AGORA: An Access Profile for Agent-Ready Government Records

Composing existing open-data standards into a conformance-testable consumption layer, with a reference implementation across 46 official sources in Colombia, Peru and Mexico

Cristian Correa*

Abstract

Two decades of open-government-data policy produced a mature stack of standards for *describing* public data (DCAT and its national profiles), for *designing* agency APIs (national API guidance), and for *encoding* specific domains (OCDS for procurement, GTFS for transit, Akoma Ntoso for legislation). What that stack does not deliver is a uniform, conformance-testable contract for *consuming* a public record: resolving an identifier, querying by field, learning what changed since yesterday, trusting a timestamp, and doing all of it from a program or an AI agent rather than a browser. We call this the consumption gap. It is where open data fails in practice, and it is now the layer that autonomous agents hit first.

This paper makes four contributions. First, a layered analysis of the existing standards landscape that locates the gap precisely and argues that a new metadata vocabulary or domain schema would be the wrong response. Second, AGORA (Agent-ready Government Open Records Access), a minimal access profile composed over OpenAPI, JSON Schema, DCAT and the domain schemas: eight requirements (typed records, resolvable identifiers, a uniform query contract, change feeds, a provenance envelope, bulk access, agent surfaces, and an operational contract) arranged in three conformance levels, each with a machine-checkable test. Third, a reference implementation: a production platform that exposes 80 endpoints over 46 official sources in Colombia, Peru and Mexico through one contract, one MCP server and markdown twins of its documentation. Over 80 days and 25,153 authenticated requests, records served at the source's pace answered with a median latency of 2.6 s and a 95th percentile of 27.9 s, while the same contract over a dataset maintained under the profile answered in 75 ms at the median and 366 ms at the 95th percentile; 7.3% of requests failed because the source did. Fourth, an AI-assisted publishing harness for institutions, in which a language model proposes record schemas, field annotations and mappings to domain standards, and a fixed set of verification gates decides what ships. An assessment of 37 official sources against the profile's criteria finds that agent-oriented surfaces are absent from every one, that change feeds and freshness metadata are the rarest capabilities, and that the only sources approaching LEVEL 2 are those already publishing under a governed domain standard. We close with an adoption path drawn from the histories of GTFS and OCDS: a reference implementation and validators first, a neutral steward second, a specification that stays small throughout.

Keywords: open government data, data standards, DCAT, OCDS, API design, Model Context Protocol, AI agents, data provenance, Latin America, digital public infrastructure

1 Introduction

A public record is a fact a state has agreed to make available: whether a company is registered, whether a vehicle has an outstanding fine, which contracts an agency awarded last month, what a court decided in a case. Every modern open-data policy, from the Sebastopol principles of 2007 to the European High-Value Datasets

*Croma. Correspondence: cris@usecroma.com. Version 1, 26 August 2026.

Regulation of 2023, rests on the premise that the value of such records is realised only when someone other than the state can use them [12, 33]. The policies succeeded at their first objective. Most governments now run a national catalog, most catalogs speak a DCAT dialect, and the number of published datasets is counted in the tens of thousands per country [29].

They did not succeed at the second. The literature on open-data utilisation has documented the same pattern for a decade: portals optimised for the number of datasets rather than for reuse, files that are stale or contextless, and a persistent absence of programmatic access [3, 21, 40, 52]. A catalog entry says that a dataset exists and where a file can be downloaded. It does not let a program resolve one record, filter by a field, page through results, ask what changed since a given date, or know how fresh the answer is. Those are consumption questions, and no standard in wide use answers them for government records as a class.

Two things have changed since that literature was written. The first is the consumer. Software agents driven by large language models now retrieve public data on behalf of people, at a volume that has become measurable in web traffic [6], and they do so through structured tool interfaces rather than through a browser [2, 24]. An agent cannot fill a web form, wait for a certificate to render, and read the result; it needs a typed answer, a stable identifier and a timestamp. The second is that the cost of building a well-typed access layer over an existing system has fallen sharply, because the same language models can do most of the tedious part: reading a data dictionary, proposing a schema, writing the field annotations, mapping fields onto a domain standard. What has not fallen is the cost of being wrong, which is why the interesting engineering question is not whether a model can draft an interface but which checks decide whether the draft ships.

This paper is written from the position of an operator. Croma runs a public API that exposes official records from Colombia, Peru and Mexico to developers and agents through one contract: 80 endpoints over 46 sources at the time of writing, with an MCP server and a markdown rendering of every documentation page [10]. Building it meant confronting, source by source, the same handful of missing properties: no identifier that resolves, no query beyond a single lookup form, no way to learn what changed, no timestamp, no machine-readable error, no availability commitment. It also meant measuring, in production, what those absences cost the consumer. Over the 80 days reported here, the median request that had to be answered at the source's pace took 2.6 seconds and the slowest five percent took more than 27; one request in fourteen failed because the source did. The same contract, over the two sources we maintain as datasets under the profile described below, answered in 75 milliseconds at the median.

Contributions.

(1) A layered map of the government open-data standards landscape (policy, catalog, API guidance, domain schema, platform, agent surface) that locates the consumption gap and explains why the mature layers cannot close it on their own (sections 2 and 3).

(2) Agora, a minimal access profile with eight requirements and three conformance levels, each requirement paired with the existing standard it composes and with a test a validator can run without human judgement (section 4, section A).

(3) A reference implementation in production, described at the level of its contract and its two source archetypes, with latency, freshness and availability measurements from 25,153 requests (sections 5 and 7).

(4) A publishing harness for institutions in which a language model proposes and a fixed set of gates disposes, with the gates specified as machine-checkable conditions and the model's authority bounded explicitly (section 6).

(5) An assessment of 37 official sources in three countries against the profile's criteria, from public documentation, with the rubric and per-source evidence published alongside the paper (section 7, section C).

(6) An adoption and governance path derived from the two domain standards that achieved global uptake,

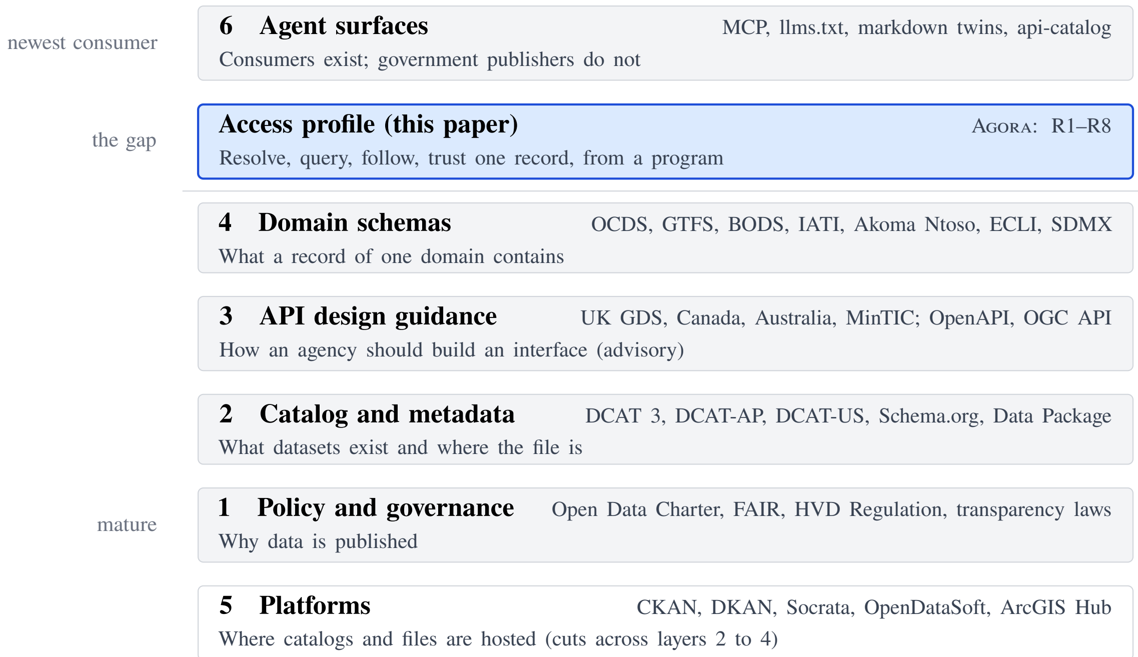


Figure 1: Six layers of the government open-data landscape. The mature layers decide why data is published, what datasets exist, how an agency should design an API, and how a domain is encoded. The access layer, which decides how one record is resolved, queried, followed over time and trusted, is where the profile in this paper sits. The agent layer above it is the newest consumer and the one least served today.

GTFS and OCDS, and adapted to the Latin American institutional landscape (section 8).

What this paper does not propose. It does not propose a new metadata vocabulary, a new domain schema, or a new transport. DCAT describes datasets well; OCDS, GTFS, BODS and Akoma Ntoso encode their domains well; OpenAPI and JSON Schema describe interfaces well. Agora is a profile in the DCAT-AP sense of the word: a constrained, testable way of using standards that already exist, for a purpose they were not individually designed for.

2 Background: the standards landscape by layer

The open-government-data field is not short of standards. It is short of a way to tell which of them bears on a given problem. We organise the landscape into six layers (fig. 1), ordered from policy to consumer, and characterise each by what it decides and what it leaves undecided.

2.1 Layer 1: policy and governance

The International Open Data Charter's six principles (open by default; timely and comprehensive; accessible and usable; comparable and interoperable; for improved governance and citizen engagement; for inclusive development and innovation) are the most widely adopted statement of intent, adopted by 29 national and 146 subnational governments at the time of writing [31]. They descend from the Sebastopol principles of 2007 and the Sunlight Foundation's ten principles of 2010, which already asked for machine-processable, timely, primary and permanent data [33, 44]. The FAIR principles, written for research data, supply the vocabulary most later work borrows: findable, accessible, interoperable, reusable [48].

Three legal instruments turn intent into obligation and are worth singling out because they name the access layer directly. The European Union's Open Data Directive requires member states to make high-value datasets available free of charge, in machine-readable form, *through APIs and, where the annex requires it, as a bulk download*, under CC0 or CC BY 4.0 or an equivalent open licence; the implementing regulation of 2023 enumerates six thematic categories, including company registers and mobility, and applies from 9 June 2024 [12, 13]. The Interoperable Europe Act, in force since April 2024, makes interoperability assessments mandatory for public services that cross borders [14]. In the United States, the OPEN Government Data Act obliges every federal agency to maintain a machine-readable inventory of its data assets [47]. In Latin America, the equivalents are transparency laws (Colombia's Ley 1712 of 2014, Mexico's Ley General de Transparencia of 2015, re-issued in 2025, and Peru's open-data strategy decree of 2017) and open-by-default decrees; they mandate publication and catalogs, but not, as of this writing, programmatic access [1, 7, 8, 35–37].

Benchmarks reward what these instruments require. The OECD's OURdata Index scores availability, accessibility and government support for reuse across forty countries, and its 2023 edition found only 48% of high-value datasets available as open data [29]; the Open Data Barometer and its successor, the Global Data Barometer, score policy and impact. None scores whether a published dataset answers a per-record query. A government can therefore rank well while every one of its registries is reachable only through a web form, and several do.

2.2 Layer 2: catalog and metadata

The W3C Data Catalog Vocabulary (DCAT), a Recommendation since 2014 and in its third version since August 2024, is the base vocabulary for describing datasets, their distributions and, since version 3, dataset series [50]. Europe's DCAT-AP profile, stewarded by the SEMIC action of the Interoperable Europe programme and at version 3.0.1 since October 2025, is the de facto continental dialect with national children in Germany, Italy and Switzerland and thematic children for geospatial and statistical data [41]. The United States publishes DCAT-US, the schema of every federal agency's `data.json`: version 1.1 remains in production and harvested, and version 3.0 is published with its implementation guidance still in draft [16, 38]. Alongside DCAT sit Schema.org's Dataset type, which powers Google Dataset Search [27], Berners-Lee's five-star deployment scheme, which still frames how publishers think about machine readability [5], and the Frictionless Data Package, a lightweight JSON descriptor favoured by data journalists and research groups [11].

This layer answers the question *what datasets exist and where is the file*. It deliberately does not answer *how do I get record 4711*. A DCAT distribution has an access URL and a media type; it has no notion of a record, a key, a filter or a change. The DCAT-AP working group's own guidance treats data services as a distribution type, which is the closest the vocabulary comes, and even there the service's semantics are out of scope.

