks about the format of a single file.

Fork: a copy of someone else's repository maintained by another team. Copies diverge: you install the one that is tested on your core version and record its address. An edit kept only in-house and not contributed back also becomes a fork, and then the maintainer is you.

Frictionless Data Package: a standard for describing a dataset with a file next to the data. The published descriptor is called `datapackage.json`. The current version of the standard is 2.0.

Gated access: a dataset whose description is open to everyone, while the files are issued after a request: the user provides information about themselves, accepts terms and sometimes waits for the author's approval. Gated datasets on Hugging Face work this way; to a dataset search engine such a dataset is a dataset without files.

Generalist repository: a research repository that accepts data from any discipline and any author: Zenodo, Figshare, Harvard Dataverse, Dryad. Funders recommend one when a discipline has no subject repository of its own.

Geoblocking: restricting access to a site for visitors from other countries based on their IP address. Along with foreign users, web archives, search engines and data aggregators stop seeing the site.

GeoDCAT-AP: the DCAT-AP profile for geodata: how to translate an ISO 19115 layer record into a DCAT description that open data portals understand. data.europa.eu reads national geocatalogs through it.

GeoPackage: an open OGC format for geodata: a single file based on an SQLite database holding vector layers, rasters and their description. It opens in QGIS and other programs without a server.

Geoportal: a catalog of spatial data and services, often combined with a web map.

Git LFS: an add-on to Git for large files: the files themselves are stored separately, and the repository history keeps links to them. The platform has its own file size limits and storage and download quotas for this.

GTFS: a format for public transport schedules: stops, routes, trips. The GTFS Realtime extension carries vehicle positions and delays in real time.

Handle: the persistent identifier system on which DOIs are also built. A prefix is obtained from the Handle.Net Registry for an annual fee, and you run your own Handle server. DSpace issues Handles by default, but with the placeholder prefix `123456789`, which means nothing outside the site until the organisation registers its own.

Headless setup: the catalog platform works only as a core: it stores descriptions, serves an API and harvests other catalogs, while the site for people is built separately. The Toronto and Canada portals on top of CKAN, for example, work this way.

HealthDCAT-AP: the DCAT-AP profile for describing health data: the access body, the data category, the number of people in the dataset, age coverage, the legal basis. The European Commission is preparing it for the catalogs required by the European Health Data Space Regulation (EHDS); in 2026 it is a draft specification.

High-value datasets: six groups of data that EU public bodies must, under Regulation 2023/138, make available free of charge, in machine-readable form and via API: geospatial, earth observation and environment, meteorological, statistics, companies and company ownership, mobility.

Hybrid (hybrid catalog): a site on which products of different types live side by side, such as datasets and map layers, or an organisation that runs catalogs of several types. Not a separate type: each part of a hybrid is classified and assessed by its own type. The most developed form is a data hub.

IATI: a standard for disclosing data on development aid: who funds a project, how much, where and with what result. International organisations, donors and foundations publish in it.

Indicator catalog: a catalog of official statistics in which the unit of publication is an indicator with breakdowns by year, territory and other attributes. The user picks the slice they need rather than downloading "the file for a year".

INSPIRE: EU rules for spatial data: how to describe layers and how to serve them through a catalog. In force since 2007; since December 2025 the directive is being simplified; the separate INSPIRE Geoportal closed on 1 July 2026, and its role passed to data.europa.eu.

International Open Data Charter: six principles for publishing government data, agreed in 2015 by governments, civil society organisations and experts: open by default; timely and comprehensive; accessible and usable; comparable and interoperable; for improved governance and citizen engagement; for inclusive development and innovation.

ISO 19110: a feature catalogue standard for geodata: which feature types a layer contains and what attributes they have. In essence, a data dictionary for a map layer. Supported in the GeoNetwork core.

ISO 19115: a standard for describing a map layer: what is depicted, for what period, who the author is, in which coordinate system. Records are exchanged in XML according to ISO 19139; the new edition is encoded according to ISO 19115-3.

ISO/IEC 11179: a standard for metadata registries: how to describe an individual data element, its definition, permitted values, owner and status.

JSON: a text format in which programs exchange data. A person reads it less easily than a table, a program more easily than a PDF.

JSON Lines: a file in which each line is a separate JSON record. It is read line by line without loading the whole file into memory, so it suits large dumps with nested fields.

JSON-stat: a lightweight format for a statistical table in JSON: values, dimensions and labels in one response. Version 2.0 is the default format of the Eurostat API and PxWebApi 2.

Knowledge graph: a database where every variable, place or entity has its own identifier and the links between them are recorded explicitly. It allows a question to be answered across many sources at once. Data Commons works this way.

Landing page: the page a DOI or other persistent identifier leads to: the dataset description, a ready citation, the path to the files. The identifier itself does not store files, so without a live landing page it leads nowhere.

llms.txt: a short text file about a site for a language model: what is published here and where to look.

Long-term support (LTS) release: a data release that is promised to be kept unchanged for a named period. For CELLxGENE Census it is at least five years. The period is written down in advance, not "while the server runs".

Map server: a program that serves map layers to other programs via WMS, WFS, ArcGIS REST and similar protocols. Examples are GeoServer, MapServer, QGIS Server, ArcGIS Server. It usually keeps no search records itself, so without a catalog alongside, a layer can only be found by someone who knows the address.

MCP: the Model Context Protocol, by which a catalog gives an AI agent a ready list of queries, code lists and instructions so that it does not guess addresses. An MCP server runs on top of the catalog's API. A good server is read-only, needs no key for open datasets and links to the dataset in its answer. More in the chapter on AI.

Metadata: data about data: title, description, publisher, licence, coverage, format.

Metadata completeness and depth: completeness shows what share of the necessary description fields is filled in across a catalog's datasets; depth, how detailed the description schema itself is. A third property is accuracy: whether what is in the field is true. A placeholder value such as a period of "1900–2099" makes a field formally filled in but not complete in substance.

Metadata harvesting: the automatic periodic retrieval of one catalog's metadata by another.

Metadata profile: a refinement of a general vocabulary for a country or sector: which fields are mandatory, which are recommended, and which code list a value comes from. The vocabulary itself does not change: DCAT-AP and DCAT-US are profiles of the same DCAT.

Metadata registry: a catalog that stores not data but its descriptions, with versions and statuses: data elements, code lists, table structures. The standard for data element registries is ISO/IEC 11179, for statistical structures SDMX.

Metadata standard: a shared language for describing a dataset from the outside: what it is, who the publisher is, for what period, where to download it and on what terms. Other catalogs, search engines and programs read it. Examples are DCAT, DataCite, ISO 19115, schema.org. It differs from a data standard in not describing the columns inside the file.

Microdata: records about individual people, households or organisations from censuses and surveys, as opposed to ready summary tables. They are anonymised before publication, and access to the files is often restricted.

Microdata harmonisation: bringing the variables of different censuses and surveys to the same codes and definitions so that countries and years can be compared. This is how IPUMS works.

ML catalog: a catalog of datasets, and often of models, for training and evaluating machine learning systems. A dataset in it is split into parts, loads from code and is described by a card with biases and limitations.

ML hub: a shared hub to which anyone uploads datasets and models: Hugging Face, Kaggle, ModelScope. A data hub is something else: several catalogs of one organisation under a common banner.

Moderation (review before publication): a step at which a curator, moderator or organisation administrator looks at a prepared dataset and either releases it or returns it to the author. It differs from a draft by involving a second person: an author publishes a draft themselves. User texts are moderated too: data requests, comments, discussions.

MQA: an assessment of the records data.europa.eu has harvested from a portal: whether DCAT-AP fields are filled in and whether links respond. A catalog's score ranges from 0 to 7.5 and is the average over the descriptions the assessment has counted. A portal that sends nothing to the European catalog gets no score.

NetCDF: a format for multidimensional arrays: variables on a grid over time and space. With the CF conventions it holds climate, oceanographic and atmospheric data.

OAI-PMH: the protocol by which research repositories serve record descriptions to a harvester. It carries only descriptions, without a file list, so a record harvested through it often does not lead to the data itself.

Object identifier: a persistent identifier for an individual record inside the data: a building, a road, an organisation. Different datasets are joined by it without matching names and geometries. An example is GERS in Overture Maps. More in the chapter on identifiers.

Object storage: a cloud or self-hosted file store where each file sits under its own key in a "bucket" and is served over HTTP: Amazon S3, Azure Blob Storage, Google Cloud Storage and S3-compatible systems such as MinIO and Ceph. Space is almost unlimited, but storage and downloads are paid for. Most large catalogs keep uploaded files there.

OCDS: the open data standard for procurement: the whole path of a contract from planning and tender to implementation.

ODIN: an assessment of countries' official statistics run by Open Data Watch. It looks at the statistical office site and the pages it links to: whether there is data on 22 topics and whether it is open. A country's open data portal may not be part of this assessment.

ODRL: a W3C language for machine-readable terms of use: what is permitted, what is prohibited, what must be done. Used in data spaces.

OGC API: a family of modern web interfaces to geodata from the OGC consortium: Features for features, Records for a layer catalog and others. Gradually replacing WFS and CSW.

Ontology: a rigorous description of a field of knowledge: classes of concepts, their properties and rules by which a program can draw inferences. Stricter than a vocabulary: besides broader/narrower it specifies, for example, that a gene has a location on a chromosome. Usually written in OWL; ontology catalogs include OLS, BioPortal, AgroPortal.

Open Data Maturity: an annual assessment of European countries since 2015: policy, portal, metadata quality and impact, in equal parts. It is a survey of national representatives, not a crawl of files. 36 countries took part in the 2025 report.

Open data portal: a catalog through which a public body or organisation publishes datasets under open licences: registers, budgets, statistics, transport information. Three conditions together make data open: machine-readable form, an open licence and access without applications or registration.

Open dataset passport: the Russian description of a dataset under methodological recommendations version 4.0 (minutes of 12 December 2024, No. 49pr): identifier, title, owner, file link, format and dates. The machine-readable form is a `meta.csv`, `meta.xml` or `meta.json` file. It is not a DCAT profile: international harvesters and dataset search engines do not read it.

OpenAIRE: European open science infrastructure. It collects descriptions of publications, datasets and software from repositories via OAI-PMH and from DataCite into a shared graph of research outputs and provides search over it (OpenAIRE Explore). The OpenAIRE CONNECT service builds separate gateways over this graph for research communities.

OpenAPI: a format for describing a programmatic interface: which requests it accepts and what it returns. Documentation is built and requests are validated from the description.

openEO: an open protocol for processing satellite data on a server: the user sends a chain of operations, and the server returns the result.

OpenID Connect: an identity standard on top of the OAuth 2.0 protocol: the catalog sends the user to log in to an external system and receives confirmation of who they are. Keycloak, cloud account systems and the French ProConnect run on it.

OpenSharing (Delta Sharing): an open protocol by which a cloud platform gives a buyer access to a provider's tables without copying the data. Created by Databricks as Delta Sharing in 2021, it became OpenSharing in June 2026 and moved to the Linux Foundation.

ORCID: a persistent identifier for a researcher. Datasets and articles are linked to an author by it, even if their name is spelled in different ways.

Orthophoto: an aerial or satellite image corrected so that it can be overlaid on a map and used to measure distances. One of the most sought-after geoportal layers.

OWL: the W3C ontology language: classes, properties and inference rules on top of RDF. Many life science and agriculture ontologies are written in it.

Pagination: an API or dump response split into pages with a set number of records. A program goes from page to page until the last one. Many APIs have a depth limit: pages beyond a certain record are not served.

Parquet: a columnar table format: it stores column types and compresses data by column. A file is several times smaller than CSV, and analytical programs read it faster.

PC-Axis (PX): a file format for statistical tables with a built-in description of dimensions and units. PxWeb's native format; many open data portals serve it too.

Persistent identifier (PID): a name for an object that does not change, plus a service (a resolver) that sends you from that name to the object's current address. Examples are DOI, Handle, ARK. Persistence is ensured not by technology but by an owner who updates the address at every move.

Piveau: a free catalog platform built around DCAT and harvesting other catalogs. data.europa.eu runs on it.

Platform core: the capabilities included in the software distribution. This also includes a module from the same distribution that is merely switched on in settings: it has the same author and the same update cycle. A separate repository with its own compatibility is already an extension.

Processing next to the data: the catalog serves not a file but a selection or the result of a calculation performed on the server. Examples are ERDDAP, openEO, Climate Data Store.

Protected data: public sector data that cannot be published openly: personal data, trade secrets, statistically confidential data. Under the European Data Governance Act their descriptions are collected in a single information point in each country, and the data itself is issued on request under conditions that preserve protection.

Public use file: an anonymised microdata file issued after registration and agreement to terms, without an application review. More detailed and stricter is a licensed file on an approved application; stricter still is working only in a trusted research environment.

Publicly available and open data: publicly available data can be viewed on a site or map; open data can be downloaded in full in machine-readable form, under a free licence and without applications. Free issue on application is not yet open data.

Publisher: the organisation responsible for a dataset in substance and legally. It may differ from the organisation in the catalog software under whose name the dataset was created.

Publishing editor: the person who uploads a file and fills in a record according to a template. They do not set the licence or the content: the lawyer and the data owner do that.

PURL and w3id.org: free persistent address services: an address on purl.org or w3id.org redirects to where the resource currently is, and when it moves only the redirect rule changes. purl.org is run by Internet Archive, w3id.org by the W3C community; rules are added to it by a pull request to a GitHub repository.

Quality metric: a share of datasets with a recorded denominator, a rule for "what counts as filled in" and a calculation date. Such a share shows what to fix. The number of datasets in a site footer has none of these notes.

RDF: a general model for recording information as "subject, property, value" statements. DCAT, SKOS and other linked data vocabularies are built on it.

re3data: an international registry of research repositories: subjects, access conditions, certificates.

Reference manager: a program where a researcher keeps their sources and from which they insert citations into a text in the required style: Zotero, EndNote, Mendeley, JabRef. It accepts citations as BibTeX, RIS or CSL-JSON files and can save them straight from a page via meta tags.

Reproducibility: the ability to obtain the stated result again from the published data and the recorded steps. An "open data" badge and a DOI say that the file is published and can be cited. Whether the calculation matches the stated result is checked separately.