2.3 Layer 3: API design guidance

The United Kingdom, Canada, Australia, Singapore and India, among others, publish API design standards for their agencies: REST by default, OpenAPI descriptions, versioning and deprecation rules, cross-government API catalogs [4, 18, 46]. Colombia's Ministry of ICT maintains a national interoperability framework and a common exchange language, a dictionary of data elements for inter-agency exchange [22]. These documents are good engineering advice and they raise the floor for agencies that build APIs. They are also, without exception, advisory, per agency and silent on the properties a consumer of a *registry* needs across agencies: identifier semantics, change feeds, provenance, the shape of a not-found answer. An agency that follows the UK guidance to the letter can still publish a company register with no way to ask what changed since Monday.

Two technical standards from this layer matter to the profile. OpenAPI 3.1 and JSON Schema 2020-12 give an interface a machine-readable description that agents can read directly [34, 51]. The IETF's Problem Details format gives errors a shape [26], Web Linking gives discovery a header [25], and JSON canonicalisation gives a record a stable hash [39], and the `api-catalog` well-known URI gives a host a place to list its interfaces [43]. The OGC API Features standard is the one example of a government-relevant, per-record, filterable access API with an OpenAPI description and stable item URLs, and it is the pattern the profile borrows most from [20, 32].

2.4 Layer 4: domain schemas

The standards that actually changed how governments publish are the domain schemas. The Open Contracting Data Standard (OCDS), launched in 2014, is implemented by more than fifty governments and its registry lists over a hundred publishers [30]. The General Transit Feed Specification (GTFS) is used by more than ten thousand transit agencies in over a hundred countries [23]. The Beneficial Ownership Data Standard (BODS), the International Aid Transparency Initiative standard, the Fiscal Data Package, SDMX for statistics, Akoma Ntoso for legislation and judgments, and the European Case Law Identifier each did the same for one domain [9, 15, 28]. Identifier schemes such as the Legal Entity Identifier and the org-id.guide registry of organisation identifier lists sit beside them [17].

Two lessons from this layer shape everything that follows. First, the winners were small. GTFS began as a handful of CSV files that solved one consumer's problem, transit routing; OCDS is a JSON schema plus a validator. Second, they came with tooling and a neutral steward: GTFS dropped Google from its name to reduce vendor resistance, OCDS is stewarded by a non-profit and ships a validator, a data registry and a helpdesk. Neither imposed a transport, a catalog or a query language. That is why Agora composes them rather than competing with them: a procurement record under the profile is an OCDS release; a judgment's identifier is an ECLI where the jurisdiction issues one.

2.5 Layer 5: platforms

Governments deploy a small number of catalog platforms. CKAN, open source, powers data.gov, data.gov.uk, Brazil's dados.gov.br and Mexico's datos.gob.mx; DKAN, its Drupal relative, powers Peru's datosabiertos.gob.pe; Socrata (now Tyler Technologies' Data & Insights) powers Colombia's datos.gov.co and much of US state and local government; OpenDataSoft, ArcGIS Hub and uData serve the rest. Socrata is notable for shipping a per-dataset query API with field filters, paging and a system `:updated_at` column, which is why the Colombian sources hosted there score well on the profile's query criteria while the institutions' own systems do not (section 7). The platform layer is where the well-documented failure modes of open data live: vendor lock-in, stale files behind polished front ends, data without context, and fragmentation across jurisdictions.

2.6 Layer 6: agent surfaces

The newest layer has almost no government presence. The Model Context Protocol (MCP), an open standard for connecting language-model agents to tools and data, was donated to the Linux Foundation's Agentic AI Foundation in December 2025 with more than ten thousand active public servers and 97 million monthly SDK downloads reported at the time; its specification is revised on a roughly quarterly cadence, most recently in July 2026 [2, 24]. Google's Data Commons ships an MCP server over public statistics [42]. The `llms.txt` convention proposes a plain-text index a site publishes for language models [19]; as of mid-2026 a single US state, Maryland, had published one for its official portal, and the major model providers had not committed to reading them [45, 49]. We are not aware of any national registry that publishes an MCP server, a markdown rendering of its records or a structured tool description for agents. This is the whitest space in the landscape, and the profile's **R7** is written for it.

3 The consumption gap

Reading the layers together, a pattern emerges. Layers 1 and 2 decide that data exists and describe it. Layer 4 decides what a record of a given domain contains. Layer 3 advises an agency on how to build an interface, without deciding what the interface guarantees. Layer 6 has consumers waiting for interfaces that do not exist. Between describing a dataset and encoding a record there is no layer that says, for government records as a class, what a consumer may rely on. We call this the consumption gap and characterise it by seven properties that a consumer needs and that no layer guarantees.

G1. **Record resolution.** Given an identifier the state itself issues (a NIT, a RUC, a radicado, an OCID), obtain the record, or a definitive statement that there is none.
G2. **Field query.** Filter by the fields the domain cares about, with bounded, pageable results, rather than through a single lookup form or a full download.
G3. **Change.** Learn what changed since a point in time, without re-reading everything.
G4. **Freshness.** Know when the answer was true: the record's own last change and the moment the snapshot was taken.
G5. **Provenance.** Know which institution the record comes from, under what licence, and which version of the record this is.
G6. **Cross-registry identity.** Recognise that a company in the procurement register and the same company in the tax register are one entity, through identifier namespaces rather than string matching.
G7. **Machine and agent access.** Do all of the above from a program or an agent: a documented interface, machine-readable errors, declared rate limits, a tool description an agent can read.

Three kinds of evidence support the claim that this is where open data fails, and we present each in the sections that follow. The first is documentary: an assessment of 37 official sources in three countries against thirteen criteria derived from G1 to G7 (section 7.1). Its headline is that per-record programmatic access is documented by a minority of sources, change feeds and freshness metadata by almost none, and agent surfaces by none at all; where a source scores well it is because it publishes through a national catalog platform or a governed domain standard, not because the institution designed for consumption. The second is operational: eighty days of production traffic through a uniform contract over those sources (section 7.2). Its headline is that the consumer absorbs the variance of the source. The same request shape answers in a quarter of a second from one registry and in twenty-four seconds from another; one request in fourteen fails upstream. The third is structural: the profile's own conformance tests, which fail on most sources for the same three requirements (section 4).

Why the mature layers cannot close the gap. It is tempting to propose closing the gap by extension: add record-level properties to DCAT, add change-feed semantics to OCDS, add a mandatory API clause to every national guidance document. Each of these is being tried in isolation, and each is the wrong altitude. DCAT describes datasets and should keep doing so; a record-level query language does not belong in a catalog vocabulary. OCDS describes contracting and should keep doing so; a change feed is a property of any registry, not of procurement. National API guidance is per country by construction; a consumer that spans countries needs a contract that does not depend on which ministry wrote the guidance. What is needed is a thin layer with its own name, small enough that a domain standard can be embedded in it unchanged and a catalog can point at it unchanged. That is the design brief for AGORA.

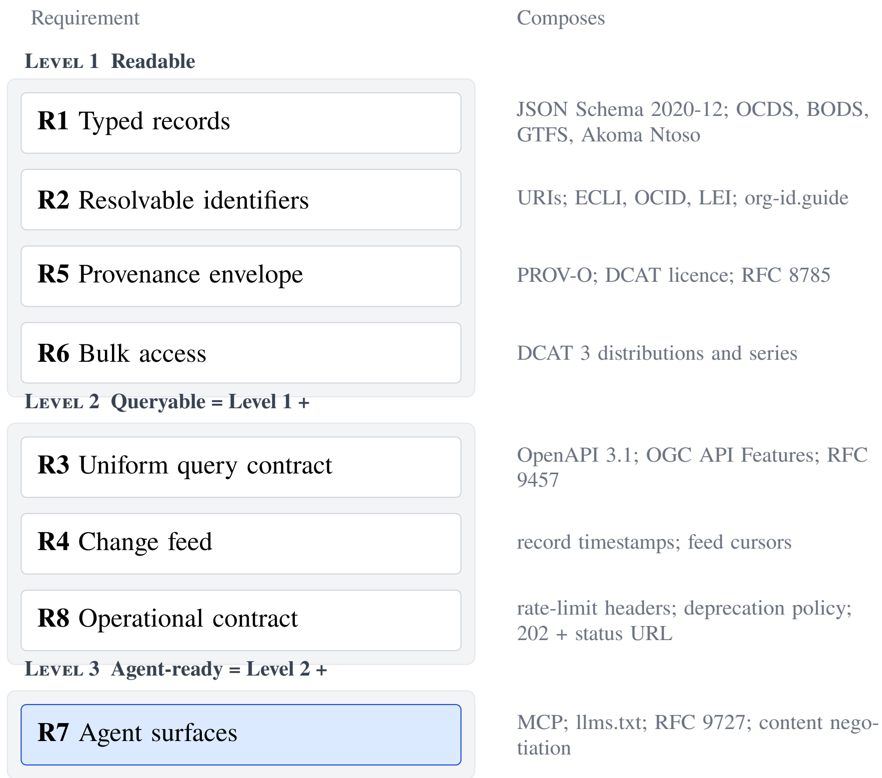


Figure 2: The eight requirements of AGORA and the three conformance levels they form. Each level is a superset of the one below it. LEVEL 1 makes a registry readable by a program; LEVEL 2 makes it queryable and followable over time; LEVEL 3 makes it consumable by an agent with no human in the loop. Every requirement composes an existing standard (right column) rather than introducing a competing one.

Why now. The gap has existed since the first portal. Two things make it urgent. Agents are a consumer class that cannot route around missing structure the way a human with a browser can; for them the gap is not friction but a wall, and requests from AI crawlers reached 4.2% of HTML traffic seen by one large network in 2025, on par with the largest search crawler [6]. And the tooling to close the gap has become cheap: a language model can draft a typed schema and its annotations from a data dictionary in minutes, so the binding constraint on an institution is no longer engineering hours but confidence that what ships is correct. The profile supplies the contract; the harness in section 6 supplies the confidence.

4 The AGORA access profile

AGORA (Agent-ready Government Open Records Access) is an access profile: a constrained, testable way of publishing a registry so that a program or an agent can consume it without knowing which institution built it. It is specified as eight requirements, **R1** to **R8**, grouped into three conformance levels (fig. 2). Each requirement names the existing standard it composes, states what a publisher must do, and states the test a validator runs. The tests are the specification; the prose explains them.

4.1 Design principles

Compose, do not compete. Every requirement is expressed in terms of a standard that already has a steward. Records are JSON Schema documents; interfaces are OpenAPI documents; datasets are DCAT resources;

procurement records are OCDS releases; errors are Problem Details. The profile adds constraints and a few field names, nothing else. A publisher already conformant to a domain standard should reach Level 2 without changing a record.

Minimal, then testable. A requirement enters the profile only if a validator can decide it without human judgement. This rules out worthy properties, such as "data quality" or "completeness", that are not decidable from the outside, and it is why the profile has eight requirements rather than the thirty a checklist would accumulate. The GTFS and OCDS histories both say the same thing: the smallest specification with a validator wins.

Agent-ready without being agent-only. Every surface an agent needs is also a surface a developer needs. The MCP tool description in **R7** is generated from the OpenAPI document; the markdown twin of a documentation page is the same content negotiated by media type; the `llms.txt` index points at the same URLs a human would bookmark. Nothing in the profile creates a second, agent-specific data path that could drift from the first.

Provenance by default. A record without a timestamp and a source is a rumour. The envelope in **R5** is mandatory at every level, because it is the property whose absence the consumer can least work around and the one publishers most often omit (section 7.1).

The source's identifiers, not ours. The profile does not mint identifiers for records that already have one. A company is its registry number; a judgment is its radicado or its ECLI; a procurement process is its OCID. Where an identifier has no scheme, **R2** asks the publisher to declare one, not the consumer to invent one.

4.2 Requirements

R1 Typed records composes JSON Schema 2020-12 and the domain schemas

Every record type the registry publishes has a JSON Schema. Every property carries a human-readable `description` and, where a value is coded, an enumeration or a reference to the code list. Field names are stable across versions; a renamed field is a new version. Where a governed domain schema exists for the record type (OCDS, BODS, GTFS, Akoma Ntoso, SDMX), the record either *is* an instance of it or carries a declared mapping to it.

Test. Fetch the schema from the interface description; validate a sample of records against it; every property has a non-empty description; every enumerated property lists its values.

R2 Resolvable identifiers composes URIs, ECLI, OCID, LEI and org-id.guide

Every record has an identifier the publishing institution issues, and a canonical URL that resolves to the record (or to a definitive not-found answer). The identifier's scheme is declared: its namespace, its syntax, and the registry that issues it, using an org-id.guide code where one exists. Identifiers for the same entity in different registries are related by namespace, so that a consumer can state "the entity known to registry X as *a*" without string matching.

Test. For a sample of records, the canonical URL returns the record; a syntactically valid but unassigned identifier returns `found: false` with HTTP 200, not an error; the scheme declaration is present in the schema.

R3 **Uniform query contract** composes OpenAPI 3.1 and the OGC API Features query pattern

The interface offers at least two operations per record type: *lookup* by identifier and *search* by declared fields. Search results are bounded (a declared maximum page size), paged by an opaque cursor, and sorted deterministically. Which fields are searchable is declared in the schema, not discovered by trial. A not-found lookup is an answer, not an error: it is the same shape as a found record with `found` set to `false`. Errors use Problem Details [26].

Test. The OpenAPI document declares both operations; a search with a cursor from a previous page returns the next page and never a duplicate; a request past the page-size bound is rejected with a Problem Details body naming the bound.

R4 **Change feed** composes a monotonic cursor over the record's own timestamps

Every record carries `_updated_at`, the instant the publisher last changed it. Search accepts `updated_after`, returning only records changed since that instant, in ascending order of change, so that a consumer can replay the registry from any point without re-reading it. Deletions are records too: a tombstone with `_deleted_at` appears in the feed.

Test. Two searches with `updated_after` set to $t_1 < t_2$ return sets where the second is a subset of the first; every returned record has `_updated_at` greater than the parameter; the feed is stable under repetition.

R5 **Provenance envelope** composes PROV-O concepts and DCAT licence metadata

Every response wraps its payload in an envelope that states: `source` (the institution, as an org-id.guide identifier where possible), `as_of` (the instant the underlying data was last synchronised or, for a live lookup, the instant of retrieval), `record_version` (a hash of the canonical record, JCS-serialised [39]), and `license` (a URI). The distinction between "last changed" (**R4**) and "last checked" (`as_of`) is deliberate: the first is a fact about the record, the second about the publisher's process, and consumers need both.

Test. Every response carries all four fields; `as_of` is never later than the response time; two responses for the same record with equal `record_version` are byte-identical after canonicalisation.

R6 **Bulk access** composes DCAT distributions and dataset series

The complete set of records of each type is available as a versioned bulk export, described as a DCAT distribution with `dct:modified` and a checksum, and the export is derivable from the change feed (or the feed from the export): the two are two views of one dataset, not two datasets.

Test. The DCAT description is reachable from the interface description; the export's record count equals the count reported by the search operation at the same `as_of`; a random sample of exported records matches the lookup operation.

R7 **Agent surfaces** composes MCP, llms.txt, RFC 9727 api-catalog and content negotiation

The publisher exposes (i) an MCP server whose tools are generated from the OpenAPI document, one tool per operation, with the schema's descriptions as the tool descriptions; (ii) a markdown rendering of every documentation page and of every record, negotiated with `Accept: text/markdown` at the same URL and also available at a `.md` suffix; (iii) an `llms.txt` index at the host root that lists the interface description, the catalog, the documentation and the feed; and (iv) an `api-catalog` well-known resource [43]. A tool description states what a not-found answer means, so that an agent does not retry a definitive negative.

Test. The MCP server's tool list is isomorphic to the OpenAPI operations; every documentation URL answers markdown under negotiation; `/llms.txt` and `/.well-known/api-catalog` resolve and their links resolve.

R8 **Operational contract** composes OpenAPI, RFC 9457 and rate-limit headers

The publisher declares rate limits in response headers, a versioning and deprecation policy with a minimum notice period, an availability commitment and a status page, and a contact. Every error is a Problem Details document with a stable machine-readable `type`. A request that cannot be answered within a declared bound is accepted for asynchronous completion (HTTP 202 with a status URL) rather than left to time out.

Test. Rate-limit headers are present on every response; the deprecation policy is linked from the interface description; the status page resolves; an induced error returns Problem Details.

4.3 Conformance levels

The three levels (table 1) are ordered by what they let a consumer stop doing. At Level 1 the consumer stops parsing HTML and guessing at freshness. At Level 2 it stops downloading the whole registry to answer one question and stops diffing snapshots to learn what changed. At Level 3 it stops needing a developer to write the integration at all. The levels also order the effort for a publisher: Level 1 is achievable from a bulk file and a schema, which most national catalogs already host; Level 2 needs a service; Level 3 is generated from Level 2 by tooling.

Level	Requirements	What a consumer may rely on
Level 1 Readable	**R1**, **R2**, **R5**, **R6**	Records are typed, identified, timestamped and obtainable in bulk. A program can load the registry and trust what it loaded.
Level 2 Queryable	Level 1 + **R3**, **R4**, **R8**	Records can be looked up, searched and followed over time through a documented, bounded, versioned interface.
Level 3 Agent-ready	Level 2 + **R7**	An agent can discover, understand and use the registry with no human in the loop, and cite what it used.

Requirements **R5** (envelope) and **R2** (identifiers) are in Level 1 on purpose: they are the properties whose absence a consumer can least repair after the fact.

Table 1: Conformance levels. Each level is a superset of the previous one; a validator reports the highest level all of whose tests pass.

4.4 The envelope

The envelope is the one concrete wire format the profile introduces, and it is small (listing 1; the JSON Schema is in section B). It separates the record from the statements about the record, so that a domain-standard payload can be embedded unchanged: an OCDS release goes in `data` exactly as OCDS specifies it, and the envelope says where it came from and when.

Listing 1: A lookup response under the profile. The payload is an OCDS release and is not altered by the envelope; `found`, `source`, `as_of`, `record_version` and `license` are the profile's contribution.

```
{
  "found": true,
  "source": { "id": "PE-OECE", "name": "OECE", "org_id": "XI-PB-PE-OECE" },
  "as_of": "2026-08-26T08:15:00Z",
  "record_version": "sha256:9f2c...e1",
  "license": "https://creativecommons.org/licenses/by/4.0/",
  "data": {
    "ocid": "ocds-dgv273-seacev3-873200",
    "_updated_at": "2026-08-21T14:02:11Z",
    "tender": { "title": "...", "status": "complete" }
  }
}
```

A not-found answer has the same envelope with `found: false` and no `data`. This is the single most consequential rule in the profile for agents: a definitive negative that arrives as HTTP 200 is an answer an agent can act on, while a 404 is indistinguishable from a broken URL and will be retried or reported as a failure. The reference implementation's tool descriptions state this explicitly, and it removed an entire class of agent misbehaviour we observed before adopting it.

4.5 Relationship to existing standards

Table 2 lists, for each requirement, the standard it composes and what the profile adds. In every row the addition is a constraint or a field name, never a competing vocabulary. A DCAT-AP catalog can describe a profile-conformant registry as a `dcat:DataService` with its distributions, and the profile's validator can be run from the catalog entry. An OCDS publisher becomes Level 1 by adding the envelope and a schema pointer, and Level 2 by adding `updated_after` to its existing release search, which the OCDS record package model already anticipates. The profile is, in that sense, the layer OCDS and DCAT have been assuming exists.

Req.	Composes	What the profile adds
R1	JSON Schema 2020-12; OCDS, BODS, GTFS, Akoma Ntoso, SDMX	Mandatory descriptions and enumerations; declared mapping to the domain schema
R2	URIs; ECLI, OCID, LEI; org-id.guide	Declared identifier scheme per record type; canonical URL; `found: false` semantics
R3	OpenAPI 3.1; OGC API Features; RFC 9457	Lookup and search as the minimum; bounded cursor paging; declared searchable fields
R4	Record timestamps; Atom-style feeds	`_updated_at`, `updated_after`, tombstones; replayability test
R5	PROV-O; DCAT licence terms; RFC 8785	The envelope: `source`, `as_of`, `record_version`, `license`
R6	DCAT 3 distributions and series	Feed and export as two views of one dataset; count and sample tests
R7	MCP; llms.txt; RFC 9727; HTTP content negotiation	Tools generated from OpenAPI; markdown twins at the same URL; not-found stated in tool descriptions
R8	OpenAPI; RFC 9457; RateLimit header fields	Declared limits, deprecation notice period, status page, asynchronous completion

Table 2: What each requirement composes and what it adds. The profile introduces one wire structure (the envelope) and three field names; everything else is a constraint on how an existing standard is used.

5 Reference implementation

The profile was not designed on paper and then built. It was extracted from a platform that had to expose several dozen registries through one contract and kept meeting the same absences. This section describes that platform at the level of its contract, so that the measurements in section 7 can be read, and so that a second implementer has something concrete to compare against. Product detail and retrieval mechanics are out of scope; the public documentation covers the former [10], and the latter is the publisher's concern under the profile, not the consumer's.

5.1 Scope

At the time of writing the platform exposes 80 endpoints: 75 over 46 official or quasi-official sources in Colombia (38 endpoints), Mexico (25) and Peru (12), and five borderless content endpoints (web search, extraction and research) that are outside this paper's scope. The sources span the registries a lender, a compliance team or a journalist consults in practice: company and taxpayer registers, vehicle and traffic-fine registries, court dockets and case law, procurement, disciplinary and fiscal sanctions, financial statements, official gazettes, regulatory circulars, federal legislation and prosecutors' bulletins. Each endpoint is a *lookup* or a *search* in the sense of **R3**, described in one OpenAPI 3.1 document, and every record type has a JSON Schema with described fields in the sense of **R1**.

5.2 Two source archetypes

Sources fall into two archetypes, and the distinction is the single most important fact about the measurements that follow.

Answered at the source's pace. For most sources, a request is answered by consulting the institution's own system at request time. The platform contributes the contract: a typed record, the envelope, a definitive not-found, a machine-readable error with a stable code on failure, declared rate limits, and asynchronous

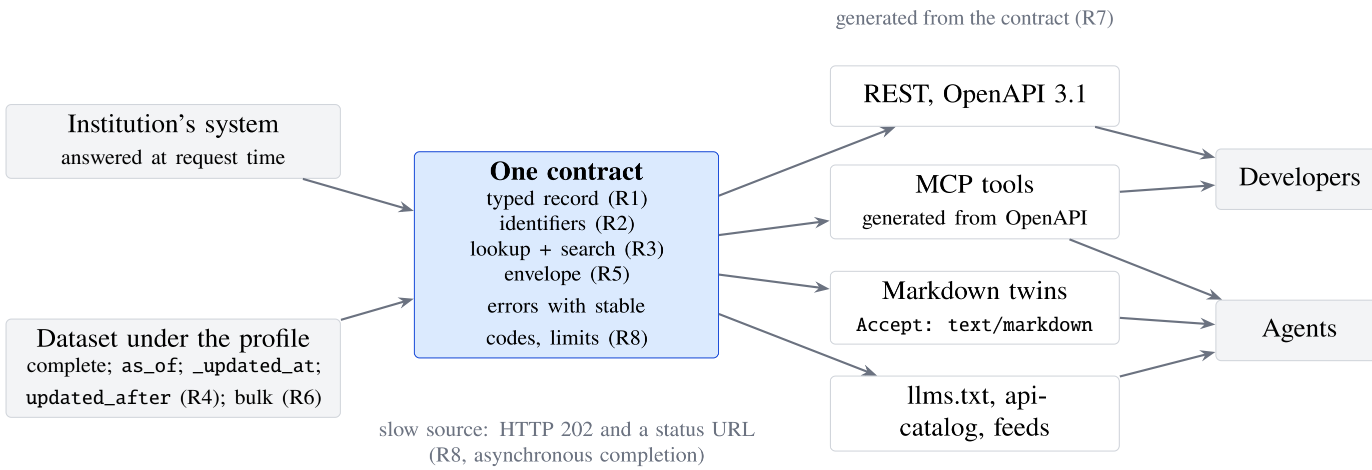


Figure 3: Reference architecture. One contract (schema, envelope, OpenAPI, Problem Details) fronts two archetypes: sources answered at their own pace at request time, and datasets maintained under the profile. The agent surfaces (MCP tools, markdown twins, `llms.txt`, api-catalog) are generated from the contract, never hand-written, so they cannot drift from it.

completion when the source is slow (the request is accepted with HTTP 202 and a status URL, which the consumer polls). What the platform cannot contribute is speed or availability beyond the source's own: if the institution's system takes twenty seconds or is down, the consumer sees twenty seconds or a `source_unavailable` problem. Seventy-two endpoints fell in this archetype during the measurement window.

Answered from a dataset maintained under the profile. For a source whose records can be enumerated, the platform maintains the complete set of records as a dataset: typed by the schema, synchronised on the source's own publication cadence, with every record carrying `_updated_at` and every response carrying `as_of`. Lookups and searches are answered from the dataset. Search accepts the fields the domain cares about rather than the ones the source's form happens to offer, and `updated_after` returns only what changed. During the window this archetype covered Peru's public procurement (OECE, published by the institution under OCDS) and, from the last day of the window, the 311,878 tesis of Mexico's Supreme Court. This archetype is what a Level 2 publisher looks like from the outside; the platform stands in for the institution until the institution publishes under the profile itself.