Requester Pays: a cloud bucket mode in which the owner pays for storage and whoever reads the file pays for the download. A cloud account is needed for this. Such a file cannot be downloaded anonymously, so the mode does not suit open data.

Research repository: a catalog that accepts and preserves research data for the long term, issues persistent identifiers for it (most often DOIs) and helps cite it.

Reuse: an application, article, study or visualisation built on a dataset and linked to its record on the portal. Reuses show who needs the dataset; data.gouv.fr has a model section of this kind.

RFC 9727 (api-catalog): a 2025 IETF standard: at `/.well-known/api-catalog` a site serves a program a list of its interfaces. Rare in 2026.

RO-Crate: a way to package research files together with a description: next to the data sits a `ro-crate-metadata.json` based on schema.org. This is how a research object is deposited and transferred. A national catalog is not harvested with this file. Recommendation 1.3 is dated 22 June 2026.

Role: a set of rights given to a person within an organisation, collection or community: a member sees private datasets, an editor creates and edits, an administrator also manages people. Rights are given to roles rather than people so that individual permissions do not have to be gone through when staff change.

ROR: an open registry of research organisations with persistent identifiers. In a dataset description, a ROR link identifies the author's or publisher's organisation unambiguously.

SAML 2.0: a 2005 OASIS standard for single sign-on: an organisation's account system passes the catalog a signed assertion of who logged in and what attributes they have. The basis of university login federations and Shibboleth.

schema.org/Dataset: markup on a dataset page by which search engines find it. Google requires a description of 50 to 5,000 characters.

SDMX: a standard for exchanging statistics: not a file for one year but a series with dimensions. It includes a model for describing tables (dimensions, code lists, attributes), the SDMX-ML, SDMX-JSON and SDMX-CSV formats and API rules. It is maintained by seven international organisations, among them Eurostat, the IMF and the OECD; the current version is 3.1 (2025), but most services work with 2.1.

Search index: a separate copy of dataset descriptions built for fast search and filtering; in catalogs it is usually Solr, Elasticsearch or OpenSearch. If the index has diverged from the description database, a dataset exists but cannot be found.

Shared hub: someone else's platform where an author publishes their dataset together with a record: Zenodo, Figshare, Hugging Face, Kaggle, Source Cooperative. A data hub is something else: several catalogs of one organisation under a common banner. An ML hub is a kind of shared hub where models sit next to datasets.

Signed (temporary) link: a file address in object storage with a signature and an expiry, after which it stops working. Portals issue one on download so that the file comes straight from the cloud: on HDX such a link lives for a few minutes. It cannot be cited in an article or a program; only the file's persistent address on the portal can.

Single sign-on (SSO): logging in to a catalog with an account from an external system (an agency, a university, a cloud provider or a government identity system) rather than a separate portal password. The external system checks identity using SAML 2.0 or OpenID Connect. A departing employee loses access to the catalog together with the organisation's other systems.

Sitemap: a file listing page addresses. A search engine crawls a site by it.

SKOS: a way to describe a vocabulary of terms and the links between them (a 2009 W3C Recommendation): each concept has an address, names in different languages, a code, broader/narrower relations and mappings to concepts in other vocabularies.

Software class: the kind of program a catalog is built on: a cloud service, a ready-made platform on your own server, a map server or protocol server, a CMS site, in-house development. The class determines who updates the program and what leaving costs; the catalog type determines what is published in it.

Spatial data infrastructure (SDI): a shared system of rules, catalogs and services through which a country's or region's authorities describe and serve their layers. The visible part is a geoportal and a shared catalog; in the EU such infrastructures are built under the INSPIRE directive.

SPDX: short licence codes, so that a program can recognise a licence without parsing legal text.

STAC: a protocol for describing spatial files: where a file is, what territory and date it relates to, how to download it. It arose for satellite imagery and is now used for other geodata too. It is not a program: STAC is served by separate servers, static sets of JSON files and modules of general-purpose platforms such as GeoServer.

StatDCAT-AP: an extension to DCAT-AP for statistical tables: dimensions, attributes, a link to an SDMX structure. It is needed so that a statistical office's tables appear on an open data portal. The current version is 1.0.1 (2019). Since 10 June 2026 version 3.0.0 has been a working draft aligned with DCAT-AP 3.0.1.

Static site: a site made of ready HTML files. A generator builds them from sources, and the hosting only serves the files: there is no separate database or application server.

Synthetic data: artificial records that reproduce the statistical properties of real ones but do not describe real people. They are published instead of closed microdata.

Table Schema: a Frictionless family standard for a table's data dictionary: column names, types, permitted values.

Table store: the catalog's database into which rows of tables from files are loaded so they can be shown in the browser, filtered and served in parts through an API. In CKAN it is DataStore on PostgreSQL; in cloud services such as Huwise and Socrata the dataset is a table in their database. A full dump is still needed alongside.

Technical specification: the document by which a contractor builds or configures a system: what it must be able to do, on what terms and how the result will be checked. For a data catalog it is above all a list of requirements.

Template portal: the same portal build that a government or vendor hands out to dozens or hundreds of public bodies: each has its own showcase, but the software, description fields and dump are shared. Examples are GD Catalog in Thailand, BODIK ODCS in Japan, Municipium in Italy.

Test data contamination: the test part of a benchmark getting into a model's training data, most often along with texts from the internet. A model's result on such a benchmark is inflated. Protection is hidden answers for the test part and special markers in files by which they are filtered out of training sets.

Theme and tag: a theme is taken from a shared code list (the EU Publications Office themes, ISO 19115 topic categories), so datasets from different portals add up by it. A tag is the publisher's free word; to a program, "transport" and "Transport" are different tags.

Tiered access: access to a dataset in tiers: description, variables and summary statistics, a sample, all data. Each tier is opened separately: to everyone, to specific people or after conditions. An example is Redivis.

Time series: values of one indicator for the same breakdowns, arranged in time: the dollar-to-euro rate by day, a region's population by year. In a good indicator catalog a series has a persistent address built from code-list codes.

Tombstone page: a page left in place of a withdrawn dataset with a DOI: the citation, the DOI itself and an explanation of why the data is no longer there. It is needed because a DOI cannot be deleted.

Trust score: an optional catalog score in the dataportals-registry, from 0 to 100. It is made up of the owner type, the catalog type, the licence, a re3data record and whether the site responds. It is a filter for thousands of other people's catalogs. It is not a certificate and does not open files.

Trusted research environment: a secure environment where an approved researcher works with confidential data without downloading it; results are checked before release. An example is SecureLab at the UK Data Service.

Undocumented (implicit) API: a programmatic interface that a site serves although the owner has not described it: the API from which the page itself gets its data, or a platform's default interface such as WordPress's `/wp-json/`. Useful for retrieving data, but it changes without warning.

Version and revision: a dataset version is a fixed state of the data that can be cited; in research repositories each version has its own identifier. A revision is a minor edit to the description within one version and does not create a new version.

VIVO: free software for an organisation's researcher profiles: people, publications, projects, sometimes datasets. It is not a data catalog in the strict sense, but its installations appear among search engines in registries.

WARC and WACZ: web archive formats. WARC stores page snapshots together with server responses; Internet Archive and national libraries read it. WACZ is the same archive packed into a ZIP together with an index; it can be opened in a browser without a server.

Web archive: a collection of snapshots of sites taken by a crawler on different dates. The largest is Internet Archive (web.archive.org); national libraries and archives keep their own web archives. It preserves pages well, and data from a programmatic interface and large files poorly.

WMS and WFS: OGC protocols for geodata. WMS serves a picture of a layer for viewing, WFS the features themselves, which can be downloaded and processed. WMTS serves a map in pre-cut tiles, so it is faster than WMS but also only for viewing.

XBRL: a financial reporting language in which each line of a report is tagged with a machine-readable code. Companies in many countries file their reports to regulators in it.

XKOS: a SKOS extension for statistical classifications from the DDI Alliance: classification levels, correspondence tables between editions, "includes" and "excludes" notes.

Zarr: a format for a multidimensional array made of separate chunks. A program reads the needed chunk over the network. Climate fields, model outputs and other arrays that do not fit in one file are published this way.

Zenodo: a general research repository at CERN. It accepts data from any discipline, issues DOIs and can automatically archive GitHub repository releases.

Using the Registry

The book's figures come from the dataportals-registry export in DuckDB (`data/datasets/datasets.duckdb`), release 1.27.0 of 15 October 2026, with queries run on 15 October 2026. The `catalogs` table has one row per catalog, the `software` table one row per platform. All the book's figures were calculated from exactly this release. The type names below are as in the registry schema, not as in chapter titles.

How to get the export

The database file is in the registry repository itself, in the `data/datasets` folder. Next to it are the same records in Parquet and JSONL: `full.parquet` and `full.jsonl`. The registry is updated every few days, so to check the book's figures you need the release they were calculated from, not the latest state of the repository:

```
git clone https://github.com/datenoio/dataportals-registry.git
cd dataportals-registry
git checkout v1.27.0
duckdb -readonly data/datasets/datasets.duckdb
```

With the first query, check the total number of records against the total in the chapter "Catalogs in Numbers": release 1.27.0 gives 43,614.

```
SELECT count(*) FROM catalogs;
```

If the numbers do not match, you have a different export, and the other queries will give different figures too.

Catalogs by type

```
SELECT catalog_type, count(*) AS n
FROM catalogs
GROUP BY 1
ORDER BY 2 DESC;
```

Platforms within each type

```
SELECT catalog_type, sid, n
FROM (
  SELECT catalog_type, software.id AS sid, count(*) AS n,
         row_number() OVER (PARTITION BY catalog_type ORDER BY count(*) DESC) AS rn
  FROM catalogs
  GROUP BY 1, 2
)
WHERE rn <= 6
ORDER BY catalog_type, n DESC;
```

Shares of custom-built solutions, APIs and closed catalogs by type

```
SELECT catalog_type,
       round(100.0 * count(*) FILTER (WHERE software.id = 'custom') / count(*), 0) AS
pct_custom,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api,
       round(100.0 * count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) /
count(*), 1) AS pct_dead
FROM catalogs
GROUP BY 1
ORDER BY count(*) DESC;
```

Owners

```
SELECT owner.type, count(*) AS n
FROM catalogs
GROUP BY 1
ORDER BY 2 DESC;
```

Prevalence of interfaces

```
SELECT e.type, count(*) AS n
FROM (SELECT unnest(endpoints) AS e FROM catalogs)
GROUP BY 1
ORDER BY 2 DESC
LIMIT 30;
```

Licences, national catalogs, DOIs

About 420 catalogs with licence text or a link, about 2,000 with DOIs and 613 national ones: this query. The chapter on licences.

```
SELECT
  count(*) FILTER (WHERE rights.license_id IS NOT NULL
                      OR rights.license_url IS NOT NULL
                      OR rights.license_name IS NOT NULL) AS with_license,
  count(*) FILTER (WHERE properties.is_national) AS n_national,
  count(*) FILTER (WHERE properties.has_doi) AS with_doi,
  count(*) AS total
FROM catalogs;
```

18,000 with a defined licence model and 5,600 explicitly marked "unknown" are rows of the next query, not a sum with the 420. An empty value is records the chapter does not assign to any option. The slices do not add up to the total number of catalogs: licence text and licence model are different fields.

```
SELECT rights.rights_type AS model, count(*) AS n
FROM catalogs
GROUP BY 1
ORDER BY n DESC;
```

Restricted access, about 1,100 in the same chapter, is the value `restricted` in the `access_mode` list.

```
SELECT mode, count(*) AS n
FROM (SELECT unnest(access_mode) AS mode FROM catalogs)
GROUP BY 1
ORDER BY n DESC;
```

Statuses and programmatic interface

868 inactive and 98 decommissioned are the `inactive` and `deprecated` rows. 65% and 34% with and without an interface are the `api` shares.

```
SELECT status, count(*) AS n
FROM catalogs
GROUP BY 1
ORDER BY n DESC;
```

```
SELECT
  count(*) FILTER (WHERE api) AS with_api,
  count(*) FILTER (WHERE NOT coalesce(api, false)) AS without_api,
  count(*) AS total
FROM catalogs;
```

The interface share for dataset lists, about 14%, is the `Datasets list` type.

```
SELECT
  count(*) AS n,
  round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api
FROM catalogs
WHERE catalog_type = 'Datasets list';
```

Sitemaps, about 7,000, are among the types in the "Prevalence of interfaces" query. If you need only them:

```
SELECT e.type, count(*) AS n
FROM (SELECT unnest(endpoints) AS e FROM catalogs)
WHERE lower(e.type) LIKE '%sitemap%'
GROUP BY 1
ORDER BY n DESC;
```

Owners, countries, content

Local government, about 17,600, and geoportals among them, about 14,400:

```
SELECT catalog_type, count(*) AS n
FROM catalogs
WHERE owner.type = 'Local government'
GROUP BY 1
ORDER BY n DESC;
```

Countries in the chapter on numbers. A catalog with several countries is counted in each. Records without a single country form a separate row with empty or ambiguous coverage; the chapter has about 1,800 of them.

```
SELECT country_id, count(*) AS n
FROM (
  SELECT unnest(list_transform(coverage, x -> x.location.country.id)) AS country_id
  FROM catalogs
)
GROUP BY 1
ORDER BY n DESC
LIMIT 15;
```

The content breakdown in the chapter "What Gets Published": one catalog may produce several rows.

```
SELECT t AS content_type, count(*) AS n
FROM (SELECT unnest(content_types) AS t FROM catalogs)
GROUP BY 1
ORDER BY n DESC;
```

Classification: owners, access and software classes by type

The chapter "How to Classify Catalogs". Owners within each type, for example 58% of geoportals run by local government and 85% of research repositories by universities and research organisations:

```
SELECT catalog_type, owner.type AS owner_type, count(*) AS n,
       round(100.0 * count(*) / sum(count(*)) OVER (PARTITION BY catalog_type), 0) AS pct
FROM catalogs
GROUP BY 1, 2
ORDER BY catalog_type, n DESC;
```

The share of catalogs with restricted access to at least part of the data: 91% for marketplaces, 49% for microdata catalogs:

```
SELECT catalog_type,
       count(*) FILTER (WHERE list_contains(access_mode, 'restricted')) AS restricted,
       count(*) AS n,
       round(100.0 * restricted / n, 0) AS pct
FROM catalogs
GROUP BY 1
ORDER BY pct DESC;
```

Software classes are the `subtype` field in the `software` table. In the chapter they are combined into five groups: cloud service (`managed_saas_service`), ready-made platform (the `*_platform` subtypes), server or domain system (`geospatial_service_middleware`, `protocol_or_api_server`, `domain_data_infrastructure`), CMS site (`cms_or_app_framework`) and in-house development (`custom`).

```
SELECT CASE WHEN c.software.id = 'custom' THEN 'custom' ELSE s.subtype END AS class,
       count(*) AS n,
```

```
       round(100.0 * count(*) / sum(count(*)) OVER (), 1) AS pct
FROM catalogs c
LEFT JOIN software s ON s.id = c.software.id
GROUP BY 1
ORDER BY n DESC;
```

Platforms the book names

The identifier is `software.id`. The roughly 1,900 CKAN installations are counted only among open data portals, not among all records with this id.

```
SELECT software.id AS sid, catalog_type, count(*) AS n
FROM catalogs
WHERE software.id IN (
  'ckan', 'dkan', 'arcgishub', 'arcgisserver', 'geoserver', 'emapa',
  'geonetwork', 'geonode', 'jkan', 'opensdg', 'datasette',
  'opendatasoft', 'socrata', 'pxweb', 'nada', 'dspace',
  'dataverse', 'inveniordm', 'invenio', 'custom'
)
GROUP BY 1, 2
ORDER BY n DESC;
```

Open data portals

The chapter "Open Data Portals". Owners: 48% local government, 19% central:

```
SELECT owner.type, count(*) AS n,
       round(100.0 * count(*) / sum(count(*)) OVER (), 0) AS pct
FROM catalogs
WHERE catalog_type = 'Open data portal'
GROUP BY 1
ORDER BY n DESC;
```

Countries: Russia 572, US 565, Spain 485. A catalog with several countries is counted in each:

```
SELECT country_id, count(*) AS n
FROM (
  SELECT unnest(list_transform(coverage, lambda x: x.location.country.id)) AS country_id
  FROM catalogs
  WHERE catalog_type = 'Open data portal'
)
GROUP BY 1
ORDER BY n DESC
LIMIT 15;
```

Platforms with the share of APIs and inactive catalogs: CKAN 1,906 and 95% with an API, in-house development 1,144 and 19%:

```
SELECT software.id AS sid, count(*) AS n,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api,
       round(100.0 * count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) /
count(*), 1) AS pct_dead
FROM catalogs
WHERE catalog_type = 'Open data portal'
GROUP BY 1
```

```
ORDER BY n DESC
LIMIT 25;
```

Platforms for central and local government: CKAN 420 and 707, Gosweb 421 for local government only:

```
SELECT owner.type, software.id AS sid, count(*) AS n
FROM catalogs
WHERE catalog_type = 'Open data portal'
  AND owner.type IN ('Central government', 'Local government')
GROUP BY 1, 2
QUALIFY row_number() OVER (PARTITION BY owner.type ORDER BY count(*) DESC) <= 10
ORDER BY 1, n DESC;
```

The top three platforms in each country: this shows that in Russia the count comes from sections on Gosweb, in Italy from Municipium, in Japan from BODIK ODCS:

```
SELECT country_id, sid, n
FROM (
  SELECT country_id, sid, count(*) AS n,
         row_number() OVER (PARTITION BY country_id ORDER BY count(*) DESC) AS rn
  FROM (
    SELECT unnest(list_transform(coverage, lambda x: x.location.country.id)) AS country_id,
           software.id AS sid
    FROM catalogs
    WHERE catalog_type = 'Open data portal'
  )
  GROUP BY 1, 2
)
WHERE rn <= 3 AND country_id IN ('RU', 'US', 'ES', 'FR', 'IT', 'TH', 'ID', 'JP', 'CN')
ORDER BY country_id, n DESC;
```

Geoportals

The chapter "Geoportals". Owners: 58% local government. In the chapter, central government is combined with federal (`Federal government`), and "Other" is `Other` plus `Community`:

```
SELECT owner.type, count(*) AS n,
       round(100.0 * count(*) / sum(count(*)) OVER (), 0) AS pct
FROM catalogs
WHERE catalog_type = 'Geoportal'
GROUP BY 1
ORDER BY n DESC;
```

Countries: US 5,050, Poland 2,303, Czechia 1,408. The top three platforms in each country show local vendors (GISPLAN in Czechia, iObčina in Slovenia, ⊠⊠⊠⊠⊠ in Japan):

```
SELECT country_id, sid, n
FROM (
  SELECT country_id, sid, count(*) AS n,
         row_number() OVER (PARTITION BY country_id ORDER BY count(*) DESC) AS rn
  FROM (
    SELECT unnest(list_transform(coverage, lambda x: x.location.country.id)) AS country_id,
           software.id AS sid
    FROM catalogs
```

```
    WHERE catalog_type = 'Geoportal'
  )
  GROUP BY 1, 2
)
WHERE rn <= 3
ORDER BY country_id, n DESC;
```

Platforms with the share of APIs and inactive catalogs: ArcGIS Server 4,674, in-house development 1,043 and 11% with an API, GeoNode 11.2% inactive. Russian geoportals (286) are counted by the owner's country: add `AND owner.location.country.id = 'RU'`.

```
SELECT software.id AS sid, count(*) AS n,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api,
       round(100.0 * count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) /
count(*), 1) AS pct_dead
FROM catalogs
WHERE catalog_type = 'Geoportal'
GROUP BY 1
ORDER BY n DESC
LIMIT 40;
```

Cloud services from local vendors, excluding ArcGIS Hub and e-mapa: 132 services, about 5,300 geoportals, 21% with an API; for 104 services no API is recorded on any site:

```
WITH v AS (
  SELECT c.software.id AS sid, count(*) AS n, count(*) FILTER (WHERE c.api) AS with_api
  FROM catalogs c
  JOIN software s ON s.id = c.software.id
  WHERE c.catalog_type = 'Geoportal'
    AND s.subtype = 'managed_saas_service'
    AND c.software.id NOT IN ('arcgishub', 'emapa')
  GROUP BY 1
)
SELECT count(*) AS vendors, sum(n) AS sites,
       round(100.0 * sum(with_api) / sum(n), 0) AS pct_api,
       count(*) FILTER (WHERE with_api = 0) AS vendors_without_api
FROM v;
```

Catalog interfaces at geoportals: about 3,200 DCAT-US and DCAT-AP dumps at ArcGIS Hub sites, CSW at GeoNetwork and GeoNode, about 140 STAC servers:

```
SELECT c.software.id AS sid, e.type, count(DISTINCT c.id) AS n
FROM (SELECT id, software, unnest(endpoints) AS e FROM catalogs WHERE catalog_type =
'Geoportal') c
WHERE e.type IN ('dcatus11', 'dcatap201', 'ogcrecordsapi', 'csw202', 'stacserverapi',
'oaipmh20')
GROUP BY 1, 2
HAVING n > 20
ORDER BY 2, n DESC;
```

Research repositories

The chapter "Research Data Repositories". 8,404 in total, 119 inactive or decommissioned, 527 with restricted access:

```
SELECT count(*) AS n,
       count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) AS dead,
       count(*) FILTER (WHERE list_contains(access_mode, 'restricted')) AS restricted
FROM catalogs
WHERE catalog_type = 'Scientific data repository';
```

Owners: 85% universities and research organisations:

```
SELECT owner.type, count(*) AS n,
       round(100.0 * count(*) / sum(count(*)) OVER (), 0) AS pct
FROM catalogs
WHERE catalog_type = 'Scientific data repository'
GROUP BY 1
ORDER BY n DESC;
```

Countries: US 2,018, Germany 764, UK 503; international ones are recorded as `World`:

```
SELECT country_id, count(*) AS n
FROM (
  SELECT unnest(list_transform(coverage, lambda x: x.location.country.id)) AS country_id
  FROM catalogs
  WHERE catalog_type = 'Scientific data repository'
)
GROUP BY 1
ORDER BY n DESC
LIMIT 20;
```

Platforms with the share of APIs and inactive catalogs: in-house development 2,836 and 5% with an API, IPT 360 and 12% inactive:

```
SELECT software.id AS sid, count(*) AS n,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api,
       round(100.0 * count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) /
count(*), 1) AS pct_dead
FROM catalogs
WHERE catalog_type = 'Scientific data repository'
GROUP BY 1
ORDER BY n DESC
LIMIT 30;
```

Top platforms by country: this shows WEKO3 in Japan, DABAR in Croatia, OPUS in Germany:

```
SELECT country_id, sid, n
FROM (
  SELECT country_id, sid, count(*) AS n,
         row_number() OVER (PARTITION BY country_id ORDER BY count(*) DESC) AS rn
  FROM (
    SELECT unnest(list_transform(coverage, lambda x: x.location.country.id)) AS country_id,
           software.id AS sid
    FROM catalogs
    WHERE catalog_type = 'Scientific data repository'
  )
  GROUP BY 1, 2
)
WHERE rn <= 4 AND country_id IN ('US', 'DE', 'GB', 'JP', 'CN', 'HR', 'FR', 'RU', 'World')
ORDER BY country_id, n DESC;
```

Interfaces: OAI-PMH at 2,068, a sitemap at 2,029, DCAT at about 200:

```
SELECT e.type, count(DISTINCT id) AS n
FROM (
  SELECT id, unnest(endpoints) AS e
  FROM catalogs
  WHERE catalog_type = 'Scientific data repository'
)
GROUP BY 1
ORDER BY n DESC
LIMIT 25;
```

Topics: environment about 2,300, living organisms about 1,600. One repository produces several rows; topics come from two code lists:

```
SELECT t.type, t.name, count(DISTINCT id) AS n
FROM (
  SELECT id, unnest(topics) AS t
  FROM catalogs
  WHERE catalog_type = 'Scientific data repository'
)
GROUP BY 1, 2
ORDER BY n DESC
LIMIT 20;
```

Indicator catalogs

The chapter "Indicator Catalogs". Count, custom-built share, API share and inactive: 3,405, 44%, 37% and 29:

```
SELECT count(*) AS n,
       round(100.0 * count(*) FILTER (WHERE software.id = 'custom') / count(*), 0) AS
pct_custom,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api,
       count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) AS dead
FROM catalogs
WHERE catalog_type = 'Indicators catalog';
```

Owners: 46% central government; the custom-built and API shares within each group:

```
SELECT owner.type, count(*) AS n,
       round(100.0 * count(*) / sum(count(*)) OVER (), 0) AS pct,
       round(100.0 * count(*) FILTER (WHERE software.id = 'custom') / count(*), 0) AS
pct_custom,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api
FROM catalogs
WHERE catalog_type = 'Indicators catalog'
GROUP BY 1
ORDER BY n DESC;
```

Platforms with the API share: Swing 214, Knoema 125, PxWeb 110, DHIS2 74. For WordPress and Drupal the `api` field marks the CMS's own API, not the series:

```
SELECT software.id AS sid, count(*) AS n,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api,
```

```
       round(100.0 * count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) /
count(*), 1) AS pct_dead
FROM catalogs
WHERE catalog_type = 'Indicators catalog'
GROUP BY 1
ORDER BY n DESC
LIMIT 30;
```

Countries for each platform: this shows that Swing is the Netherlands and Belgium, HCI the US, Géoclip France, Knoema Africa:

```
WITH t AS (
  SELECT software.id AS sid,
         unnest(list_transform(coverage, lambda x: x.location.country.id)) AS country_id
  FROM catalogs
  WHERE catalog_type = 'Indicators catalog'
    AND software.id IN ('swing', 'knoema', 'pxweb', 'dhis2', 'opensdg', 'hci', 'geoclip',
'statsuite', 'sdmxri')
)
SELECT sid, string_agg(country_id || ':' || n, ', ' ORDER BY n DESC) AS countries
FROM (SELECT sid, country_id, count(*) AS n FROM t GROUP BY 1, 2)
GROUP BY 1;
```

Catalogs with SDMX endpoints: 170; DCAT endpoints appear at fewer than ten:

```
SELECT count(*) FILTER (WHERE list_has_any(list_transform(endpoints, lambda e: e.type),
         ['sdmx:datastructure', 'sdmx:dataflows', 'sdmx-rest', 'sdmx:codelist',
'sdmx:conceptscheme'])) AS with_sdmx
FROM catalogs
WHERE catalog_type = 'Indicators catalog';

SELECT e.type, count(*) AS n
FROM (SELECT unnest(endpoints) AS e FROM catalogs WHERE catalog_type = 'Indicators
catalog') t
WHERE lower(e.type) LIKE '%dcat%'
GROUP BY 1;
```

Search engines and aggregators

The chapter "Search Engines and Aggregators". Owners: 68% universities and research organisations. In the chapter, "Communities and civil society organisations" is `Community` plus `Civil society`, and "Other" is `Regional government` plus `Other`:

```
SELECT owner.type, count(*) AS n,
       round(100.0 * count(*) / sum(count(*)) OVER (), 0) AS pct
FROM catalogs
WHERE catalog_type = 'Data search engine'
GROUP BY 1
ORDER BY n DESC;
```

Territories: the EU as a whole 37, Italy 23, the whole world 21:

```
SELECT country_id, count(*) AS n
FROM (
  SELECT unnest(list_transform(coverage, lambda x: x.location.country.id)) AS country_id
  FROM catalogs
```

```
  WHERE catalog_type = 'Data search engine'
)
GROUP BY 1
ORDER BY n DESC
LIMIT 20;
```

Platforms with the share of APIs and inactive catalogs: OpenAIRE 50, in-house development 35 and 20% with an API, VIVO 28, Idra 6 and all inactive:

```
SELECT software.id AS sid, count(*) AS n,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api,
       count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) AS dead
FROM catalogs
WHERE catalog_type = 'Data search engine'
GROUP BY 1
ORDER BY n DESC;
```

The full list: this shows the 17 Italian UNIFIND instances on VIVO and the Idra demo instances:

```
SELECT software.id AS sid, owner.type AS owner_type, status, name, link
FROM catalogs
WHERE catalog_type = 'Data search engine'
ORDER BY sid, name;
```