5.3 The contract as implemented

Figure 3 shows the shape. The elements that correspond to the profile's requirements are:

- **Records and identifiers (R1, R2).** Every endpoint's response schema is a typed record with described fields; identifiers are the institution's own (NIT, RUC, radicado, OCID, registration number). A syntactically valid identifier with no record returns `found: false` with HTTP 200. The documentation states, per endpoint, that this is a definitive answer and not an error; the MCP server's instructions repeat it.
- **Query (R3).** Lookup by identifier and search by declared fields, with bounded page sizes. Errors are a fixed envelope with a stable machine-readable code. Batch variants exist where consumers asked for them.
- **Change (R4).** Dataset-backed endpoints accept `updated_after` and expose `_updated_at`. Source-paced endpoints cannot offer it, because the source does not; this is the requirement that most clearly separates the

archetypes.

- **Envelope (R5).** Every endpoint is bound to one named source in its description. Dataset-backed responses carry `as_of` and the record's own timestamp; source-paced responses are answered at retrieval time and do not yet carry an explicit `as_of`. Responses may be reused within a declared freshness window, and a reused response says so in a header.
- **Agent surfaces (R7).** One MCP server exposes every endpoint as a tool, with tool descriptions generated from the same schema descriptions the OpenAPI document carries. Every documentation page and every changelog entry answers markdown at its own URL under content negotiation and at a `.md` suffix; the site publishes `llms.txt`, an `api-catalog` well-known resource, and Link headers pointing at all of them.
- **Operations (R8).** Rate limits are declared in headers, with a daily policy per key; a versioning and deprecation policy is published; a status page exists; slow sources are answered asynchronously.

5.4 What the implementation does not yet satisfy

The platform is not a full Level 3 publisher and this paper does not claim it is. Bulk export (**R6**) exists for the dataset archetype only; source-paced endpoints cannot offer it by construction. The envelope (**R5**) is complete only for dataset-backed responses: source-paced responses lack an explicit `as_of`, and `record_version` (a canonical hash) is not yet emitted by either. Errors carry stable codes in a fixed envelope but are not yet RFC 9457 Problem Details (**R3**, **R8**). Cross-registry identifier namespaces (**R2**, the org-id.guide declaration) are documented per endpoint rather than declared in the schema. Each of these is scheduled, and each is a smaller change than the ones already made; we list them because a reference implementation that overstated its conformance would undermine the validator it argues for.

6 A publishing harness for institutions

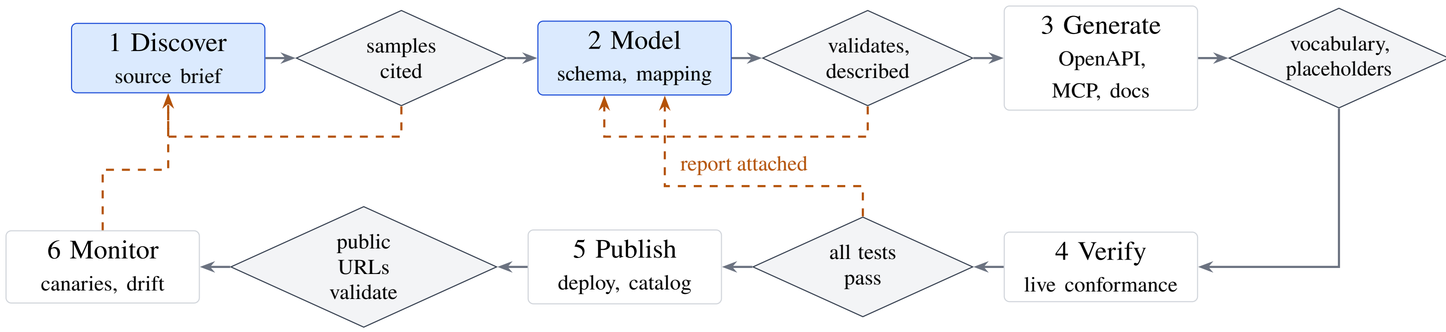
The profile says what a conformant registry looks like. This section is about how an institution gets there, and specifically about how a language model can do most of the work without being trusted to decide whether the work is right. The design comes from our own onboarding loop, which has taken several dozen sources from "a system with a data dictionary" to "a conformant, documented, agent-visible endpoint", and which changed shape once we stopped asking the model to build things and started asking it to propose things for a fixed set of gates to accept or reject.

6.1 The shape of the problem

An institution that wants to publish a registry under the profile starts from one of three positions: a database or internal API with a data dictionary (the common case), an existing public export or catalog entry (a bulk CSV on the national portal), or an existing public interface that predates the profile (a search form, a SOAP service). In all three, the expensive steps are the same: understanding the record, writing the schema and its descriptions, deciding the identifier scheme, mapping onto a domain standard where one applies, generating the interface and its documentation, and verifying that all of it is true of the live system. The first four are language tasks over documents the institution already has; a model does them well and fast. The last is not a language task, and it is where errors become consequences.

6.2 The pipeline

Figure 4 shows six stages.



Shaded stages: the model proposes. Diamonds: machine-checkable gates. Dashed: a failed gate returns its report to the stage that proposed, and monitoring re-enters the loop at discovery.

Figure 4: The publishing harness. Six stages; the model proposes in the first three and the last is monitoring. Between stages sit gates that are machine-checkable conditions on artifacts, not judgements about text. A proposal that fails a gate goes back to the model with the failure attached; nothing reaches the publish stage that has not passed every gate, and a human reviews the gate reports, not the model's prose.

1. Discover. Inputs: the institution's data dictionary, sample records, existing documentation, and (if one exists) the current public interface. The model produces a *source brief*: the record types present, the identifier each carries and who issues it, the fields and their apparent types and code lists, the update cadence as far as it can be inferred, and the not-found behaviour of the existing system. The brief is a document for people; it is also the input the next stage is held to.

2. Model. The model proposes a JSON Schema per record type with a description for every field, an identifier-scheme declaration, and, where a governed domain schema applies, a field-level mapping to it (this procurement record's `fecha_adjudicacion` is OCDS `awards[].date`). The proposal is annotated: every description cites the dictionary entry or sample it came from. Gate: the schema validates the samples; every field has a description; the mapping's target fields exist in the domain schema's own definition; enumerations cover every value seen in the samples.

3. Generate. From the accepted schema the harness generates, deterministically, the OpenAPI document, the envelope wrapper, the Problem Details vocabulary, the MCP tool manifest, the markdown documentation and the `llms.txt` entry. The model's role is limited to prose the generator cannot write: worked examples and the plain-language description of the registry. Gate: the generated artifacts are consistent with each other by construction (they share a source); the model's prose is checked against a fixed vocabulary that excludes anything the institution does not want published, and against placeholder rules for personal identifiers in examples.

4. Verify. The conformance tests from section 4.2 run against the live interface: lookups by known identifiers, a not-found probe with a valid-but-unassigned identifier, cursor paging, `updated_after` replay, envelope fields, rate-limit headers, the markdown negotiation, the MCP tool list against the OpenAPI operations. Gate: every test for the target level passes. This gate is the harness's centre of gravity: it cannot be argued with, it produces the report a human reads, and it is what the model is trying to satisfy in the earlier stages.

5. Publish. The artifacts are deployed; the catalog entry (DCAT) is written or updated to point at the interface; the change is announced through the feed. Gate: the published URLs resolve and the validator

Stage	The model may	The model may not
Discover	Read documents; propose record types, identifiers, cadence; flag ambiguities	Probe production systems beyond what the institution authorises; infer fields that no sample supports
Model	Propose schemas, descriptions, code lists, domain mappings, with citations to the dictionary	Invent a field, a value or a mapping without a cited source; change a stable field name
Generate	Write examples and plain-language descriptions	Write the OpenAPI, envelope, MCP manifest or docs by hand; use real personal identifiers in examples
Verify	Read the gate report and revise its proposal	Modify a test, its threshold, or its input; declare a test passed
Publish	Draft the announcement	Deploy; edit the catalog entry
Monitor	Diagnose a drift report and propose a fix	Suppress a canary; change a schema in place

Table 3: The model's authority by stage. The pattern is uniform: the model proposes over documents and revises against reports; the harness generates and verifies; people approve gate reports.

passes against the public address, not the staging one.

6. Monitor. The conformance tests run continuously as canaries; the record schema is checked against a sample of live records for drift (a field that changed type, a new enumeration value, a changed not-found shape); freshness is checked against the declared cadence. A failure opens a report that re-enters the loop at the stage it implicates.

6.3 What the model may and may not do

Table 3 states the boundary. The principle behind it is that a model is a good author and a poor witness: it can write a schema that is plausible, complete and well-described far faster than a person, and it cannot be relied on to know whether the schema is true of the system. The harness therefore never lets the model be the last step. Every artifact that ships is either generated deterministically from something a gate accepted, or is prose that a fixed vocabulary check passed. The tests are the institution's, not the model's; a proposal that fails is returned with the failure attached, and the model revises. This is the same loop a careful engineer runs; the difference is that the loop is explicit, the gates are named, and the model cannot skip one.

6.4 Observations from use

Three things from our own use of this loop are worth recording, with the caveat that they are an operator's observations and not a controlled study.

First, the binding constraint moved. Before the gates were explicit, the slow step in adding a source was writing and reviewing the connector; after, it was the verification step waiting on the institution's system, which is to say the source's own pace and availability. The engineering effort per source fell from days to hours; the elapsed time did not fall as much, for reasons the source controls.

Second, the gates caught what review did not. The three failures the verify stage catches most often are a not-found shape that differs from the documented one (the system returns an empty list for one identifier format and an error for another), a field whose sample values were all of one type but whose live values are not, and a page size that the documentation states and the system does not honour. Each of these had, before the gates, reached consumers.

Third, the vocabulary gate matters more than it looks. An institution has things it does not want said about its systems, and a model writing documentation will say them, helpfully and accurately, unless prevented. A fixed list of excluded terms, applied to every generated sentence, is a blunt instrument and the right one: it is the one check that has never had a false negative.

6.5 Cost

For an institution the marginal cost of a registry under the harness is dominated by two things it already pays for: the people who know the data, and the system that serves it. The model's contribution is bounded by the number of fields (descriptions and mappings) and is cheap at current prices; the generator's is fixed; the verifier's is proportional to the number of tests, which is small and does not grow with the registry. The cost that does not fall is institutional: deciding that a registry is public, that its identifiers are stable, that its not-found answer is definitive, and that it will stay up. The harness makes those decisions visible and checkable. It does not make them.

7 Evaluation

We evaluate the claim that the consumption gap is where open data fails, and the profile's fitness to close it, with two kinds of evidence: a documentary assessment of 37 official sources against criteria derived from the profile, and production measurements from the reference implementation over 80 days.

7.1 Source assessment

Method. We selected the 37 sources behind the reference implementation's endpoints in Colombia (20), Peru (8) and Mexico (9): the registries a consumer in those countries actually asks for, rather than a sample of catalog entries. Each was assessed on 26 August 2026 against thirteen criteria (table 4) derived from the seven gap properties and the eight requirements, using only public documentation: the institution's site, the national open-data portal, published data dictionaries and reputable press. Each cell is one of *yes*, *partial*, *no* or *unknown*, with a note and an evidence URL; *unknown* was used whenever documentation could not establish the answer, and never inferred. Assessors did not probe any system beyond reading documentation. The full table with evidence is in section C and in the paper's data files. Confidence is recorded per source; eight sources are at medium or low confidence because official pages could not be fetched (certificate or access errors) and the assessment leaned on search excerpts.

Results. Figure 5 shows every source against every criterion, and fig. 6 the totals per criterion. Four findings stand out.

Programmatic access is a minority property, and where it exists it is inherited. Six of 37 sources (16%) document an API for the assessed service and a further six a partial one. Of the six, four are Colombian datasets on datos.gov.co, whose Socrata platform provides a query API, field filters, paging and a system `:updated_at` column to every dataset it hosts; one is Peru's OECE, which publishes procurement under OCDS with a release API and yearly bulk files; one is Mexico's official gazette, the DOF, whose SIDOF system documents JSON services and a status dashboard. No institution in the sample designed and documented a per-record API for its own lookup service; every *yes* on **C1** is a property of a platform or a domain standard the institution adopted. The four top-scoring sources (SECOP, Procuraduría, Supersociedades, RUES; 9.5 to 11.5 of 13) are exactly the four whose data is on datos.gov.co, and the fifth (OECE, 9.5) is the OCDS publisher. This is the strongest single piece of evidence for the profile's composition principle: the way to get a registry to Level 2 is to attach it to something that already has the properties, not to ask the institution to invent them.

Change and freshness are the rarest capabilities. One source (SECOP, via Socrata's `:updated_at`) offers a documented way to retrieve only what changed; nineteen offer something partial (a dataset-level

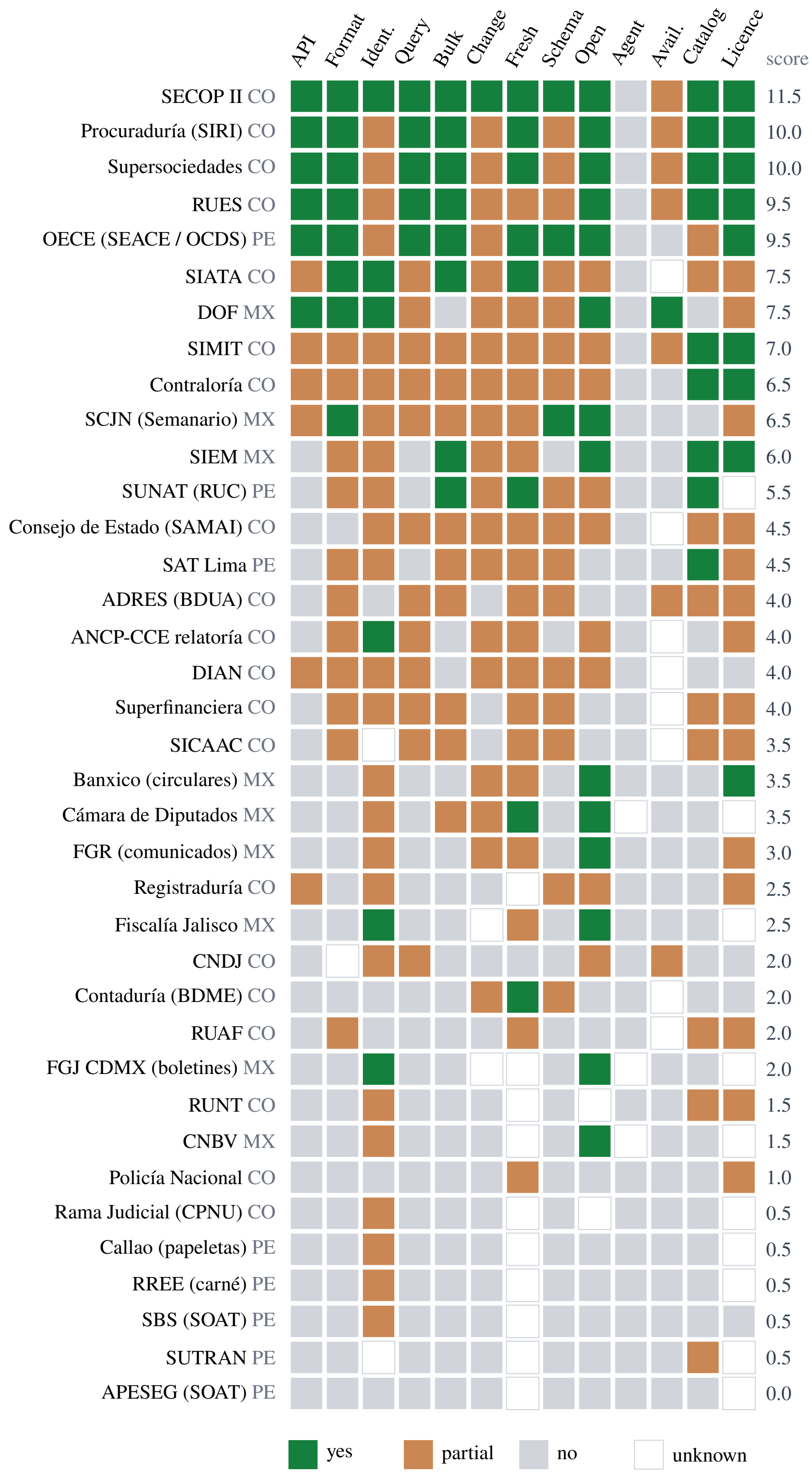


Figure 5: Thirty-seven official sources in Colombia, Peru and Mexico against the thirteen criteria, from public documentation on 26 August 2026. Sources are ordered by score (yes = 1, partial = 0.5). The four sources at the top publish through a national catalog platform with a query API (Socrata's SODA on datos.gov.co) or under a governed domain standard (OCDS); the block at the bottom is single-record lookup forms. Column **C10**, agent surfaces, is empty.

	Criterion	A source scores *yes* when
C1	API	the institution (or the national portal on its behalf) documents a programmatic interface returning machine-readable records
C2	Format	records are obtainable as JSON, XML or CSV rather than only HTML or PDF
C3	Identifier	each record has a stable published identifier resolving to a permanent URL
C4	Query	programmatic filtering by field with pagination, beyond a single lookup form
C5	Bulk	the complete dataset can be downloaded
C6	Change	consumers can retrieve only what changed since a point in time
C7	Freshness	records or responses carry an explicit last-updated or as-of timestamp
C8	Schema	a data dictionary, JSON Schema or OpenAPI document is published
C9	Open access	programmatic access needs no per-session human verification or personal login
C10	Agent surface	an llms.txt, an MCP server, a markdown rendering or structured data for AI exists
C11	Availability	a published availability commitment or status page exists
C12	Catalog	the service or dataset is listed in the national catalog with DCAT-style metadata
C13	Licence	explicit open licence or reuse terms are stated

Table 4: Assessment criteria. **C1** to **C4** and **C6** to **C8** map onto **R1** to **R5**; **C5** onto **R6**; **C10** onto **R7**; **C9**, **C11** and **C13** onto **R8**; **C12** onto the catalog layer the profile composes.

last-modified date, or a date field in the record that a consumer can filter on with care); fifteen offer nothing. Record-level freshness (**C7**) is documented by eight, partially by eighteen, and unknown for ten, the highest *unknown* count of any criterion, which is itself a finding: the documentation does not say when the answer was true. These are the two requirements (**R4**, **R5**) that a consumer can least work around, and they are the two a platform like Socrata supplies for free and an institution's own form never does.

Agent surfaces are absent. No source in three countries publishes an `llms.txt`, an MCP server, a markdown rendering or any structured data meant for AI consumption; probes returned 404s or application shells. The three *unknown*s are hosts that could not be fetched. This matches the state of the field in section 2 and is the empty column the profile's **R7** is written to fill. It is also the cheapest column to fill: everything in **R7** is generated from the OpenAPI document and the schema, so a Level 2 publisher is one build step from Level 3.

The bottom of the table is the interactive form. Eight sources score at or below one point: the vehicle-fine and insurance lookups in Peru, the police and court dockets in Colombia, and a state prosecutor's bulletin in Mexico. Each is a page where a person types one identifier and reads one answer, with no API, no bulk, no schema, no timestamp and, for several, a per-session human verification step. These are among the most consulted registries in their countries. They are also, from the consumer's side, the sources with the highest latency and failure rates in section 7.2. The two facts are the same fact.

Per country, Colombia scores highest on average because of the catalog platform; Mexico has the most open access (**C9**: nine of nine) but the fewest query interfaces; Peru has the single best publisher (OECE) and the weakest tail. Licence terms (**C13**) are explicit for nine sources, partial for fourteen and unknown for ten; three national portals default to CC BY or CC BY-SA at the dataset level, which is what the *yes* and *partial* cells mostly reflect.

7.2 Production measurements

Method. The reference implementation logs, for every authenticated request, the endpoint, the HTTP status, the end-to-end duration, whether the response was reused within its freshness window, and an error code when the status is 400 or above. No request or response bodies are logged. We report the window from 8 June to 26 August 2026 (80 days): 25,153 requests, of which 20,598 answered 200, 922 answered 202 (accepted

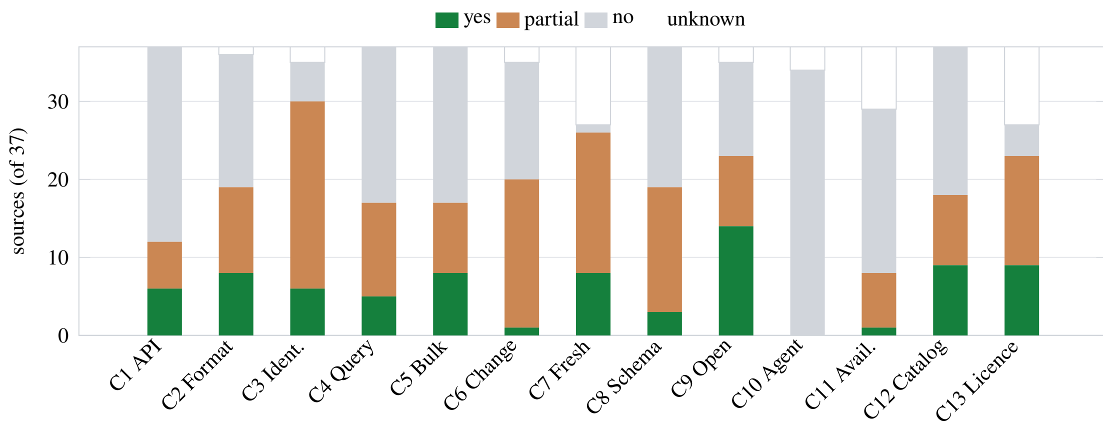


Figure 6: Per-criterion totals across the 37 sources. Change feeds (**C6**) and agent surfaces (**C10**) are the rarest capabilities; freshness (**C7**) is mostly partial (a dataset-level date rather than a record-level one) or unknown; identifiers (**C3**) are mostly partial (a stable URL pattern that the institution does not document as such).

Class	Successful requests	p50 (ms)	p90 (ms)	p95 (ms)
Source-paced lookups, all	17,660	2,067	15,533	26,185
answered by the source	14,772	2,593	18,526	27,922
reused within freshness window	2,888	100	322	434
Dataset under the profile (OECE)	84	75	248	366
Asynchronous status polls	1,655	81	163	274

Server-side durations, HTTP 200 only, 8 June to 26 August 2026. Source-paced lookups additionally produced 922 asynchronous acceptances (202) and 1,225 upstream failures (500 to 503).

Table 5: Latency by class. The same contract answers in 75 ms at the median from a dataset maintained under the profile and in 2.6 s at the median, 27.9 s at the 95th percentile, when the source answers at its own pace.

for asynchronous completion), 1,833 answered 500 to 503 (the source failed or was unavailable), and 1,800 answered a client error (invalid input, unauthorised, or rate-limited). Durations are server-side, from request receipt to response, so they exclude the consumer's network but include the source's. We classify endpoints into the two archetypes of section 5.2: 72 source-paced lookup endpoints with at least one success, and the dataset-backed OECE endpoints (the SCJN dataset went live on the last day of the window and contributes two requests, which we exclude). Status polls for asynchronous jobs and the five content endpoints are reported separately and not used in the comparison.

Results. Table 5 and fig. 7 give the distributions.

The consumer inherits the source's variance. Across the 72 source-paced endpoints, the median latency of the median request was 2.6 s and the 95th percentile 27.9 s; 636 successful requests (4.3%) took longer than 30 s. Per endpoint (fig. 8), medians range from a quarter of a second (an environmental sensor network that publishes a documented data API) to 24 s (a police certificate service that is an interactive form), with six of 52 endpoints above 20 s at the median. These are the same registries that sit at the bottom of fig. 5. Nothing in the contract can hide this; asynchronous completion (922 requests) makes it survivable for the consumer but does not make it fast.