Large search engines recorded under other types: Dateno, NIAID Discovery Portal, Research Data Australia, GESIS Search, DataONE, B2FIND, Earthdata Search, OmicsDI:

```
SELECT catalog_type, software.id AS sid, status, name, link
FROM catalogs
WHERE regexp_matches(lower(link),

'(dateno\.io|//data\.niaid|researchdata\.edu\.au|search\.gesis|//www\.dataone\.org|b2find|se
arch\.earthdata|omicsdi)')
ORDER BY catalog_type, name;
```

Metadata catalogs

The chapter "Metadata Catalogs and Registries". Count, custom-built share, API share, inactive and restricted access: 172, 38%, 67%, 0 and 18:

```
SELECT count(*) AS n,
       round(100.0 * count(*) FILTER (WHERE software.id = 'custom') / count(*), 0) AS
pct_custom,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api,
       count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) AS dead,
       count(*) FILTER (WHERE list_contains(access_mode, 'restricted')) AS restricted
FROM catalogs
WHERE catalog_type = 'Metadata catalog';
```

Owners: 31% universities and research organisations, 30% international organisations. In the chapter, `Community` and `Other` are combined into "Communities and other":

```
SELECT owner.type, count(*) AS n,
       round(100.0 * count(*) / sum(count(*)) OVER (), 0) AS pct
```

```
FROM catalogs
WHERE catalog_type = 'Metadata catalog'
GROUP BY 1
ORDER BY n DESC;
```

Platforms with the API share: in-house development 65 and 17%, FAIR Data Point 47, Metadata Browser (`mwmb`) 16, Skosmos 10:

```
SELECT software.id AS sid, count(*) AS n,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api
FROM catalogs
WHERE catalog_type = 'Metadata catalog'
GROUP BY 1
ORDER BY n DESC;
```

Platforms by country: this shows that the Netherlands is FAIR Data Points, the UK Metadata Browser, Australia Aristotle:

```
SELECT country_id, sid, count(*) AS n
FROM (
  SELECT unnest(list_transform(coverage, lambda x: x.location.country.id)) AS country_id,
         software.id AS sid
  FROM catalogs
  WHERE catalog_type = 'Metadata catalog'
)
GROUP BY 1, 2
ORDER BY country_id, n DESC;
```

Interfaces: DCAT in Turtle at 47 FAIR Data Points, the Skosmos REST API at 9 installations, SDMX endpoints at registries on Fusion Metadata Registry:

```
SELECT software.id AS sid, e.type, count(DISTINCT id) AS n
FROM (
  SELECT id, software, unnest(endpoints) AS e
  FROM catalogs
  WHERE catalog_type = 'Metadata catalog'
)
GROUP BY 1, 2
ORDER BY 1, n DESC;
```

ML catalogs

The chapter "ML Catalogs". 100 in total, 2 inactive, 5 with restricted access, 78% custom-built, 23% with an API:

```
SELECT count(*) AS n,
       count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) AS dead,
       count(*) FILTER (WHERE list_contains(access_mode, 'restricted')) AS restricted,
       round(100.0 * count(*) FILTER (WHERE software.id = 'custom') / count(*), 0) AS
pct_custom,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api
FROM catalogs
WHERE catalog_type = 'Machine learning catalog';
```

Owners: 37 universities and research organisations, 31 companies:

```
SELECT owner.type, count(*) AS n
FROM catalogs
WHERE catalog_type = 'Machine learning catalog'
GROUP BY 1
ORDER BY n DESC;
```

Countries: international 34, US 19, China 12, Russia 6:

```
SELECT country_id, count(*) AS n
FROM (
  SELECT unnest(list_transform(coverage, lambda x: x.location.country.id)) AS country_id
  FROM catalogs
  WHERE catalog_type = 'Machine learning catalog'
)
GROUP BY 1
ORDER BY n DESC
LIMIT 15;
```

Platforms and the API share: in-house development 78 and 13 with an API, Jekyll 9:

```
SELECT software.id AS sid, count(*) AS n,
       count(*) FILTER (WHERE api) AS with_api
FROM catalogs
WHERE catalog_type = 'Machine learning catalog'
GROUP BY 1
ORDER BY n DESC;
```

The full list of records. From it, the chapter manually sorts records into eight kinds (hubs, competition platforms, collections, benchmark sites and others); the registry has no such field:

```
SELECT name, link, software.id AS sid, owner.type AS owner_type, api, status
FROM catalogs
WHERE catalog_type = 'Machine learning catalog'
ORDER BY sid, name;
```

API catalogs, marketplaces and dataset lists

The chapter "API Catalogs, Marketplaces and Dataset Lists". Count, in-house development share, API share and restricted access share: 115, 74%, 51% and 14% for API catalogs; 56, 98%, 21% and 91% for marketplaces; 107, 89%, 14% and 1% for lists:

```
SELECT catalog_type, count(*) AS n,
       round(100.0 * count(*) FILTER (WHERE software.id = 'custom') / count(*), 0) AS
pct_custom,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api,
       round(100.0 * count(*) FILTER (WHERE list_contains(access_mode, 'restricted')) /
count(*), 0) AS pct_restricted
FROM catalogs
WHERE catalog_type IN ('API Catalog', 'Data marketplace', 'Datasets list')
GROUP BY 1;
```

Owners: 45% of API catalogs belong to central government, 70% of marketplaces to companies:

```
SELECT catalog_type, owner.type, count(*) AS n,
       round(100.0 * count(*) / sum(count(*)) OVER (PARTITION BY catalog_type), 0) AS pct
FROM catalogs
WHERE catalog_type IN ('API Catalog', 'Data marketplace', 'Datasets list')
GROUP BY 1, 2
ORDER BY 1, n DESC;
```

Countries: for marketplaces, South Korea 16, US 12, China 6:

```
SELECT catalog_type, country_id, count(*) AS n
FROM (
  SELECT catalog_type,
         unnest(list_transform(coverage, lambda x: x.location.country.id)) AS country_id
  FROM catalogs
  WHERE catalog_type IN ('API Catalog', 'Data marketplace', 'Datasets list')
)
GROUP BY 1, 2
QUALIFY row_number() OVER (PARTITION BY catalog_type ORDER BY count(*) DESC) <= 10
ORDER BY 1, n DESC;
```

Platforms: for API catalogs, Azure API Management 22, WSO2 API Manager 2:

```
SELECT catalog_type, software.id AS sid, count(*) AS n,
       count(*) FILTER (WHERE api) AS with_api
FROM catalogs
WHERE catalog_type IN ('API Catalog', 'Data marketplace', 'Datasets list')
GROUP BY 1, 2
ORDER BY 1, n DESC;
```

Transport developer portals, about 35 of 115, are selected by name and address; the list is worth reviewing by eye, because the pattern also catches extras (Metropolia, gov.br/conecta):

```
SELECT name, link
FROM catalogs
WHERE catalog_type = 'API Catalog'
  AND regexp_matches(lower(name || ' ' || link),

'transit|transport|metro|\bbus\b|rail|traffic|trafik|tmb|mta|bart|cta|marta|wmata|trimet|tra
nslink|entur|waltti|föli|foli|metlink|511|digitraffic|tdx|koroad|apiportal.ns|sptrans|oc
transpo|skåne|cumtd|cota|deutschebahn');
```

Dataset lists on GitHub and GitLab: 16; crypto exchange trading history: 13:

```
SELECT
  count(*) FILTER (WHERE regexp_matches(lower(name || ' ' || link),
'github.io|github.com|gitlab')) AS on_git,
  count(*) FILTER (WHERE regexp_matches(lower(name || ' ' || link),

'binance|bybit|okx|kucoin|mexc|bitget|kraken|gatedata|bitmex|crypto|polymarket|paradigm'))
AS crypto
FROM catalogs
WHERE catalog_type = 'Datasets list';
```

Sitemaps at API catalogs: 26 of 115:

```
SELECT catalog_type, e.type, count(DISTINCT id) AS n
```

```
FROM (
  SELECT id, catalog_type, unnest(endpoints) AS e
  FROM catalogs
  WHERE catalog_type IN ('API Catalog', 'Data marketplace', 'Datasets list')
)
GROUP BY 1, 2
ORDER BY 1, n DESC;
```

Capability map

The chapter "Capability Map", queries of 15 October 2026. Interfaces are taken from the `endpoints` list, DOIs from the `has_doi` flag; research repositories are counted together with generalist ones. The registry hardly records schema.org markup, so its share is not calculated.

Shares by type: DCAT at 26% of open data portals, OAI-PMH at 25% and DOIs at 17% of research repositories, CSW or OGC API Records at 20% of geoportals, SDMX at 5.2% of indicator catalogs:

```
WITH t AS (
  SELECT CASE WHEN catalog_type IN ('Scientific data repository', 'General research
repository')
              THEN 'Science' ELSE catalog_type END AS ct,
         properties.has_doi AS doi,
         list_transform(coalesce(endpoints, []), lambda e: lower(e.type)) AS et
  FROM catalogs
)
SELECT ct, count(*) AS n,
       round(100.0 * count(*) FILTER (WHERE len(list_filter(et, lambda x: x LIKE 'dcat%' OR
x LIKE '%:dcatus11')) > 0) / count(*), 1) AS pct_dcat,
       round(100.0 * count(*) FILTER (WHERE len(list_filter(et, lambda x: x LIKE 'oai%')) >
0) / count(*), 1) AS pct_oai,
       round(100.0 * count(*) FILTER (WHERE len(list_filter(et, lambda x: x LIKE 'csw%' OR
x LIKE 'ogcrecords%')) > 0) / count(*), 1) AS pct_csw_records,
       round(100.0 * count(*) FILTER (WHERE len(list_filter(et, lambda x: x LIKE '%sdmx%'))
> 0) / count(*), 1) AS pct_sdmx,
       round(100.0 * count(*) FILTER (WHERE doi) / count(*), 1) AS pct_doi
FROM t
WHERE ct IN ('Open data portal', 'Science', 'Geoportal', 'Indicators catalog')
GROUP BY 1
ORDER BY n DESC;
```

DCAT at open data portals by platform: Socrata 239 of 245, OpenDataSoft 348 of 454, DKAN 103 of 161, CKAN 633 of 1,906, in-house development 24 of 1,144:

```
SELECT software.id AS sid, count(*) AS n,
       count(*) FILTER (WHERE len(list_filter(
         list_transform(coalesce(endpoints, []), lambda e: lower(e.type)),
         lambda x: x LIKE 'dcat%' OR x LIKE '%:dcatus11')) > 0) AS with_dcat
FROM catalogs
WHERE catalog_type = 'Open data portal'
GROUP BY 1
ORDER BY n DESC
LIMIT 10;
```

DOIs and OAI-PMH at research repositories by platform: Figshare 266 DOIs of 288, Dataverse 172 DOIs and 177 OAI-PMH of 257, DSpace 218 DOIs and 665 OAI-PMH of 829, in-house development 158 DOIs and 19 OAI-PMH of 2,840 (research repositories together with four general-purpose

ones; research repositories alone number 2,836, of which 157 have DOIs):

```
SELECT software.id AS sid, count(*) AS n,
       count(*) FILTER (WHERE properties.has_doi) AS with_doi,
       count(*) FILTER (WHERE len(list_filter(
         list_transform(coalesce(endpoints, []), lambda e: lower(e.type)),
         lambda x: x LIKE 'oai%')) > 0) AS with_oai
FROM catalogs
WHERE catalog_type IN ('Scientific data repository', 'General research repository')
GROUP BY 1
ORDER BY n DESC
LIMIT 10;
```

A catalog service at geoportals by platform: ArcGIS Hub 3,257 of 3,464, GeoNetwork 686 of 762, GeoNode 628 of 770, ArcGIS Server 1 of 4,674, e-mapa 0 of 1,927, in-house development 4 of 1,043:

```
SELECT software.id AS sid, count(*) AS n,
       count(*) FILTER (WHERE len(list_filter(
         list_transform(coalesce(endpoints, []), lambda e: lower(e.type)),
         lambda x: x LIKE 'csw%' OR x LIKE 'ogcrecords%')) > 0) AS with_catalog_service
FROM catalogs
WHERE catalog_type = 'Geoportal'
GROUP BY 1
ORDER BY n DESC
LIMIT 10;
```

SDMX at indicator catalogs by platform: of 177 catalogs, 113 on Knoema:

```
SELECT software.id AS sid, count(*) AS n
FROM catalogs
WHERE catalog_type = 'Indicators catalog'
  AND len(list_filter(
        list_transform(coalesce(endpoints, []), lambda e: lower(e.type)),
        lambda x: x LIKE '%sdmx%')) > 0
GROUP BY 1
ORDER BY n DESC;
```

Metadata export

The chapter "Metadata Export", queries of 15 October 2026. At least one standard way to serve descriptions (DCAT, OAI-PMH, CSW, OGC API Records, SDMX or STAC) at 9,330 of 43,614 catalogs. A DCAT dump at 5,233, both profiles, DCAT-US and DCAT-AP, at 3,152. The registry hardly records schema.org markup: there are 38 such endpoints:

```
WITH f AS (
  SELECT list_transform(coalesce(endpoints, []), lambda e: lower(e.type)) AS et
  FROM catalogs
)
SELECT count(*) AS n,
       count(*) FILTER (WHERE len(list_filter(et, lambda x: x LIKE 'dcat%' OR x LIKE
'%dcatus11'
                        OR x = 'dcat-us')) > 0) AS dcat,
       count(*) FILTER (WHERE len(list_filter(et, lambda x: x LIKE 'dcatus11' OR x LIKE
'%:dcatus11'
```

```
                        OR x = 'dcat-us')) > 0
                        AND len(list_filter(et, lambda x: x LIKE 'dcatap%')) > 0) AS
both_profiles,
       count(*) FILTER (WHERE len(list_filter(et, lambda x: x LIKE 'dcat%' OR x LIKE
'%dcatus11'
                        OR x = 'dcat-us' OR x LIKE 'oai%' OR x LIKE 'csw%' OR x LIKE
'ogcrecords%'
                        OR x LIKE '%sdmx%' OR x LIKE 'stac%')) > 0) AS any_standard,
       count(*) FILTER (WHERE len(list_filter(et, lambda x: x LIKE 'schemaorg%')) > 0) AS
schemaorg
FROM f;
```

DCAT at open data portals by platform: Socrata 239 of 245, OpenDataSoft 348 of 454, DKAN 103 of 161, CKAN 633 of 1,906, in-house development 24 of 1,144. The query is in the section "Capability map".