A dataset under the profile removes it. The 84 successful requests answered from the OECE dataset had a median of 75 ms and a 95th percentile of 366 ms, thirty-five times faster at the median than source-paced

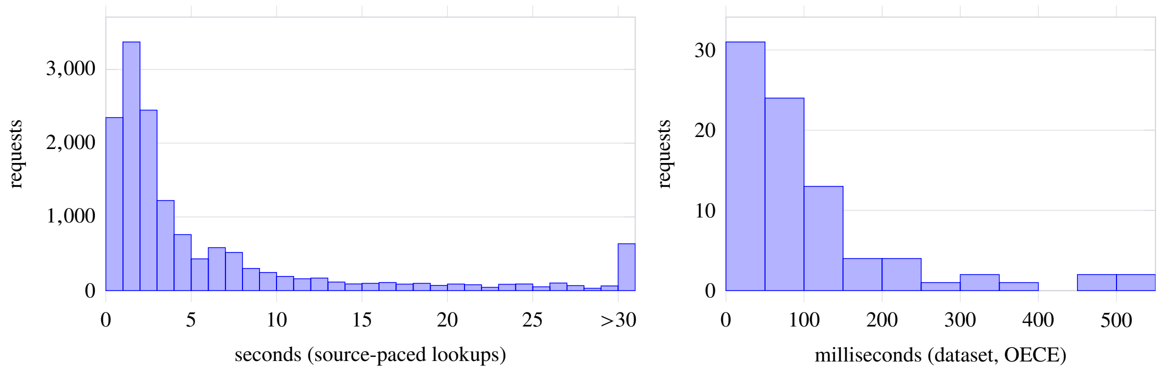


Figure 7: Left: distribution of server-side latency for successful source-paced lookups answered by the source (n = 14,772), in one-second bins; the last bin collects everything above 30 s (636 requests, 4.3%). Right: the same for lookups answered from the OECE dataset (n = 84), in 50 ms bins. The axes differ by two orders of magnitude.

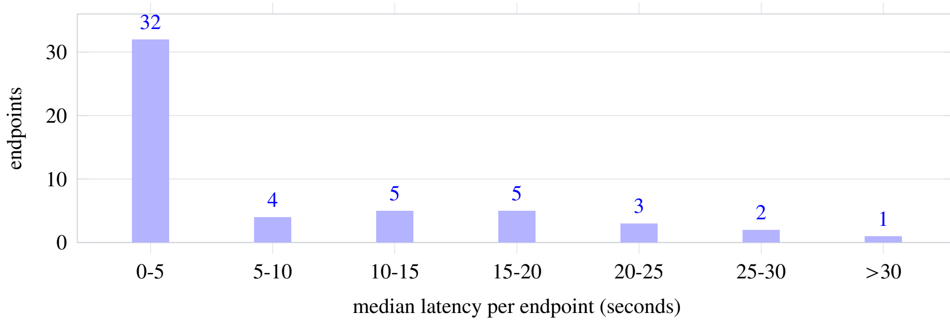


Figure 8: Median latency per source-paced endpoint, for the 52 endpoints with at least ten successful source-answered requests. Thirty-two endpoints answer in under five seconds at the median; twenty take longer, and six take more than twenty seconds. The consumer of a uniform contract inherits this spread from the sources.

lookups and seventy-five times at the tail, with no upstream failures. Responses reused within their freshness window (2,888) show the same order of magnitude (100 ms median). The point is not that a database is faster than a web form; it is that the profile's **R4** and **R5** are what make a maintained dataset *legitimate*: a consumer can see `as_of`, can ask `updated_after`, and therefore does not need to go to the source for every question. A dataset without those two fields is a stale copy; with them it is a publication.

Availability is the source's. Of 25,153 requests, 1,833 (7.3%) failed with a 5xx status; the top error codes name the source (procurement, disciplinary records, company register, court docket, fiscal records) or a content provider. Against the source-paced successes alone, the upstream failure rate was 6.5%; against the dataset, zero. One request in fourteen is the cost, to a consumer, of a registry that has no availability commitment (**C11**: one *yes* in 37), and it is a cost the consumer pays in retries, timeouts and, for agents, in wrong conclusions drawn from a failed call.

7.3 Threats to validity

The source assessment is documentary. It measures what institutions *publish about* their services, which is the right object for a profile whose Level 3 is about discoverability, but it can under-state capability that exists undocumented and over-state capability that is documented and broken. Ten cells are *unknown* for the freshness criterion alone. Eight sources are at medium or low confidence because official hosts refused automated fetches, an irony we note. The rubric was applied by one assessor per country with a fixed instruction set; a second assessor would move some *partial* cells in both directions, and we publish the evidence so that they can.

The production measurements are from one operator's traffic, shaped by that operator's customers (predominantly Colombian financial-services consumers, hence the weight of Colombian endpoints) and by its own engineering. The dataset class is small ($n = 84$) and covers one source; the comparison is between archetypes, not a controlled experiment, and the difference is large enough that the small sample does not threaten the direction of the result. Durations exclude the consumer's network. Requests that were rate-limited (1,316) or malformed are excluded from latency but included in the totals.

Finally, the author operates the reference implementation and has an interest in the profile's adoption. We have tried to make that a strength rather than a weakness by reporting what the implementation does not satisfy (section 5) and by publishing every number in this section with the data behind it.

8 Adoption and governance

A profile that nobody publishes under is a paper. This section sets out how Agora could be adopted, drawn from the two domain standards that achieved global uptake and adjusted for the institutions of the region where the reference implementation runs.

8.1 Lessons from GTFS and OCDS

GTFS began in 2005 as a data exchange between one transit agency, TriMet in Portland, and one consumer, Google Maps; it was a handful of CSV files that solved routing, and it spread because developers built applications on it and agencies wanted to be in them [23]. Its name changed from *Google* to *General* Transit Feed Specification in 2010, which its stewards credit with reducing resistance from agencies and vendors wary of one company owning their data format. OCDS launched in 2014 with a JSON schema, a validator and a helpdesk, stewarded by a non-profit funded by philanthropy, and it spread through flagship publishers whose data others wanted to compare [30]. Both standards stayed small, both shipped tooling before they sought mandates, and both moved to a neutral steward before they were widely adopted. Neither began as a specification looking for an implementation. We take these as the three conditions for the profile.

8.2 Stage one: reference implementation and validators

The profile's first form is running software, not a document: the reference implementation, a conformance validator that any publisher can run against any interface, and a badge the validator issues per level. The validator is the specification (section 4.1); publishing it first keeps the prose honest. In this stage the profile adopts every relevant domain standard on output, so that an OCDS or ECLI publisher sees the profile as a way of being read by agents rather than as a competitor, and it publishes the agent surfaces (**R7**) for every source it covers, because those are the surfaces with no government competition today.

8.3 Stage two: a minimal open specification

Once the validator has run against a dozen publishers and the requirements have survived contact with them, the profile is written down as a specification: the eight requirements, the envelope schema, the test definitions, and the composition table, under CC0 or CC BY 4.0, with the reference implementation and validator under an OSI licence. The specification stays at the altitude of section 4.2: it names what a validator can check and nothing else. Proposals to add quality, completeness or semantics requirements are, by policy, redirected to the domain standards where they belong.

8.4 Stage three: neutral stewardship

A standard controlled by one company will not be adopted by governments, and should not be. The profile's stewardship passes to a working group with publishers, consumers and civil-society members, hosted by a body governments already trust. In Latin America the candidates exist: the Latin American Open Data Initiative (ILDA), which co-runs the regional OCDS helpdesk; Red GEALC, the digital-government network of the Organization of American States; the Inter-American Development Bank's digital-government programmes; and the national digital-government offices that already run the catalogs (MinTIC in Colombia, PCM's Secretaría de Gobierno y Transformación Digital in Peru, the ATDT in Mexico). The profile's name is neutral by design and carries no operator's brand.

8.5 Demand first

GTFS and OCDS were pulled by consumers, not pushed by mandates. The profile's consumers are already present: lenders, insurers, compliance and KYC teams, journalists, civic technologists and, now, every agent framework that can read an MCP tool list. The adoption strategy is therefore to make those consumers visible to publishers, through the reference implementation's usage and through the validator's badge, so that an institution sees a queue of applications waiting on its registry rather than a request to do work.

8.6 Alignment with mandates

Where mandates exist, the profile is the form of compliance. The European High-Value Datasets regulation requires APIs and bulk download for company registers and mobility data [12]; Level 2 satisfies both clauses and adds what the regulation leaves unspecified (identifiers, change, provenance). The OPEN Government Data Act requires inventories [47]; the profile's DCAT description is one. In Colombia, Peru and Mexico the transparency laws and open-data decrees require publication and catalogs [1, 7, 37]; the profile gives the catalogs something to point at. Mexico's 2025 open-data guidelines for the federal administration are the most recent instrument in the region and the natural place for a Level 1 requirement to appear.

8.7 What would change this strategy

Three developments would. If a government or the European Union publishes an official cross-registry access or agent standard, the profile's work becomes conformance and tooling for that standard rather than a parallel specification. If DCAT-AP or DCAT-US adds record-level query and change semantics, the profile becomes a DCAT-AP extension. If OCDS, BODS or Akoma Ntoso coverage in the region deepens to the point where most registries are published under a domain standard, the profile's value concentrates in **R4**, **R5** and **R7**, the parts those standards do not cover, and it should shrink accordingly.

9 Limitations and future work

The profile is untested outside its author's implementation. Every requirement has been exercised by one operator against several dozen sources in three countries. It has not been implemented by a government, and the harness has not been run inside an institution on its own systems. The two are the obvious next steps, and the first pilot will change the profile.

Entity resolution is declared, not solved. **R2** asks publishers to declare identifier namespaces so that consumers can relate records across registries without string matching. It does not resolve entities: the profile deliberately stops at "registry X calls this *a*" and leaves "*a* in X is *b* in Y" to a resolution service, because a wrong merge is worse than no merge and the property is not testable from outside. A profile for resolution services, composed over LEI and org-id.guide, is future work.

Personal data. Many of the registries in the assessment concern people: disciplinary records, criminal certificates, vital status, affiliations. The profile's bulk requirement (**R6**) and change feed (**R4**) are not appropriate for every such registry, and a conformant publisher of personal records may legitimately stop at lookup-by-identifier with rate limits and purpose declarations. The profile should acquire a personal-data annex that says which requirements a lawful basis relaxes, written with data-protection authorities rather than for them.

The assessment is a snapshot by three assessors. Its value is in the evidence trail and the rubric, which we publish so that it can be re-run, disputed and extended. A community-maintained assessment with a second rater per cell would be a better instrument.

The production evidence is one operator's. The dataset class is small and the traffic mix is that operator's customers. A second implementer's numbers would be the most useful thing a reader could contribute.

Agents as evaluators. The strongest test of Level 3 is whether an agent, given a question and the surfaces in **R7**, answers it correctly and cites the record. We have observed this qualitatively and have not measured it. A benchmark of registry questions with ground truth, run against publishers at each level, is the evaluation this paper should have had and the next one will.

10 Conclusion

The open-data movement built the layers it set out to build. Governments publish; catalogs describe; domain standards encode. What it did not build, because no single layer owned it, is the contract a consumer needs to use one record: resolve it, query it, follow it, trust its timestamp, and do so from a program. That gap was tolerable while the consumer was a person with a browser and time. It is not tolerable for an agent, and agents are now a measurable share of the traffic public data receives.

Agora is a small answer to a large gap. Eight requirements, three levels, one envelope, and a validator, composed over the standards that already exist rather than beside them. The reference implementation shows that the contract can be held uniformly across 46 sources that share nothing else, and the production measurements show what the contract cannot hide: a registry answered at a web form's pace is thirty-five times slower at the median and seventy-five times slower at the tail than the same registry published under the profile, and fails one time in fourteen. The assessment of 37 sources shows that the properties the profile asks for are rare, that where they exist they were inherited from a platform or a domain standard, and that the agent surfaces are missing everywhere.

The harness is the part we expect to matter most in practice. The cost of writing a typed, documented, agent-visible interface over a registry has collapsed; the cost of being wrong has not. Letting a model propose and a fixed set of gates dispose is how an institution can have the first without paying the second. The gates are the profile's tests. That is the whole design: a specification small enough to be a validator, a validator strict enough to be a gate, and a loop in which the machine that writes is never the machine that decides.

Acknowledgements

The production measurements in this paper come from Croma's public API, which the author operates. The source assessment was carried out from public documentation only. We thank the institutions whose records make the work possible, and the developers whose questions shaped the profile. Errors are mine.

References

[1] Agencia de Transformación Digital y Telecomunicaciones (México). Acuerdo por el que se expiden los Lineamientos en materia de Datos Abiertos de la Administración Pública Federal. Diario Oficial de la Federación, 11 de septiembre de 2025, 2025. URL https://dof.gob.mx/nota_detalle.php?codigo=5767742&fecha=11/09/2025.

[2] Anthropic. Donating the Model Context Protocol and establishing the Agentic AI Foundation. https://www.anthropic.com/news/donating-the-model-context-protocol-and-establishing-of-the-agentic-ai-foundation, December 2025. 9 December 2025. Reports more than 10,000 active public MCP servers and 97M+ monthly SDK downloads. Accessed 2026-08-26.

[3] Judie Attard, Fabrizio Orlandi, Simon Scerri, and Sören Auer. A systematic review of open government data initiatives. *Government Information Quarterly*, 32(4):399–418, 2015. doi:10.1016/j.giq.2015.07.006.

[4] Australian Government. API Design Standard. https://api.gov.au/, 2024. Accessed 2026-08-26.

[5] Tim Berners-Lee. Linked Data. https://www.w3.org/DesignIssues/LinkedData.html, 2006. W3C Design Issues note, 27 July 2006; the five-star open data scheme was added in 2010. Accessed 2026-08-26.

[6] Cloudflare. The 2025 Cloudflare Radar Year in Review: the rise of AI, post-quantum, and record-breaking DDoS attacks. https://blog.cloudflare.com/radar-2025-year-in-review/, December 2025. AI bots accounted for 4.2% of HTML request traffic (Googlebot 4.5%); Anthropic crawl-to-refer ratio up to 500,000:1. Accessed 2026-08-26.

[7] Congreso de la República de Colombia. Ley 1712 de 2014, por medio de la cual se crea la Ley de Transparencia y del Derecho de Acceso a la Información Pública Nacional y se dictan otras disposiciones. Diario Oficial No. 49.084, 6 de marzo de 2014, 2014. URL http://www.secretariasenado.gov.co/senado/basedoc/ley_1712_2014.html.

[8] Congreso de la Unión (México). Decreto por el que se expiden la Ley General de Transparencia y Acceso a la Información Pública; la Ley General de Protección de Datos Personales en Posesión de Sujetos Obligados; la Ley Federal de Protección de Datos Personales en Posesión de los Particulares; y se reforma el artículo 37, fracción XV, de la Ley Orgánica de la Administración Pública Federal. Diario Oficial de la Federación, 20 de marzo de 2025, 2025. URL https://dof.gob.mx/nota_detalle.php?codigo=5752569&fecha=20/03/2025.

[9] Council of the European Union. Council conclusions inviting the introduction of the European Case Law Identifier (ECLI) and a minimum set of uniform metadata for case law. Official Journal of the European Union, C 127, 29 April 2011, pp. 1–7 (2011/C 127/01), 2011. URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52011XG0429(01).

[10] Croma. Croma API documentation. https://docs.usecroma.com, 2026. Public API over official records in Colombia, Peru and Mexico. Accessed 2026-08-26.

[11] Data Package Working Group. Data Package (v2) is released! https://datapackage.org/blog/2024-06-26-v2-release/, June 2024. Frictionless Data, 26 June 2024. Accessed 2026-08-26.

[12] European Commission. Commission Implementing Regulation (EU) 2023/138 of 21 December 2022 laying down a list of specific high-value datasets and the arrangements for their publication and re-use. Official Journal of the European Union, L 19, 20 January 2023, 2023. URL https://eur-lex.europa.eu/eli/reg_impl/2023/138/oj. Applicable from 9 June 2024.

[13] European Parliament and Council of the European Union. Directive (EU) 2019/1024 of the European Parliament and of the Council of 20 June 2019 on open data and the re-use of public sector information (recast). Official Journal of the European Union, L 172, 26 June 2019, pp. 56–83, 2019. URL https://eur-lex.europa.eu/eli/dir/2019/1024/oj.

[14] European Parliament and Council of the European Union. Regulation (EU) 2024/903 of the European Parliament and of the Council of 13 March 2024 laying down measures for a high level of public sector interoperability across the Union (Interoperable Europe Act). Official Journal of the European Union, L, 2024/903, 22 March 2024, 2024. URL https://eur-lex.europa.eu/eli/reg/2024/903/oj. Entered into force 11 April 2024; applies from 12 July 2024.

[15] Miranda Evans and Kadie Armstrong. Pausing development of the Beneficial Ownership Data Standard. https://www.openownership.org/en/blog/pausing-development-of-the-beneficial-ownership-data-standard/, November 2025. Open Ownership, 18 November 2025. Accessed 2026-08-26.

[16] Federal Chief Data Officers Council and Federal Committee on Statistical Methodology and Data.gov (GSA). DCAT-US Schema v3.0. https://resources.data.gov/resources/dcat-us3/, 2026. Implementation guidance marked as draft as of 2026-05-07; v1.1 data.json remains harvested during the transition. Accessed 2026-08-26.

[17] Global Legal Entity Identifier Foundation. ISO 17442: The LEI Code Structure. https://www.gleif.org/en/about-lei/iso-17442-the-lei-code-structure, 2026. Accessed 2026-08-26.

[18] Government Digital Service and Central Digital and Data Office. API design guidance. https://www.gov.uk/government/collections/api-design-guidance, 2024. UK Government collection. Accessed 2026-08-26.

[19] Jeremy Howard. The /llms.txt file. https://llmstxt.org/, September 2024. Published 3 September 2024. Accessed 2026-08-26.

[20] International Organization for Standardization. ISO 19168-1:2025 Geographic information – Geospatial API for features – Part 1: Core. Technical report, ISO, January 2025. URL https://www.iso.org/standard/86508.html. Edition 2; replaces ISO 19168-1:2020. Identical to OGC API - Features - Part 1: Core (OGC 17-069r4).

[21] Marijn Janssen, Yannis Charalabidis, and Anneke Zuiderwijk. Benefits, adoption barriers and myths of open data and open government. *Information Systems Management*, 29(4):258–268, 2012. doi:10.1080/10580530.2012.716740.

[22] Ministerio de Tecnologías de la Información y las Comunicaciones (Colombia). Lenguaje Común de Intercambio de Información. https://lenguaje.mintic.gov.co/, 2026. Accessed 2026-08-26.

[23] MobilityData. General Transit Feed Specification (GTFS). https://gtfs.org/, 2026. States that over 10,000 agencies in 100+ countries use GTFS; renamed from Google Transit Feed Specification in 2010 (see https://gtfs.org/about/). Accessed 2026-08-26.

[24] Model Context Protocol. Model Context Protocol Specification, revision 2026-07-28. Technical report, Agentic AI Foundation, The Linux Foundation, July 2026. URL https://modelcontextprotocol.io/specification/2026-07-28/. Changelog: https://modelcontextprotocol.io/specification/2026-07-28/changelog. Accessed 2026-08-26.

[25] Mark Nottingham. Web Linking. RFC 8288, Internet Engineering Task Force, October 2017. URL https://www.rfc-editor.org/info/rfc8288.

[26] Mark Nottingham, Erik Wilde, and Sanjay Dalal. Problem Details for HTTP APIs. RFC 9457, Internet Engineering Task Force, July 2023. URL https://www.rfc-editor.org/info/rfc9457.

[27] Natasha Noy. Making it easier to discover datasets. https://blog.google/products/search/making-it-easier-discover-datasets/, September 2018. Google (The Keyword), 5 September 2018. Accessed 2026-08-26.

[28] OASIS LegalDocumentML (LegalDocML) TC. Akoma Ntoso Version 1.0. Part 1: XML Vocabulary. Oasis standard, OASIS, August 2018. URL https://docs.oasis-open.org/legaldocml/akn-core/v1.0/os/part1-vocabulary/akn-core-v1.0-os-part1-vocabulary.html. OASIS Standard, 29 August 2018.

[29] OECD. 2023 OECD Open, Useful and Re-usable data (OURdata) Index: Results and Key Findings. Oecd public governance policy papers, OECD Publishing, Paris, December 2023. URL https://doi.org/10.1787/a37f51c3-en. Covers 36 OECD countries and 4 accession countries (data collected in 2022).

[30] Open Contracting Partnership. Open Contracting Data Standard. https://www.open-contracting.org/data-standard/, 2026. States that OCDS is implemented by 50+ national and subnational governments. Accessed 2026-08-26.

[31] Open Data Charter. International Open Data Charter: Principles. https://opendatacharter.org/principles/, 2015. Six principles; adopted by 29 national and 146 sub-national governments as of 2026-08-26 (https://opendatacharter.org/government-adopters/). Accessed 2026-08-26.

[32] Open Geospatial Consortium. OGC API - Features - Part 1: Core, Version 1.0.1 (corrigendum). Technical Report 17-069r4, OGC. URL https://docs.ogc.org/is/17-069r4/17-069r4.html. Also published as ISO 19168-1. Accessed 2026-08-26.

[33] Open Government Working Group. The 8 Principles of Open Government Data. https://opengovdata.org/, December 2007. Sebastopol, California, 7–8 December 2007. Accessed 2026-08-26.

[34] OpenAPI Initiative. OpenAPI Specification v3.1.0. Technical report, The Linux Foundation, February 2021. URL https://spec.openapis.org/oas/v3.1.0.html. Published 15 February 2021.

[35] Poder Ejecutivo (Perú). Decreto Legislativo N.° 1412, Decreto Legislativo que aprueba la Ley de Gobierno Digital. Diario Oficial El Peruano, 13 de setiembre de 2018, 2018. URL https://busquedas.elperuano.pe/normaslegales/decreto-legislativo-que-aprueba-la-ley-de-gobierno-digital-decreto-legislativo-n-1412-1691026-1.

[36] Presidencia de la República (México). Decreto por el que se establece la regulación en materia de Datos Abiertos. Diario Oficial de la Federación, 20 de febrero de 2015, 2015. URL https://dof.gob.mx/nota_detalle.php?codigo=5382838&fecha=20/02/2015.

[37] Presidencia del Consejo de Ministros (Perú). Decreto Supremo N.° 016-2017-PCM, que aprueba la "Estrategia Nacional de Datos Abiertos Gubernamentales del Perú 2017–2021" y el "Modelo de Datos Abiertos Gubernamentales del Perú". Diario Oficial El Peruano, 12 de febrero de 2017, 2017. URL https://www.gob.pe/institucion/pcm/normas-legales/292314-016-2017-pcm.

[38] Project Open Data. DCAT-US Schema v1.1 (Project Open Data Metadata Schema). Technical report, U.S. General Services Administration, Data.gov, November 2014. URL https://resources.data.gov/resources/dcat-us/. Publication date 6 November 2014. Accessed 2026-08-26.

[39] Anders Rundgren, Bret Jordan, and Samuel Erdtman. JSON Canonicalization Scheme (JCS). RFC 8785, Internet Engineering Task Force, June 2020. URL https://www.rfc-editor.org/info/rfc8785.

[40] Igbal Safarov, Albert Meijer, and Stephan Grimmelikhuijsen. Utilization of open government data: A systematic literature review of types, conditions, effects and users. *Information Polity*, 22(1):1–24, 2017. doi:10.3233/IP-160012.

[41] SEMIC / Interoperable Europe. DCAT-AP 3.0.1: DCAT Application Profile for data portals in Europe. Semic recommendation, European Commission, Interoperable Europe, October 2025. URL https://semiceu.github.io/DCAT-AP/releases/3.0.1/. SEMIC Recommendation published 2025-10-27; 3.0.2 is a SEMIC Draft Specification (2026-06-17). Accessed 2026-08-26.

[42] Keyur Shah. Introducing the Data Commons Model Context Protocol (MCP) Server: Streamlining Public Data Access for AI Developers. https://developers.googleblog.com/en/datacommonsmcp/, September 2025. Google Developers Blog, 24 September 2025. Accessed 2026-08-26.

[43] Kevin Smith. api-catalog: A Well-Known URI and Link Relation to Help Discovery of APIs. RFC 9727, Internet Engineering Task Force, June 2025. URL https://www.rfc-editor.org/info/rfc9727.

[44] Sunlight Foundation. Ten Principles for Opening Up Government Information. https://sunlightfoundation.com/policy/documents/ten-open-data-principles/, August 2010. 11 August 2010. Accessed 2026-08-26.

[45] Technical.ly. Maryland is building a roadmap for state info and LLMs, using a simple text file to guide chatbot queries. https://technical.ly/civics/maryland-uses-llms-txt-to-guide-ai-chatbots/, 2026. Accessed 2026-08-26.

[46] Treasury Board of Canada Secretariat. Government of Canada Standards on APIs. https://www.canada.ca/en/government/system/digital-government/modern-emerging-technologies/government-canada-standards-apis.html, 2024. Accessed 2026-08-26.

[47] United States Congress. Foundations for Evidence-Based Policymaking Act of 2018, Public Law 115-435; Title II: Open, Public, Electronic, and Necessary Government Data Act (OPEN Government Data Act). https://www.govinfo.gov/app/details/PLAW-115publ435, January 2019. Approved 14 January 2019.

[48] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, et al. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3: 160018, 2016. doi:10.1038/sdata.2016.18.

[49] Colin Wood. AI agents are swarming. Does your website need an llms.txt file? https://statescoop.com/llmstxt-government-websites-ai/, January 2026. StateScoop, 12 January 2026. Accessed 2026-08-26.