API and bulk download

The chapter "API and Bulk Download", queries of 15 October 2026. The "has a programmatic entry point" flag in the registry coincides with a non-empty `endpoints` list: 28,471 of 43,614. A standard way to take all descriptions (DCAT, OAI-PMH, CSW or OGC API Records) at 8,894 catalogs. Shares by type: 70% of geoportals, 69% of open data portals, 63% of research repositories, 37% of indicator catalogs, 23% of ML catalogs:

```
WITH t AS (
  SELECT catalog_type,
         list_transform(coalesce(endpoints, []), lambda e: lower(e.type)) AS et
  FROM catalogs
), f AS (
  SELECT catalog_type,
         len(et) > 0 AS any_ep,
         len(list_filter(et, lambda x: x LIKE 'dcat%' OR x LIKE '%dcatus11' OR x LIKE
'oai%'
                                    OR x LIKE 'csw%' OR x LIKE 'ogcrecords%')) > 0 AS
all_desc,
         len(list_filter(et, lambda x: x LIKE '%sitemap%')) > 0 AS sitemap
  FROM t
)
SELECT coalesce(catalog_type, 'total') AS catalog_type, count(*) AS n,
       count(*) FILTER (WHERE any_ep) AS any_ep,
       round(100.0 * count(*) FILTER (WHERE any_ep) / count(*), 0) AS pct_any,
       count(*) FILTER (WHERE all_desc) AS all_desc,
       round(100.0 * count(*) FILTER (WHERE all_desc) / count(*), 0) AS pct_all_desc,
       round(100.0 * count(*) FILTER (WHERE sitemap) / count(*), 0) AS pct_sitemap
FROM f
GROUP BY ROLLUP (catalog_type)
ORDER BY n DESC;
```

In-house development versus ready-made platforms: 697 of 7,034 with a programmatic entry point and 58 with a standard description dump for in-house builds, 76% with a programmatic entry point for the rest:

```
WITH t AS (
  SELECT software.id = 'custom' AS is_custom,
         list_transform(coalesce(endpoints, []), lambda e: lower(e.type)) AS et
  FROM catalogs
```

```
)
SELECT is_custom, count(*) AS n,
       count(*) FILTER (WHERE len(et) > 0) AS any_ep,
       round(100.0 * count(*) FILTER (WHERE len(et) > 0) / count(*), 0) AS pct_any,
       count(*) FILTER (WHERE len(list_filter(et, lambda x: x LIKE 'dcat%' OR x LIKE
'%dcatus11'
                        OR x LIKE 'oai%' OR x LIKE 'csw%' OR x LIKE 'ogcrecords%')) > 0) AS
all_desc
FROM t
GROUP BY 1;
```

The registry records RSS and Atom feeds mainly at geoportals and research repositories; CKAN portals hardly have them recorded, although a feed is in the CKAN core. So the chapter does not give a registry-based share of feeds.

Identifiers and DOIs

The chapter "Persistent Identifiers, DOIs and Versions", queries of 15 October 2026. A DOI flag at 2,008 of 43,614 catalogs: 1,396 research repositories and 8 generalist ones, 584 geoportals, the rest isolated cases:

```
SELECT catalog_type,
       count(*) FILTER (WHERE properties.has_doi) AS with_doi,
       count(*) AS n
FROM catalogs
GROUP BY 1
HAVING with_doi > 0
ORDER BY with_doi DESC;
```

The flag is mostly inherited from the platform record (`software.pid_support.has_doi`), so it should be read as "the platform can". GeoNetwork's platform is marked "Yes", and 555 of 764 installations got the flag; in-house development is marked "No", and 175 flags were set manually:

```
SELECT c.software.id AS sid,
       s.pid_support.has_doi AS platform_flag,
       count(*) FILTER (WHERE c.properties.has_doi) AS with_doi,
       count(*) AS n
FROM catalogs c
LEFT JOIN software s ON s.id = c.software.id
GROUP BY 1, 2
HAVING with_doi > 0
ORDER BY with_doi DESC
LIMIT 10;
```

The sample for the Handle check: 120 of 782 OAI-PMH endpoints at active DSpace installations, in a reproducible random order. For each, the Handle of the first record from `ListIdentifiers` is taken and checked via `https://hdl.handle.net/api/handles/…`:

```
SELECT e.url
FROM (SELECT unnest(endpoints) AS e
      FROM catalogs
      WHERE software.id = 'dspace' AND status = 'active')
WHERE lower(e.type) LIKE 'oai%'
ORDER BY hash(e.url)
```

```
LIMIT 120;
```

Access, roles and workflows

The chapter "Access, Roles and Publishing Workflows", queries of 15 October 2026. Restricted access to at least part of the data at 1,131 of 43,614 catalogs: 537 restricted only, 594 together with open:

```
SELECT count(*) AS total,
       count(*) FILTER (WHERE list_contains(access_mode, 'restricted')) AS restricted,
       count(*) FILTER (WHERE access_mode = ['restricted']) AS only_restricted,
       count(*) FILTER (WHERE list_contains(access_mode, 'restricted')
                          AND list_contains(access_mode, 'open')) AS mixed
FROM catalogs;
```

The share by platform: DHIS2 97%, NADA 72%, Pure 36% against CKAN 0.6% and Dataverse 1.9%:

```
SELECT software.id AS platform,
       count(*) FILTER (WHERE list_contains(access_mode, 'restricted')) AS restricted,
       count(*) AS n,
       round(100.0 * restricted / n, 1) AS pct
FROM catalogs
WHERE software.id IN ('nada', 'dhis2', 'gen3', 'redivis', 'pure', 'ckan', 'dataverse',
                      'dspace', 'inveniordm', 'geonetwork', 'opendatasoft', 'socrata',
'arcgishub')
GROUP BY 1
ORDER BY pct DESC;
```

The list for measuring registration on CKAN portals: 1,839 active catalogs. For each, the API address is taken from the `ckan` endpoint, or, if there is none, the site address with `/api/3`:

```
SELECT id, link, catalog_type, owner.type AS otype,
       coverage[1].location.country.id AS country,
       [x.url for x in endpoints if x.type = 'ckan'][1] AS api
FROM catalogs
WHERE software.id = 'ckan' AND status = 'active'
ORDER BY id;
```

Microdata catalogs

The chapter "Microdata Catalogs". 326 in total, 59 not working, 159 with restricted access, 13% custom-built, 70% with an API:

```
SELECT count(*) AS n,
       count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) AS dead,
       count(*) FILTER (WHERE list_contains(access_mode, 'restricted')) AS restricted,
       round(100.0 * count(*) FILTER (WHERE software.id = 'custom') / count(*), 0) AS
pct_custom,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api
FROM catalogs
WHERE catalog_type = 'Microdata catalog';
```

Owners and the NADA share for each: 55% central government, 59% of them on NADA; international organisations 90% on NADA:

```
SELECT owner.type, count(*) AS n,
       round(100.0 * count(*) / sum(count(*)) OVER (), 0) AS pct,
       round(100.0 * count(*) FILTER (WHERE software.id = 'nada') / count(*), 0) AS
pct_nada
FROM catalogs
WHERE catalog_type = 'Microdata catalog'
GROUP BY 1
ORDER BY n DESC;
```

Platforms with the API share and number not working: NADA 158 and 47 not working, in-house development 43 and 12% with an API:

```
SELECT software.id AS sid, count(*) AS n,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api,
       count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) AS dead,
       count(*) FILTER (WHERE list_contains(access_mode, 'restricted')) AS restricted
FROM catalogs
WHERE catalog_type = 'Microdata catalog'
GROUP BY 1
ORDER BY n DESC;
```

Countries: 139 countries and 27 worldwide catalogs; US 18, Canada 12, Germany 11:

```
SELECT country_id, count(*) AS n
FROM (
  SELECT unnest(list_transform(coverage, lambda x: x.location.country.id)) AS country_id
  FROM catalogs
  WHERE catalog_type = 'Microdata catalog'
)
GROUP BY 1
ORDER BY n DESC;
```

Non-working catalogs without HTTPS and on IP addresses: 38 of 58 without HTTPS, 7 of 10 on IP addresses:

```
SELECT count(*) FILTER (WHERE link LIKE 'http://%') AS http,
       count(*) FILTER (WHERE link LIKE 'http://%' AND status = 'inactive') AS http_dead,
       count(*) FILTER (WHERE regexp_matches(link, '^https?://[0-9.]+')) AS ip,
       count(*) FILTER (WHERE regexp_matches(link, '^https?://[0-9.]+') AND status =
'inactive') AS ip_dead
FROM catalogs
WHERE catalog_type = 'Microdata catalog';
```

Countries with a non-working NADA and no other active microdata catalog: 18 of 38:

```
WITH c AS (
  SELECT status, software.id AS sid,
         unnest(list_transform(coverage, lambda x: x.location.country.id)) AS cc
  FROM catalogs
  WHERE catalog_type = 'Microdata catalog'
),
dead AS (SELECT DISTINCT cc FROM c WHERE status = 'inactive' AND sid = 'nada' AND
length(cc) = 2)
```

```
SELECT count(*) AS countries,
       count(*) FILTER (WHERE NOT EXISTS (
         SELECT 1 FROM c WHERE c.cc = dead.cc AND c.status = 'active')) AS without_active
FROM dead;
```

Comparing types and hybrids

The chapter "Comparing Types and Hybrids". The summary table: count and the shares of in-house development, API, non-working and restricted access by type:

```
SELECT catalog_type, count(*) AS n,
       round(100.0 * count(*) FILTER (WHERE software.id = 'custom') / count(*), 0) AS
pct_custom,
       round(100.0 * count(*) FILTER (WHERE api) / count(*), 0) AS pct_api,
       round(100.0 * count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) /
count(*), 1) AS pct_dead,
       round(100.0 * count(*) FILTER (WHERE list_contains(access_mode, 'restricted')) /
count(*), 0) AS pct_restricted
FROM catalogs
GROUP BY 1
ORDER BY n DESC;
```

The main owner and the top four platforms of each type, excluding in-house development:

```
SELECT catalog_type, owner.type, count(*) AS n,
       round(100.0 * count(*) / sum(count(*)) OVER (PARTITION BY catalog_type), 0) AS pct
FROM catalogs
GROUP BY 1, 2
QUALIFY row_number() OVER (PARTITION BY catalog_type ORDER BY count(*) DESC) = 1;

SELECT catalog_type, string_agg(sid || ' ' || n, ', ' ORDER BY n DESC) AS top
FROM (
  SELECT catalog_type, software.id AS sid, count(*) AS n
  FROM catalogs
  WHERE software.id <> 'custom'
  GROUP BY 1, 2
  QUALIFY row_number() OVER (PARTITION BY catalog_type ORDER BY count(*) DESC) <= 4
)
GROUP BY 1;
```

In how many types a platform appears: WordPress in nine, CKAN in five, GeoNode, Dataverse and PxWeb in one:

```
SELECT software.id AS sid, count(DISTINCT catalog_type) AS n_types, count(*) AS n,
       string_agg(DISTINCT catalog_type, ', ') AS types
FROM catalogs
WHERE catalog_type NOT IN ('Other', 'General research repository')
  AND software.id <> 'custom'
GROUP BY 1
ORDER BY n_types DESC, n DESC
LIMIT 30;
```

Owners with catalogs of several types: 2,203 of 32,595, of which 254 with three or more, accounting for 7,358 catalogs; 118 make do with one program. An owner here is an organisation name within one country, so the count is approximate:

```
CREATE TEMP TABLE o AS
SELECT lower(trim(owner.name)) AS oname, owner.location.country.id AS cc,
       owner.type AS otype, catalog_type, software.id AS sid
FROM catalogs
WHERE coalesce(trim(owner.name), '') <> ''
  AND catalog_type NOT IN ('Other', 'General research repository');

CREATE TEMP TABLE g AS
SELECT oname, cc, any_value(otype) AS otype,
       count(DISTINCT catalog_type) AS nt, count(*) AS n, count(DISTINCT sid) AS ns,
       string_agg(DISTINCT catalog_type, ' | ' ORDER BY catalog_type) AS type_list
FROM o
GROUP BY 1, 2;

SELECT count(*) AS owners,
       count(*) FILTER (WHERE nt > 1) AS multi,
       count(*) FILTER (WHERE nt > 2) AS multi3,
       sum(n) FILTER (WHERE nt > 1) AS catalogs,
       count(*) FILTER (WHERE nt > 1 AND ns = 1) AS one_software
FROM g;
```

The share of such owners by owner type, and the most frequent pairs of types: geoportal and open data portal at 953:

```
SELECT otype, count(*) AS owners,
       round(100.0 * count(*) FILTER (WHERE nt > 1) / count(*), 0) AS pct_multi
FROM g
GROUP BY 1
ORDER BY owners DESC;

SELECT type_list, count(*) AS owners
FROM g
WHERE nt = 2
GROUP BY 1
ORDER BY owners DESC
LIMIT 15;
```

The "geoportal and open data portal" pair: 1,120 owners, combinations of programs (ArcGIS Server and CKAN: 101), and one program on both sides at 44 of 953, of which 39 are in-house builds:

```
CREATE TEMP TABLE pair AS
SELECT oname, cc, any_value(otype) AS otype
FROM o
GROUP BY 1, 2
HAVING bool_or(catalog_type = 'Geoportal') AND bool_or(catalog_type = 'Open data portal');

SELECT g.sid AS geo, d.sid AS odp, count(DISTINCT oname || cc) AS owners
FROM (SELECT DISTINCT oname, cc, sid FROM o JOIN pair USING (oname, cc) WHERE catalog_type
= 'Geoportal') g
JOIN (SELECT DISTINCT oname, cc, sid FROM o JOIN pair USING (oname, cc) WHERE catalog_type
= 'Open data portal') d
  USING (oname, cc)
GROUP BY 1, 2
ORDER BY owners DESC
LIMIT 12;

SELECT shared_sid, count(*) AS owners
FROM (
  SELECT oname, cc,
```

```
          unnest(list_intersect(
            list(DISTINCT sid) FILTER (WHERE catalog_type = 'Geoportal'),
            list(DISTINCT sid) FILTER (WHERE catalog_type = 'Open data portal'))) AS
shared_sid
  FROM o JOIN g USING (oname, cc)
  WHERE g.nt = 2 AND g.type_list = 'Geoportal | Open data portal'
  GROUP BY 1, 2
)
GROUP BY 1
ORDER BY owners DESC;
```

Catalogs of one organisation: this shows the line-ups of the Pacific Community, the World Bank, FAO and INEGI:

```
SELECT owner.name, catalog_type, software.id AS sid, status, link
FROM catalogs
WHERE owner.name IN ('Pacific Community (SPC)', 'World Bank',
                     'Food and Agriculture Organization of the United Nations',
                     'Instituto Nacional de Estadística y Geografía (INEGI)')
ORDER BY 1, 2, 3;
```

When you don't need a portal

The chapter "When You Don't Need a Portal", queries of 15 October 2026 against the same export. Dataset lists: 107, a programmatic interface at 15, in-house development at 95, inactive and withdrawn: 7:

```
SELECT
  count(*) AS n,
  count(*) FILTER (WHERE api) AS with_api,
  count(*) FILTER (WHERE software.id = 'custom') AS custom,
  count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) AS dead
FROM catalogs
WHERE catalog_type = 'Datasets list';
```

A sitemap at 12. The other page engines: Drupal, WordPress, TYPO3, Jekyll, SharePoint, Liferay:

```
SELECT count(DISTINCT id) AS with_sitemap
FROM (
  SELECT id, unnest(endpoints) AS e
  FROM catalogs
  WHERE catalog_type = 'Datasets list'
)
WHERE e.type = 'sitemap';

SELECT software.id AS sid, count(*) AS n
FROM catalogs
WHERE catalog_type = 'Datasets list'
GROUP BY 1
ORDER BY n DESC;
```

Innovative catalogs

The chapter "What Innovative Catalogs Can Already Do", queries of 15 October 2026 against the same export. ERDDAP: 175 records, one inactive. Datasette: 45, one inactive. Redivis: 13, DataLad: 10, DANDI: 3. Datadex: 3, none with a programmatic interface recorded:

```
SELECT software.id AS sid, count(*) AS n,
       count(*) FILTER (WHERE status IN ('inactive', 'deprecated')) AS dead,
       count(*) FILTER (WHERE api) AS with_api
FROM catalogs
WHERE software.id IN ('erddap', 'datasette', 'redivis', 'datalad', 'dandi', 'datadex')
GROUP BY 1
ORDER BY n DESC;
```

Records with openEO in the endpoint address or type: 12, of which 3 at Copernicus Data Space Ecosystem. Catalogs with a GA4GH DRS interface: 27:

```
SELECT count(DISTINCT id) AS with_openeo,
       count(DISTINCT id) FILTER (WHERE e.url ILIKE '%dataspace.copernicus.eu%') AS cdse
FROM (SELECT id, unnest(endpoints) AS e FROM catalogs)
WHERE e.url ILIKE '%openeo%' OR e.type ILIKE 'openeo%';

SELECT count(DISTINCT id) AS with_drs
FROM (SELECT id, unnest(endpoints) AS e FROM catalogs)
WHERE e.type = 'ga4gh:drs';
```

The openEO count also includes two community sites with only a sitemap: the address contains the word, but there is no openEO server behind it. Records for checking against the chapter: Source Cooperative, Data Commons, Our World in Data, DoltHub:

```
SELECT id, catalog_type, software.id AS sid, api, link
FROM catalogs
WHERE link ILIKE '%source.coop%' OR link ILIKE '%datacommons.org%'
   OR link ILIKE '%ourworldindata.org%' OR link ILIKE '%dolthub.com%';
```

That day, only a sitemap was recorded among the endpoints for DoltHub and Our World in Data, although both have APIs. Data Commons has no endpoints at all and no programmatic interface recorded, although it has REST, SDMX and an MCP server. The main Source Cooperative record has a single STAC interface; the `beta.source.coop` record has no endpoints.

Further reading

- Query examples in the registry documentation
- Data model

Platform Comparison Tables

How to read the cells. Each cell has one of four marks. Text in brackets is not a fifth mark but a module name or a caveat.

- **core**: the capability is in the platform itself;
- **extension**: a separate module is needed; who maintains it and which core version it works with is written in the profile;
- **no**: the platform does not do this;
- **not checked**: the cell could not be filled from the October 2026 documentation. Check it yourself before choosing the platform.

Cloud services do not accept third-party extensions, so they have no "extension" cell. The "own extensions" row answers a different question: whether a module can be added at all. Details are in the profiles. The same four marks are used in the chapters on capabilities. The words "customisation" and "impossible" appear only in the shortlist comparison: they mean "we will have to write it ourselves" and "the path is closed on this platform".

Open data and lightweight catalogs

In the registry, Huwise is recorded under its old name, OpenDataSoft. Here the new name is used, with the old one in brackets. Harvesting for Huwise and Socrata was checked against the vendors' documentation on 14 October 2026. The other cells are from the chapters on capabilities.

Capability	CKAN	DKAN	Huwise (OpenDataSoft)	Socrata	ArcGIS Hub	JKAN
Search API	core	core	core	core	core	no
Dump of all descriptions	extension (ckanext-dcat)	core	core	core	core	core (data.json file)
Change feed and changed-since query	core (Atom feed, filter by modification date)	not checked	core (RSS feed, filter by date)	not checked	core (RSS feed)	no (the repository's change feed, if the site is in Git)
DCAT	extension (ckanext-dcat, DCAT-AP 3 profile by default)	core (data.json file; the core schema is DCAT-US 1.1)	core (RDF, Turtle and DCAT; a national profile is not always available)	core (data.json file per DCAT-US)	core (DCAT-US 1.1 and DCAT-AP 2.1.1; DCAT-US 3.0 and DCAT-AP 3.0.0 are enabled separately)	core (the data.json file declares DCAT-US 1.1; the template fills a short list of fields)
schema.org	extension (structured data plugin in ckanext-dcat)	not checked	core, enabled by the administrator	core (JSON-LD on the dataset page)	no	no (the page template adds no markup)
Files hosted locally	core	core	core	core	core	core (files next to the site)

Capability	CKAN	DKAN	Huwise (OpenDataSoft)	Socrata	ArcGIS Hub	JKAN
Cloud file storage	extension (in 2.12, ckanext-file-keeper-cloud; earlier, ckanext-s3filestore)	not checked	core (stored by the vendor)	core (stored by the vendor)	core (stored by the vendor)	no (files next to the site or by link)
External link checking	extension (ckanext-archiver, also keeps copies)	not checked	not checked	not checked	not checked	no
Table API	core; CSV loading via the XLoader extension	core	core	core	core (ArcGIS services)	no
Custom fields	extension (ckanext-scheming)	core (replacing the JSON Schema file)	core (field templates)	core (fieldsets)	no (one style from ready standards per organisation)	no (template editing)
Table data dictionary	core (for tables in DataStore)	not checked	core	core	core (layer field descriptions)	no
DOI	extension (ckanext-doi; not tested on CKAN 2.11 and 2.12)	no	no	no	no	no
Citation	extension (ckanext-doi only with a DOI; ckanext-citeproc)	no	no	no	no	no
Roles	core (organisations, three roles)	core (Drupal permissions)	service core	service core (five portal roles, three dataset roles)	service core	no
Review before publication	draft: core; curator review: extension	core (states); by default a dataset is published immediately	no (there is a separate permission to publish)	core (automatic or manual approval)	no	no
Single sign-on	extension (SAML, OpenID Connect, LDAP)	not checked	core (SAML, OpenID Connect)	core (OpenID Connect; SAML through support)	core (SAML, OpenID Connect)	no
Harvesting other catalogs	extension (ckanext-harvest; RDF harvesting via ckanext-dcat, geocatalogs via ckanext-spatial)	not checked	core (CKAN, Socrata, ArcGIS, ArcGIS Hub, data.json, data.gouv.fr and other Huwise portals; CSW enabled on request to support)	core (Catalog Connector: data.json and public ArcGIS Server; exchange between Data & Insights domains)	not checked	no
Own extensions	yes	yes, through Drupal	no	no	no	template editing

Science and geodata

Capability	Dataverse	InvenioRDM	DSpace	GeoNetwork	NADA	PxWeb
API	core	core	core	core	core	core

Capability	Dataverse	InvenioRDM	DSpace	GeoNetwork	NADA	PxWeb
Description dump	core (OAI-PMH)	core (OAI-PMH)	core (OAI-PMH)	core (CSW and OAI-PMH)	core (own API and DDI Codebook; no OAI-PMH)	core (table API, not a file catalog)
Changed-since query	core (OAI-PMH by date)	core (OAI-PMH by date)	core (OAI-PMH by date, OpenSearch feeds)	core (OAI-PMH by date)	not checked	core (PxWebApi 2: tables changed in the last N days)
DCAT	not in core (third-party DCAT 3 exporter)	core	no	core	no	no
schema.org	core (since version 6.10 a brief Croissant markup is embedded in the page)	core	no	core	core (JSON-LD)	no
Files hosted locally	core	core	core	core (more often a link to a service)	core	no
Cloud file storage	core (S3, Swift, Globus; several stores)	core (S3 since v13)	core (S3; several stores)	core (S3, JCloud and CMIS modules in the distribution)	not checked	no
DOI	core	core	core	core (must be enabled and connected to DataCite)	only a field for a DOI issued separately	no
Citation	core	core	no (only meta tags for Google Scholar; the port from DSpace-CRIS is not finished)	core (HTML, text, RIS, BibTeX)	core via DataCite, if there is a DOI	no
Custom fields	core	core (configuration)	core (configuration)	core (ISO 19139, ISO 19115-3, Dublin Core); a national profile is an extension (schema plugin)	not checked	no
Table data dictionary	core (variables of uploaded tables)	no	no	core (ISO 19110 feature catalogue)	core (DDI variables)	core (table variables and values)
Restricted file access	core	core	core	core (a separate permission to download files uploaded to GeoNetwork)	core	no
Roles	core	core	core	core	core	not checked

Capability	Dataverse	InvenioRDM	DSpace	GeoNetwork	NADA	PxWeb
Review before publication	core	core	core	core (by default approval can be bypassed)	not checked	not checked
Single sign-on	core (SAML, OpenID Connect, ORCID)	core (ORCID, OpenID Connect, SAML)	core	core	not checked	not checked
Harvesting other catalogs	core (OAI-PMH client; the record links to the original)	not checked	not checked	core (CSW, OAI-PMH, OGC services, STAC)	not checked	not checked
Own extensions	external tools and previewers	platform modules	extensions	schema plugins; OGC API Records is an extension	few	few

udata, Piveau and GeoNode

These three platforms are not in the tables above, but they have already been checked in the chapters on capabilities. A "not checked" cell here means that those chapters had no answer.

Capability	udata	Piveau	GeoNode 5.1
DCAT	core: paginated and as a file per dataset	core: several formats and SPARQL	no: the catalog serves an API and CSW, not DCAT
schema.org	no: the page HTML has no markup	no: the dataset page on data.europa.eu is rendered by a script	no
Catalog protocol	core (DCAT); a nightly dump of the whole catalog to CSV is a core task	core (DCAT and SPARQL)	core (CSW and ISO 19115, via pycsw inside)
Roles	core: administrator, editor, partial editor	not checked	not checked
Review before publication	draft: core; no curator review	not checked	not checked
Harvesting other catalogs	core (DCAT, CSW, CKAN, DKAN and others; disabled by default; the search index is rebuilt after harvesting)	core	not checked

What the table does not replace

A "core" cell does not mean "switched on with a tick in an hour". A DOI in GeoNetwork and Dataverse requires a contract with DataCite. An "extension" cell does not mean "do not take it": ckanext-dcat is an ordinary part of a mature CKAN portal, not something exotic. A "no" cell means this function will have to be covered by another platform or in-house development.

Sources

- Platform profiles
- ckanext-dcat, Dataverse features

- Harvesting: Huwise harvesters, checked 14 October 2026; Socrata Catalog Connector. Datasets connected to Socrata through `data.json` become public after synchronisation. Dataverse's OAI-PMH client and udata's six harvesters: the chapter on other capabilities, checked 14 October 2026
- Roles, review before publication and single sign-on: the chapter on access and roles, checked 12 October 2026
- Software records in the dataportals-registry, release 1.27.0 of 15 October 2026. Notes such as "plugin only" were used as a hint of what to check in the documentation, not as the sole source.

Checklists

Short lists live in the chapters, next to the explanation. Here they are collected in working form: they can be printed or copied into a task. Each item links to the chapter that explains why it is needed.

Two long lists stay in their chapters, because they cannot be filled in without explanation:

- Catalog requirements and platform comparison: what to put in a specification and what of it is not mandatory for everyone.
- Self-audit: how to assess your catalog in one pass and what to record in the result.

Checking a live portal

This list works both for your own catalog and for someone else's that you are considering as a model or as a source for harvesting. It takes about half an hour.

Findability and dump

- [ ] The dataset page has an `application/ld+json` block with the Dataset type: metadata export.
- [ ] The dump declares the profile the harvester expects (DCAT-AP 3.0.1, DCAT-US 3.0), rather than merely opening at a `data.json` address.
- [ ] A `sitemap.xml` sitemap responds: API and bulk download.
- [ ] There is a file with all descriptions (`data.json`, `catalog.rdf`, `catalog.ttl`), and it opens without a login.
- [ ] The number of datasets in the dump matches the number in the site's search.
- [ ] The dump passes the validator for the relevant profile: DCAT-AP, INSPIRE or DataCite: quality metrics.
- [ ] For a geoportal, CSW or OGC API Records responds; for a research repository, OAI-PMH.
- [ ] A paginated dump reaches the last page, and API search serves the next page: API and bulk download.
- [ ] There is a "modified since date" query or a change feed, and something from the last few days is in it.
- [ ] Descriptions and the API respond from the command line, without a login, a key or a bot check, preferably from another network.
- [ ] Large databases have a bulk download as files with a date and size, not only an API with limits.
- [ ] Limits, terms of use and API documentation are published; the API has a contact.

Descriptions

- [ ] Ten random datasets, not the newest, have the licence, publisher, update date, period and territory filled in: metadata depth.