[50] World Wide Web Consortium. Data Catalog Vocabulary (DCAT) - Version 3. W3c recommendation, W3C, August 2024. URL https://www.w3.org/TR/2024/REC-vocab-dcat-3-20240822/. W3C Recommendation, 22 August 2024.

[51] Austin Wright, Henry Andrews, Ben Hutton, and Greg Dennis. JSON Schema: A Media Type for Describing JSON Documents (draft 2020-12). Internet-Draft draft-bhutton-json-schema-01, Internet Engineering Task Force, June 2022. URL https://json-schema.org/specification-links. Draft 2020-12; initial release December 2020, patch published 16 June 2022.

[52] Anneke Zuiderwijk and Marijn Janssen. Open data policies, their implementation and impact: A framework for comparison. *Government Information Quarterly*, 31(1):17–29, 2014. doi:10.1016/j.giq.2013.04.003.

A Conformance checklist

The tests a validator runs, by requirement and level. Each test is stated as a condition on observable artifacts (an interface description, a response, a header) so that it can be automated. A publisher's level is the highest level all of whose tests pass.

Level	Req.	Test
1	**R1**	Interface description links a JSON Schema per record type; a sample of ≥ 20 records validates; every property has a non-empty `description`; every enumerated property lists its values; where a domain schema is declared, the mapping's targets exist in it.
1	**R2**	Schema declares identifier scheme (namespace, syntax, issuer); canonical URL for a sample of ≥ 20 known identifiers returns the record; a valid unassigned identifier returns HTTP 200 with `found: false`.
1	**R5**	Every response carries `source`, `as_of`, `record_version`, `license`; `as_of` ≤ response time; equal `record_version` implies identical canonical payload.
1	**R6**	DCAT description reachable from the interface description with `dct:modified` and a checksum; export record count equals search count at the same `as_of`; sample of ≥ 20 exported records equals lookup.
2	**R3**	OpenAPI declares lookup and search per record type; searchable fields declared; page-size bound declared and enforced with Problem Details; cursor paging yields no duplicates across pages and terminates.
2	**R4**	Every record carries `_updated_at`; `updated_after=` t returns only records with `_updated_at` $> t$ in ascending order; results for $t_2 > t_1$ are a subset of results for t_1; deletions appear as tombstones.
2	**R8**	Rate-limit headers on every response; deprecation policy with notice period linked from the interface description; status page resolves; induced error returns Problem Details with a stable `type`; requests exceeding the declared bound answer 202 with a status URL.
3	**R7**	MCP tool list isomorphic to OpenAPI operations with matching descriptions; tool descriptions state not-found semantics; every documentation URL answers `text/markdown` under negotiation and at `.md`; `/llms.txt` and `/.well-known/api-catalog` resolve and every link in them resolves.

B Envelope schema

The JSON Schema of the response envelope. The `data` member is the record, whose schema is the record type's own (**R1**); the envelope constrains nothing inside it except the presence of `_updated_at` at Level 2.

```
{
```

```
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://agora.example/schemas/envelope.json",
  "title": "AGORA response envelope",
  "type": "object",
  "required": ["found", "source", "as_of", "license"],
  "properties": {
    "found": { "type": "boolean",
      "description": "Whether a record exists for the request. false is a definitive answer, not
    an error." },
    "source": { "type": "object", "required": ["id", "name"],
      "properties": {
        "id":     { "type": "string", "description": "Publisher identifier, stable across records
    ." },
        "name":   { "type": "string" },
        "org_id": { "type": "string", "description": "org-id.guide identifier of the institution,
     when one exists." } } },
    "as_of": { "type": "string", "format": "date-time",
      "description": "Instant the underlying data was last synchronised, or the instant of
    retrieval for a live lookup." },
    "record_version": { "type": "string", "pattern": "^sha256:[0-9a-f]{64}$",
      "description": "Hash of the JCS-canonicalised data member. Required when found is true." },
    "license": { "type": "string", "format": "uri" },
    "data": { "type": "object",
      "description": "The record, conforming to its declared schema. Absent when found is false
    .",
      "properties": {
        "_updated_at": { "type": "string", "format": "date-time",
          "description": "Instant the publisher last changed this record. Required at Level 2."
    },
        "_deleted_at": { "type": "string", "format": "date-time",
          "description": "Present on tombstones in the change feed." } } }
  },
  "if": { "properties": { "found": { "const": true } } },
  "then": { "required": ["record_version", "data"] }
}
```

C Source assessment: method and data

The 37 sources, their institutions and the service assessed are listed in table 6. The per-cell notes and evidence URLs (481 cells) are published with the paper as `scorecard-full.json`; the summary matrix is `scorecard.csv`. Assessment took place on 26 August 2026. Assessors were instructed to use only public documentation, to answer *unknown* rather than infer, to record a note of at most 25 words and an evidence URL per cell, and not to probe any system beyond reading documentation pages. Confidence per source is *high* when official documentation was fetched for most criteria, *medium* when some evidence came from search excerpts or press because official hosts could not be fetched, and *low* when most did.

Table 6: Sources assessed.

	Source	Institution and service	Conf.
CO	ADRES (BDUA)	Administradora de los Recursos del Sistema General de Seguridad Social en Salud (ADRES): Public affiliation lookup by document type and number against the Base de...	medium
CO	ANCP-CCE relatoría	Agencia Nacional de Contratacion Publica - Colombia Compra Eficiente: Relatoria search of ANCP-CCE conceptos, Consejo de Estado providencias, normativa and laudos on...	high
CO	CNDJ	Comision Nacional de Disciplina Judicial (CNDJ): Search of providencias and precedents of the disciplinary jurisdiction	medium

	Source	Institution and service	Conf.
CO	Consejo de Estado (SAMAI)	Consejo de Estado: SAMAI public case lookup (radicado, partes) and relatoria search of providencias (Mi Relatoria since 2021-12-01, legacy buscador for earlier)	medium
CO	Contaduría (BDME)	Contaduria General de la Nacion (CGN): Consultation of whether a person or entity appears in the semi-annual BDME, with PDF certificate	high
CO	Contraloría	Contraloría General de la República (CGR): Fiscal responsibility antecedentes certificate (SIBOR) by cédula or NIT	medium
CO	DIAN	Direccion de Impuestos y Aduanas Nacionales (DIAN): Search of conceptos y oficios (normograma.dian.gov.co/dian/compilacion) and validated electronic document lookup...	medium
CO	Policía Nacional	Policía Nacional de Colombia - DIJIN: Judicial records / requerimientos check by cédula	medium
CO	Procuraduría (SIRI)	Procuraduría General de la Nación (PGN): Disciplinary/fiscal/penal/contractual antecedentes certificate and SIRI sanctions registry	high
CO	Rama Judicial (CPNU)	Rama Judicial - Consejo Superior de la Judicatura: Case lookup by radicado (23 digits) and by party name / razón social	medium
CO	Registraduría	Registraduría Nacional del Estado Civil (RNEC): Cédula status (vigente, cancelada por muerte, etc.) and certificate of vigencia	medium
CO	RUAF	Ministerio de Salud y Proteccion Social (SISPRO): Registro Unico de Afiliados: citizen and entity lookup of active affiliations across social-protection subsystems	high
CO	RUES	Confecámaras (on behalf of the Cámaras de Comercio): Company lookup by NIT, name/razón social and matrícula	medium
CO	RUNT	Ministerio de Transporte / Concesión RUNT 2.0 S.A.S.: Vehicle lookup by plate (requires the owner's document number)	medium
CO	SECOP II	Agencia Nacional de Contratación Pública - Colombia Compra Eficiente: Public procurement processes and contracts (SECOP II), datos.gov.co datasets and OCDS publication	high
CO	SIATA	Area Metropolitana del Valle de Aburra (SIATA programme, with Alcaldia de Medellin): Environmental sensor data (air quality, hydrometeorology, radar, seismic) via...	medium
CO	SICAAC	Ministerio de Justicia y del Derecho (operator of SICAAC): Sistema de Informacion de la Conciliacion, el Arbitraje y la Amigable Composicion: registration and...	low
CO	SIMIT	Federación Colombiana de Municipios (FCM) - Dirección Nacional Simit: Fines lookup by document number, plate or comparendo number	medium
CO	Superfinanciera	Superintendencia Financiera de Colombia (SFC): Public status lookup for complaints and tramites filed against supervised entities (formulesuqueja consultaQuejas)	high
CO	Supersociedades	Superintendencia de Sociedades: Company financial statements (NIIF, 2015+) via SIIS and datos.gov.co; historical 1995-2015 files	high
MX	Banxico (circulares)	Banco de México: Marco normativo: circulars grouped by year and number with per-circular amendment history	high
MX	CNBV	Comisión Nacional Bancaria y de Valores (CNBV): Normatividad vigente collection: disposiciones de carácter general and their DOF modifications	low
MX	Cámara de Diputados	Cámara de Diputados del H. Congreso de la Unión: Federal laws, regulations and reform history (LeyesBiblio) plus the LEYFED compilation download	medium
MX	DOF	Secretaría de Gobernación, Diario Oficial de la Federación: Daily editions by date (dof.gob.mx), SIDOF open-data web services (sidof.segob.gob.mx/datos_abiertos) and...	high
MX	FGJ CDMX (boletines)	Fiscalía General de Justicia de la Ciudad de México (FGJCDMX): Comunicación section: boletines and tarjetas informativas	low
MX	FGR (comunicados)	Fiscalía General de la República (FGR): Prensa section (comunicados nacionales y estatales) on fgr.org.mx	high
MX	Fiscalía Jalisco	Fiscalía del Estado de Jalisco: Comunicación Social, Boletines de Prensa (representative of state fiscalía bulletin sites)	medium
MX	SCJN (Semanario)	Suprema Corte de Justicia de la Nación (SCJN): SJF tesis search (sjf2.scjn.gob.mx, sjf-semanal) and the Repositorio SCJN open-format download (bicentenario.scjn.gob.mx)	medium
MX	SIEM	Secretaría de Economía: Public establishment lookup (siem.economia.gob.mx) and the SIEM establishments dataset on datos.gob.mx	high
PE	APESEG (SOAT)	Asociación Peruana de Empresas de Seguros (APESEG), private industry association: SOAT validity by plate via web, 'Consulta SOAT' app and SMS 90900	medium
PE	Callao (papeletas)	Municipalidad Provincial del Callao (Gerencia General de Transporte Urbano): Traffic fine lookup and payment by plate or papeleta number at pagopapeletascallao.pe	medium
PE	OECE (SEACE / OCDS)	Organismo Especializado para las Contrataciones Públicas Eficientes (OECE, formerly OSCE): OCDS release/record API, yearly bulk files (JSON/CSV/XLSX) and CONOSCE...	high
PE	RREE (carné)	Ministerio de Relaciones Exteriores (RREE): Validate a carné de identidad issued by RREE (diplomatic/privileged foreigners) by carné number and issue date	medium
PE	SAT Lima	Servicio de Administración Tributaria de Lima (SAT): Lookup of traffic fines by plate, papeleta, licence or DNI; vehicle capture orders by plate; account statements	medium

	Source	Institution and service	Conf.
PE	SBS (SOAT)	Superintendencia de Banca, Seguros y AFP (SBS): Insurance history by plate (SOAT, CAT, vehicle policies, claims) for the last 5 years	high
PE	SUNAT (RUC)	Superintendencia Nacional de Aduanas y de Administración Tributaria (SUNAT): Public taxpayer lookup by RUC, by document (DNI) or by name; daily 'padrón reducido' bulk...	high
PE	SUTRAN	Superintendencia de Transporte Terrestre de Personas, Carga y Mercancías (SUTRAN): Lookup of infractions by vehicle plate (online record from 2020; earlier via Mesa...	medium

National portals. The three national catalogs were characterised alongside the sources. datos.gov.co is a Socrata (Tyler Technologies Data & Insights) instance publishing a DCAT-US `data.json`, a per-dataset query API with field filters, paging and a system `:updated_at` column, and CC BY-SA 4.0 on every sampled dataset; its discovery API reported 8,391 datasets. datosabiertos.gob.pe is a DKAN (Drupal) instance with a CKAN-compatible catalog API, governed by PCM's Secretaría de Gobierno y Transformación Digital under DS 016-2017-PCM, DL 1412 and later norms, with no portal-wide default licence; its homepage reported 4,668 datasets from 390 entities. datos.gob.mx is a CKAN 2.11 instance governed by the Agencia de Transformación Digital y Telecomunicaciones under the September 2025 lineamientos, with a working but undocumented CKAN action API and CC BY 4.0 on datasets; its CKAN API reported 1,730 datasets against a homepage claim of 7,420.