- [ ] The most frequent value of period, territory and update frequency is not a placeholder such as "1900–2099" or "the whole country" across datasets from different organisations: checking on a sample.
- [ ] The description does not repeat the title and is longer than 50 characters: Google's markup will not accept a shorter one.
- [ ] The licence is looked for where the schema keeps it: in DCAT-AP, on the distributions rather than the dataset.
- [ ] Theme, licence, language and territory come from a shared code list rather than free text: metadata standards.
- [ ] Tabular datasets have a data dictionary, and the column types match the content.
- [ ] Custom fields and schema plugins (ckanext-scheming, ckanext-validation, GeoNetwork profiles) have been checked against the repository and the branch for the current core version.

Content

- [ ] Most datasets are not one- or two-line notes, and the total data volume is comparable with the main sources on the topic: substantive check.
- [ ] "Updated" datasets really changed: the file has new periods, not just a new date in the record.
- [ ] For CSV, the encoding and delimiter are stated or unambiguous: anti-patterns.

Files and addresses

- [ ] "Download" links open. Count the share of external links that return an error, and group the errors by domain: internal storage.
- [ ] A link labelled CSV, XLSX or PDF opens a file, not a web page or the site's home page.
- [ ] External links are checked on a schedule, and a resource shows the date and result of the last check.
- [ ] The upload size limit has been raised to a realistic level: in CKAN the default is 10 MiB.
- [ ] A published file that is cited is not overwritten: a correction is a new version; the file has a checksum.
- [ ] The user is given the file's persistent address on the portal, not a temporary cloud storage link.
- [ ] The dataset page address does not change when the description or title is edited: identifiers.
- [ ] If a DOI is promised, `https://doi.org/…` leads to the dataset page, not to the home page, a test instance or `localhost`.
- [ ] If the page shows a Handle, it resolves through `hdl.handle.net`, and the prefix is not the `123456789` placeholder from the DSpace distribution.
- [ ] A withdrawn dataset with a DOI is replaced by a tombstone page, not a "not found" page.
- [ ] If the portal has moved, old addresses redirect rather than answering "not found".

Citation and access

- [ ] There is a "How to cite" block, and BibTeX or RIS with the same identifier can be downloaded: citation export.
- [ ] The citation has the dataset's publication year, not the year the record was created; there is a version or, for updated datasets, an access date.
- [ ] If the dataset has a DOI, the DataCite citation matches the one on the page by author, year and version.
- [ ] The dataset shows an organisation and a contact: access, roles and workflows.
- [ ] Descriptions of restricted datasets are visible without a login; downloading is not.

Search, languages, harvesting and feedback

- [ ] A word from the middle of a description is found by search, and with hundreds of datasets there is a filter by organisation and format: other capabilities.
- [ ] After switching language, the same word in that language finds the dataset, and the translated description matches the original text.
- [ ] A tabular file has a table on the page or a direct link. For a restricted file, the content is not visible to those it is restricted from.
- [ ] A harvested dataset has a link to the original, and a harvest source shows the date of the last successful run and the number of datasets.
- [ ] The counter sees the access method by which the data is actually used: page, file or API.
- [ ] An error report reaches the data owner. A person and a response time are named.

The network part can be checked with one script. Substitute the portal address:

```
PORTAL=https://portal.example
for path in sitemap.xml data.json catalog.rdf catalog.ttl \
            "api/3/action/package_search?rows=1" \
            "oai?verb=Identify" \
            "csw?service=CSW&request=GetCapabilities"; do
  code=$(curl -s -o /dev/null -w "%{http_code}" -L --max-time 20 "$PORTAL/$path")
  echo "$code  $path"
done
```

A `200` response at an address that fits the catalog type is a good sign. A `401` or `403` response, or a login page, means there is no machine access, even if it is promised. The CKAN, OAI-PMH and CSW addresses do not fit every platform: no response at an address for another type is not an error.

Indicator catalog

Details are in the chapter on indicator catalogs.

- [ ] Territories, industries and other dimensions are coded with shared code lists, and the code lists can be downloaded.
- [ ] The whole database can be taken in full: as an archive, or as a file per indicator plus a list of files.
- [ ] An export from the table builder gives the whole series, not just the slice shown by default.

- [ ] There is an API, and each indicator has a persistent query address.
- [ ] Besides Excel there is CSV and SDMX or JSON-stat.
- [ ] Methodology and unit of measurement sit next to the series, not only in a PDF.
- [ ] The API is described in OpenAPI, or at least on a web page with example queries.

Metadata catalog

Details are in the chapter on metadata catalogs.

- [ ] Every concept, code and description has a persistent address that does not change when the site moves.
- [ ] An outdated entry is marked, and it is visible what replaced it; entries are not silently deleted.
- [ ] An entry's address serves a program SKOS, SDMX or DCAT, not only a page.
- [ ] The whole vocabulary or list can be downloaded as a single file.
- [ ] Each entry has a named steward, and the catalog has written inclusion rules.
- [ ] The catalog is registered one level up: in BARTOC, re3data, FAIRsharing, a FAIR Data Point index or a national single point.
- [ ] Programs are not answered with a bot check instead of data.

Microdata catalog

Details are in the chapter on microdata catalogs.

- [ ] The study description is visible without a login: sample, unit of observation, period, territory.
- [ ] Variables have the question text, labelled values and frequencies.
- [ ] Each file states its access type; the conditions, what an application needs and the response time are written before the application.
- [ ] Descriptions are exported in DDI, there is an API or OAI-PMH, and they respond to a program without an "I am not a robot" check.
- [ ] The study has a DOI or a persistent address and a ready citation.
- [ ] Open files are also available as CSV or Parquet, not only in SPSS and Stata formats.
- [ ] The catalog sits on the owner's domain with HTTPS, not on an IP address or a contractor's subdomain; maintenance is budgeted.

Before installing an extension

Details are in the chapter on extension ecosystems.

- [ ] It is clear who releases the extension and answers bug reports. One person with no answers for two years and no tests is a reason to look for another module.
- [ ] Your core version is named in the automated tests, a compatibility table or a branch name. A fresh release with old tests does not count. "It worked for our neighbours" does not count.

- [ ] If there are several copies, the decision record states the address of the one you install.
- [ ] The installation command points to this repository and to a branch that exists in it.
- [ ] There is a licence file, and it permits what you need.
- [ ] For an external program, it is recorded how it gets the file: a short signed link or a long-lived key.
- [ ] The decision record states what to do if the extension stops being maintained: justifying a decision.

Before buying or installing a platform

Details are in the chapters on deployment and operations and on the cost of leaving.

- [ ] There is an official container and a short upgrade guide.
- [ ] It is known how many recent versions are supported and how often they come out.
- [ ] It has been counted how many extensions the mandatory functions need and who releases them.
- [ ] It is named who will be on duty two years from now: an in-house position, a contractor or the service vendor.
- [ ] It has been checked that all descriptions and files can be exported in full; otherwise leaving the platform will be expensive.
- [ ] A pilot has run on ten real datasets, not demo ones.

Before calling a dataset published

Details are in the chapter on the publishing process.

- [ ] The dataset has a persistent address.
- [ ] The licence on the record, on the file and in the dump is the same link or a known code. "Not specified", a code without text and a phrase in the footer do not count: licences.
- [ ] The file or service opens, or it is honestly stated why it is unavailable and how to request it.
- [ ] The update date and frequency are stated.
- [ ] The description is in the full dump and visible to the upper portal.
- [ ] A draft did not get into search before publication.
- [ ] A person or team is named who will answer about an error in the rows. An `info@` inbox without a name does not cover this.
- [ ] The inventory or queue records why this dataset is being published now.

While the dataset remains published

Details are in the chapter on the publishing process.

- [ ] The file date is no older than the promised frequency. An empty frequency means no update has been set for this dataset.

- [ ] A withdrawn dataset is marked: the old address opens and explains why there are no more releases or what replaced it.
- [ ] A request for a missing dataset goes into the queue rather than staying only an email.
- [ ] The data inventory exists as a document (a table or a description file), not only in the editor's memory.

Before switching off a catalog

Details are in the chapters on lifecycle and on archiving.

- [ ] The closure is announced in advance, months rather than days ahead; for the transition period the catalog runs read-only.
- [ ] The data is handed over to another catalog or archive, or it is stated explicitly that it has been withdrawn from publication.
- [ ] Old addresses either redirect or, for at least a year, say where the dataset has moved.
- [ ] A snapshot of descriptions is saved: a dump through the API or a DCAT file, not just site pages.
- [ ] Files that exist only in this catalog are saved together with the addresses they were taken from.
- [ ] It has been checked that the site is in Internet Archive or a national web archive and that the archive can reach it, including from abroad.
- [ ] The upper portal and the registries where the catalog was registered have been notified.

Before setting up your own portal

Details are in the chapter "When You Don't Need a Portal".

- [ ] The recipient is named: who takes the descriptions and in which profile.
- [ ] Datasets, publishing organisations and update frequency have been counted.
- [ ] A person is named who will update the publication three years from now, and separately who updates the server if a portal is installed after all.
- [ ] Someone else's catalog that already accepts these datasets has been checked: regional, national, university, Zenodo.
- [ ] If your own site or repository remains, each dataset has a licence, a date and a direct link to the file.
- [ ] The event on which the decision not to have a portal is reopened is written down.

Before an entry point for programs and a data space

Details are in the chapter on AI and data spaces.

- [ ] The description dump, the licence and a direct link to the file already work.
- [ ] An MCP server or an `llms.txt` file is set up when programs are already in use and there is someone to maintain it.

- [ ] Search by phrase sits next to ordinary search and returns dataset records.
- [ ] A chat links to the dataset. The record, file and licence remain.
- [ ] A data space is set up when several organisations exchange data and their rights differ.

Before fixing quality

Details are in the chapter on quality levels.

- [ ] The task names the level: data, descriptions, the catalog as a service or the system of catalogs.
- [ ] The person doing it is whoever owns that level: the data owner, the editor, the platform administrator or the country coordinator.
- [ ] The queue starts with files that open and can be read. The dump jumps ahead when a harvesting date is set from above.
- [ ] Someone else's score is placed on a level: a record score and MQA are descriptions, a country index is the system, the number of datasets is a record counter.

The minimum for publishing without a portal

Details are in the chapter on making a publication findable.

- [ ] One page with a persistent address. The path does not change when the title is edited.
- [ ] The description in the markup is 50 to 5,000 characters, with the same meaning as the page text.
- [ ] The licence is a link to a specific version, the same on the page, in the file and in the DOI record.
- [ ] The HTML has schema.org/Dataset and a direct link to the file. There is no empty DOI in the markup.
- [ ] The page opens without a login. If there are several pages, there is a sitemap. `robots.txt` sits at the root of the site's host name.
- [ ] A file in a harvester's profile only if there is a harvester, and that profile's mandatory fields are filled in.
- [ ] The table has a `datapackage.json`. The update date and contact are written on the page.
- [ ] A DOI if the dataset is cited. `sameAs` holds another page for the same dataset.

Ready file templates (README, `datapackage.json`, `CITATION.cff` and schema.org markup) are in the templates.

Static catalog

Details are in the chapter on static sites.

- [ ] Each dataset's page is built as HTML. The description text is in the page file, not only loaded by a script.

- [ ] The page address does not change when the title is edited. Renaming the file changes the address.
- [ ] The HTML has schema.org/Dataset. DCAT markup in page attributes does not replace it.
- [ ] The file of all descriptions opens without a login. The record has a publisher, a licence, a modification date and a direct link. The `conformsTo` line has been checked against the fields.
- [ ] The licence field holds a known licence. The value `No licence` does not count.
- [ ] The `sitemap.xml` sitemap opens.
- [ ] If an editor form is enabled, an edit shows up as a commit. A separate login server, if the hosting requires one, is recorded in the decision record.
- [ ] The site size and build time fit within the hosting limits.

A repository with data

Details are in the chapter on Git repositories.

- [ ] One dataset, one package. The descriptor is called `datapackage.json` and sits at the package root.
- [ ] `LICENSE` holds the data licence. If there is a script alongside, the code licence is named separately.
- [ ] `CITATION.cff` has `type: dataset`. There is no empty DOI in the file.
- [ ] The file is smaller than the platform limit. A large file goes in a release attachment or outside Git, and the repository keeps its address.
- [ ] The cited release is fixed with a tag. On GitHub the Zenodo link is enabled before this release, the record type is "dataset", and the licence matches `LICENSE`.

Files in the cloud or on a shared hub

Details are in the chapter on cloud storage and hubs.

- [ ] Next to the files are a `README.md`, a licence and checksums. The licence on the page, in the record and in the file is the same.
- [ ] The file address is persistent. It is not a temporary link and not a Requester Pays bucket.
- [ ] The file downloads without a login. Checked with a request without a cloud key and in a browser where you are not logged in to the platform.
- [ ] The cited release has a DOI or a copy outside the platform.
- [ ] Before uploading, it was checked that your datasets can be retrieved in full. If storage is paid for by a sponsorship programme, the decision record states the day it expires.

Bibliography and Links

This collects the external sources from the chapters' "Sources" sections. Links to other chapters of the book are not included in this list.

Catalog registries

- dataportals-registry, release 1.27.0
- Trust score in the registry
- re3data
- FAIRsharing
- dataportals.org, Open Data Inception, CKAN ecosystem catalog
- Vocabulary and ontology registries: BARTOC, Linked Open Vocabularies, Ontology Lookup Service, FAIR Data Point index
- European register for protected public sector data on data.europa.eu
- Registry of Open Data on AWS and the AWS Open Data Sponsorship Program
- Source Cooperative

Standards

- DCAT 3 (W3C), Recommendation of 22 August 2024
- DCAT-AP 3.0.1, SEMIC recommendation of 27 October 2025; 3.0.2 draft
- DCAT-US 3.0
- GeoDCAT-AP 3.1.0
- schema.org/Dataset
- SKOS, XKOS, OWL 2
- ISO/IEC 11179-3:2023, metadata registries
- HealthDCAT-AP, 2026 draft specification
- FAIR Data Point specification
- DataCite and the DataCite Metadata Schema 4.7
- Persistent identifiers: DOI states in DataCite, tombstone pages, DataCite fees, Crossref sanctions policy, Handle.Net Registry, ARK Alliance, w3id.org, PURL, CSTR
- DDI
- SDMX
- STAC
- OAI-PMH 2.0, robots.txt, RFC 9309
- Frictionless Data Package and Table Schema, CSV on the Web, the schema.data.gouv.fr schema catalog
- CITATION.cff

- Citation: DataCite content negotiation, CSL styles repository, exposing metadata to Zotero, GOST R 7.0.108-2022 (in Russian): references to electronic documents on networks
- Croissant 1.1
- RO-Crate 1.3, CDIF
- Hugging Face datasets

Platforms

- CKAN 2.12 documentation and the CKAN ecosystem catalog: extensions and portals
- InvenioRDM customisation, Dataverse external tools, GeoNetwork schema plugins, GeoServer community modules
- ckanext-dcat
- ckanext-doi, ckanext-citeproc
- Custom description fields: ckanext-scheming, Dataverse metadata blocks, InvenioRDM custom fields, GeoNetwork schema plugins, Huwise templates, World Bank Metadata Editor
- Dataverse features and data citation
- InvenioRDM licence, MIT
- GeoNetwork licence, GPL-2.0
- DKAN licence, GPL-2.0
- PxWeb and PxWebApi 2, .Stat Suite, SDMX-RI, Fusion Metadata Registry, DHIS2
- SDMX, JSON-stat, StatDCAT-AP
- ERDDAP, wis2box, Fairdata
- Harvesting in GeoNetwork, pycsw
- JKAN, Open SDG, PortalJS, Datasette
- Zenodo and archiving GitHub releases
- Metadata catalogs: FAIR Data Point, Skosmos, OntoPortal, OLS, MOLGENIS

Practices and methodologies

- Michael Nygard, Documenting Architecture Decisions, 15 November 2011: a short decision record with context, the decision itself and consequences; a superseded record is kept
- Data on the Web Best Practices (W3C), 31 January 2017: update frequency, explaining unavailable data, collecting and publishing feedback
- GSA, DCAT-US v3.0 Implementation Guide, revision 1.1 of 9 September 2026: inventory, a mandatory contact, withdrawing a dataset with an explanation
- data.gouv.fr guide: publishing a dataset (in French): with regular updates the record is sent via the API
- FAIR Principles
- International Open Data Charter, 2015
- 5-star open data

- T. E. Hardwicke et al., Analytic reproducibility in articles receiving open data badges at the journal Psychological Science, Royal Society Open Science, 2021: of 25 articles with an open data badge, the checked figures matched in 9 without help from the authors
- Joint Declaration of Data Citation Principles, FORCE11, 2014
- Research Data Alliance, Precise, Actionable Reference to Dynamic Data, 2025: how to cite dynamic data and parts of it
- Make Data Count, Data Citation Corpus: a corpus of data citations; GBIF citation guidelines: download DOIs and derived datasets
- Creative Commons
- Open Definition 2.1: what counts as an open licence; commercial use allowed, no ban on commercial use or on changes
- SPDX licence list
- CoreTrustSeal
- Open Data Maturity, 2025 report: a survey of 36 countries; EU-27 average 86%, all participants 81%; methodology: four dimensions at 25% each
- MQA: an assessment of the records data.europa.eu has harvested from portals; catalog scale 0 to 7.5; methodology, page of 6 October 2026
- OECD, Digital Government Index and OURdata Index: 2025 Results and Key Findings, 16 February 2026: OECD average 0.53 against 0.48 in the 2023 edition
- Global Data Barometer, second edition, 29 May 2025: 43 countries in Africa and Latin America
- Open Data Inventory (ODIN), 2024/25 edition: 197 countries
- Open Data Barometer: last full edition in May 2017; Global Open Data Index: archived, last edition 2016/17

How people search for data

- N. W. Paton, J. Chen, Z. Wu. Dataset Discovery and Exploration: A Survey. ACM Computing Surveys, 2023: a review of methods for discovering and exploring datasets
- K. Gregory, P. Groth, A. Scharnhorst, S. Wyatt. Lost or Found? Discovering Data Needed for Research. Harvard Data Science Review, 2020: how researchers look for other people's data, based on an international survey
- K. Sostek, D. M. Russell, N. Goyal, T. Alrashed, S. Dugall, N. Noy. Discovering Datasets on the Web Scale: Challenges and Recommendations for Google Dataset Search. Harvard Data Science Review, 2024: the experience of Google Dataset Search
- Web Data Commons: schema.org markup from Common Crawl: how much Dataset markup, and of what kind, appears on the web
- Handbook on Using Administrative Data for Research and Evidence-based Policy: how to find and obtain administrative data for research
- I. Begtin's blog begtin.substack.com (in Russian): analyses of portals, data search and publishing practices cited in the book's chapters
- I. Begtin's Telegram channel @begtin (in Russian): short notes on portals, their closure and archiving since 2016; individual posts are cited in the chapters' "Sources" sections

Preservation and closure of data

- Pew Research Center, When Online Content Disappears, 2024: the share of vanished pages and broken links on websites
- T. Berners-Lee, Cool URIs don't change, 1998; W3C URI persistence policy
- M. Laakso, L. Matthias, N. Jahn, Open is not forever: a study of vanished open access journals, 2020
- Sunlight Foundation, Classifying changes to public access to information on US government websites, 2017: eight ways government information disappears
- Esli Byt Tochnym ("To Be Precise"), In 2025, 140 datasets disappeared in Russia and 425 indicators stopped being updated (in Russian), 2025
- Congressional Research Service, Data.gov: Implementation and Perspectives on Its Functions, 2026
- End of Term Web Archive, Data Rescue Project, Harvard LIL: data.gov archive, National Digital Archive (in Russian)
- Platform closures: data.world, A fond farewell, 2026; Microsoft, Retiring the Planetary Computer Hub, 2024; QRI, Winding Qri Down, 2022
- BagIt, RFC 8493, WACZ

Data for AI

- Open Data Institute, The UK government as a data provider for AI, November 2024: general-purpose models hardly remember data.gov.uk figures
- Open Data Institute, A framework for AI-ready data, version 2, December 2025
- Open Data Product Specification: describing a data product
- Data Spaces Support Centre, Blueprint 3.0, February 2026; CEN/CLC/TS 18331:2026: data space maturity assessment
- OMB, memorandum M-25-05, 15 January 2025: an inventory of all data assets, updated within 90 days and at least once a year, a `data.json` file; training AI is named among the purposes of open access
- Open Knowledge Foundation, MCP for Open Data Portals: Trust Depends on Understanding the Data, June 2026: MCP pilots in Brazil and Uruguay, frequent questions as tests
- Open Data Watch, Making Official Statistics Work in an AI-Driven World, September 2026: official MCP servers of statistical offices
- World Bank, Data360 MCP: a server with agent instructions and a coverage check

Innovative catalogs

- data.gouv.fr, API tabulaire (in French): an API for every tabular file
- Overture Maps, GERS: persistent object identifiers
- DoltHub, What is Dolt?: table versions as in Git

- OpenNeuro, user guide; DANDI, publishing Dandisets: checks on entry
- Copernicus Data Space Ecosystem, openEO: processing next to the data
- Data Commons, HDX HAPI, OpenSanctions, statement model: combining many sources
- microdata.no, Redivis, access levels: tiered access

EU law

- INSPIRE Directive 2007/2/EC
- Directive (EU) 2019/1024 on open data
- Regulation (EU) 2023/138 on high-value datasets
- Regulation (EU) 2022/868, the Data Governance Act
- Regulation (EU) 2025/327 on the European Health Data Space (EHDS)
- Regulation (EU) 2023/2854, the Data Act
- Regulation (EU) 2024/1689 on artificial intelligence: a training data summary from general-purpose model providers

Stable Addresses for Links

The book's permanent address is datacatalogs.begtin.tech. You can link to chapters and sections from outside: their addresses do not change silently.

Addresses of the book

Edition	Site	PDF
English	datacatalogs.begtin.tech/en/	data-catalogs-book-en.pdf
Russian (original)	datacatalogs.begtin.tech	data-catalogs-book.pdf

An English chapter has the same path as the Russian one with the `/en/` prefix: for example, `https://datacatalogs.begtin.tech/en/types/classification/` and `https://datacatalogs.begtin.tech/types/classification/`. A full chapter address is the domain plus the path, with a trailing slash.

The book's former address `ivbeg.github.io/data-catalogs-book/` redirects to the new domain together with the chapter path, so old links keep working. How to cite the book as a whole is described in About this edition.

How addresses stay stable

A chapter's path is not changed silently. If a chapter is renamed, a redirect is set up from the old address. Old paths about APIs and identifiers already lead to the new chapters.

The list of paths for the registry knowledge base is maintained together with the site, separately from the text for readers.

About the Author



Ivan Begtin works on open data: he collects and analyses government and research data, builds services that help people find it, and writes about how it is published. He is the founder and CTO of the Dateno dataset search engine, the founder and director of the non-profit Infoculture ("Information Culture"), and a co-founder of the Open Data Armenia project (opendata.am).

He has worked on government openness and open data since 2009, and on data catalogs since 2010. This book grew out of the same work. He started the catalog registry from which the book's figures are calculated for his own projects, and many examples and counts in the chapters come from his notes.

Career

Ivan Begtin has worked in IT since 1999. He started out administering Unix servers, then worked as a developer and headed a data mining laboratory.

In 2009 he launched the project "The State and Its Information" (opengovdata.ru), with news and a knowledge base on open government data in Russia. The project ran until 2015. At the same time he began saving copies of government websites that were being closed or changed.

In 2010 he became one of the founders of the Apps4Russia contest, in which developers built applications on government data. In March 2012 the contest's founders set up the non-profit Infoculture, and Ivan Begtin has led it ever since. Infoculture's projects include Goszatraty (clearspending.ru), an open database of government contracts, and the annual Open Data Day in Moscow.

From 2019 to 2021 he headed the Open Data project area at the Accounts Chamber of Russia. There a team of six built the Gosraskhody portal (spending.gov.ru) in four and a half months. It launched in November 2019 with contracts, subsidies and spending on national projects, and the data itself was served as open data and through an API.

Another long-standing interest of the author is plain language in government texts. In 2014 he adapted to Russian the readability formulas that English-speaking countries use to check official texts. This book follows the same rules: simple words and the conclusion first.

Data catalogs

The first catalog was the Hub of Open Data, hubofdata.ru, on CKAN. Ivan Begtin built it in 2010–2011 under the OpenGovData brand, and it appeared before Russia's government open data portals (see the short history of catalogs). Later Infoculture launched a data portal for non-profit organisations, NGO Data (ngodata.ru), on the same platform. The book mentions both community

portals in the chapter on metadata export: in November 2024 Google Dataset Search ranked their datasets first for Russian-language queries, because their pages carry schema.org markup for search engines.

Ten years later there were so many catalogs that catalogs of catalogs became necessary. In November 2021 Infoculture launched datacatalogs.ru, a list of Russia's data catalogs, with 245 records at launch. In 2023 the work went beyond one country. The Common Data Index project started an open registry of catalogs from around the world, which had 1.7 thousand records in April 2023. Today this is the dataportals-registry: release 1.27.0 of 15 October 2026 describes 43,614 catalogs and 790 platforms.

In March 2024 the Dateno search engine went live on top of the registry. From a registry record it learns a catalog's platform and API addresses and uses them to collect dataset descriptions. As of 16 October 2026 Dateno holds 22 million datasets from 5,236 sources. Why some catalogs are easy to harvest while the harvester never reaches others is explained in the chapters "Search Engines and Aggregators" and "Discoverability".

Preservation and tools

Since 2017 Infoculture has run the National Digital Archive (ruarxive.org). It saves websites and data that may disappear, above all government ones. In 2022 the archive ran a campaign to archive open data portals, and in 2025 it saved the websites of Rosstat and the EMISS statistics system. How such work is done is described in the chapter "Archiving and Rescuing Catalogs".

For his work the author writes open source tools. undatum processes data files from the command line: CSV, NDJSON, XML and other formats. metacrafter identifies what is in a table's columns: identifiers, codes, personal data. apibackuper saves data that is served only page by page through an API. It is mentioned in the chapters on APIs and bulk download and on archiving.

The author and the book's sources

The book and the data catalog knowledge base are supported by Dateno. The catalog registry and Dateno are the author's projects, so the book cites them more often than others. This can be checked. The registry figures come from open queries, which are collected in the appendix "Using the Registry" and give the same numbers on release 1.27.0. Software versions and platform capabilities were checked against the platforms' own documentation, not against the registry.

The author's notes from his Telegram channel and newsletter are signed "I. Begtin" in the "Sources" sections, and each has a date. They are observations as of the day of the note. If a figure may have changed since, check the date.

Where to read and how to get in touch

- Telegram channel "Ivan Begtin" (in Russian): open data, data engineering, government, privacy and digital preservation. As of 16 October 2026 the channel has 9,093 subscribers.
- Newsletter begtin.substack.com: long texts on the digital, open and preserved state.
- Code: github.com/ivbeg; LinkedIn: linkedin.com/in/ivbeg; ORCID: 0000-0002-9084-9643.

- Email: ivan@begtin.tech.

Comments and corrections to the book can be sent by email or on Telegram.

Sources

- The author's profiles: GitHub: the Dateno, APICrafter, Open Data Armenia and Infoculture projects; LinkedIn: "The State and Its Information" (2009–2015), saving government websites (since 2009), Apps4Russia (since 2010), the National Digital Archive (since 2017); Telegram channel description: the channel's topics and number of subscribers as of 16 October 2026; Facebook: co-founder of Open Data Armenia, the Accounts Chamber in 2019–2021
- TAdviser, Begtin Ivan Viktorovich (in Russian): work in IT since 1999, the data mining laboratory, co-founder of Apps4Russia
- NGO Data, the Infoculture organisation page (in Russian): the organisation was set up in March 2012 by the founders of the Apps4Russia contest
- TAdviser, Gosraskhody (in Russian): launched on 18 November 2019, built from June to November 2019 by six people, open data and API; Roscongress, speaker profile (in Russian): head of the Open Data project area at the Accounts Chamber
- I. Begtin, What "Plain Russian" means from a technology perspective, Habr (in Russian), 2 October 2014: readability formulas for Russian
- I. Begtin, Telegram (in Russian): the Hub of Open Data, built in 2010–2011 under the OpenGovData brand, 14 April 2021; the launch of datacatalogs.ru, 3 November 2021; Common Data Index and the catalog registry, 10 April 2023; the launch of Dateno, 13 March 2024
- The dataportals-registry of data catalogs, release 1.27.0 of 15 October 2026; Dateno as of 16 October 2026
- National Digital Archive; Open Data Armenia