

The rule underneath all four

Anything rendered stays one click from the bytes it was rendered from. A reader that only shows you its own interpretation is asking to be trusted. One that shows you the bytes is offering to be checked.

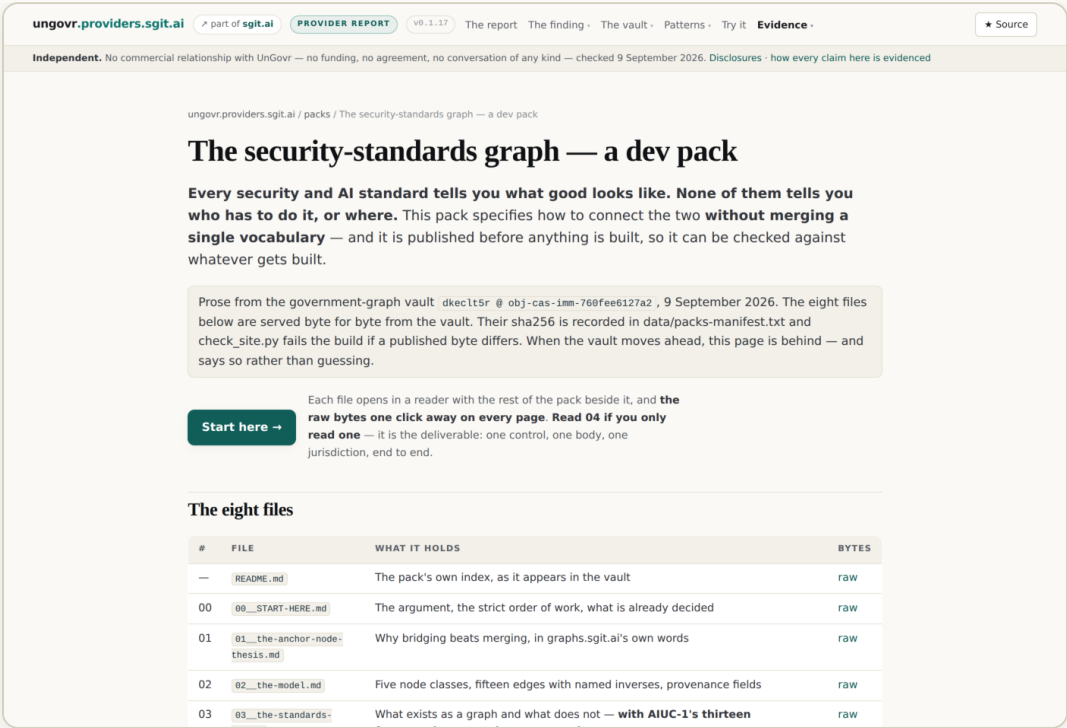
That is the estate's rule, not ours. What follows is four places it got applied in a day — a document reader, a slide viewer, a set of PDFs, and a version history — and the three defects that turned up while applying it.

The dev pack, published raw — and readable

Eight files from the vault, at `/packs/security-graph/`, **byte for byte** and hash-checked on every build.

The reader is the addition: a rail with the whole pack, the document rendered, and **the raw bytes named, hashed and linked above the text** rather than in a footnote under it.

Every `.md` URL still serves the file exactly as it always did.



`/packs/security-graph/` — eight files from the vault, each linking the bytes it was rendered from.

The manifest is what makes "byte for byte" a fact

`data/packs-manifest.txt` records the sha256 of every file. **Two paths carry each one** — the raw copy, and the markdown twin beside its rendered page — and both are checked.

Edit a published pack file here instead of in the vault and the build stops and names it.

A correction belongs in the vault, where the original lives. Making it here would leave two documents that disagree and no way to say which is the source.

ungovr.providers.sgit.ai

^ part of [sgit.ai](#)

PROVIDER REPORT

v0.1.17

The report

The finding

The vault

Patterns

Try it

Evidence

★ Source

Independent. No commercial relationship with UnGovr — no funding, no agreement, no conversation of any kind — checked 9 September 2026. [Disclosures](#) · [how every claim here is evidenced](#)

ungovr.providers.sgit.ai / packs / security-graph / 02_the-model.md

02 — The Model

View the raw file ↴

02_the-model.md — rendered here, and served unrendered at `/packs/security-graph/02_the-model.md`, sha256 `7778412bb213d545`, checked against the vault on every build. **The reader is an addition; the bytes are the document.** The rendering drops the document's own title line, because this page already carries it as its heading, and rewrites intra-pack links so they resolve here.

SECURITY-GRAPH / — 8 FILES

README.md
sha256 `2fe588b095109`

00__START-HERE.md
sha256 `a80f70de732b`

01_the-anchor-node-thesis.md
sha256 `7136286fadd2`

02_the-model.md
sha256 `7778412bb213`

03_the-standards-map.md
sha256 `864398394df6`

04_the-worked-example.md
sha256 `85e99591dd95`

05_integration.md
sha256 `2cdf08a38c05`

06_verification.md
sha256 `84a4ab3a1973`

Every file is one click from its own bytes. The hash beside each name is the sha256 this site checks on every build.

version v0.1 · 9 September 2026

← 01 The anchor-node thesis [link removed: 01_the-anchor-node-thesis.md:] · Next → 03 The standards map [link removed: 03_the-standards-map.md:]

Five node classes, and the prefix says whose claim it is

Origin is carried by the prefix and never by the colour. This is the two-edge rule this vault already ships, extended by one class.

PREFIX	CLASS	WHO ASSERTS IT	EDITABLE BY US
ung:	Anchor — entity, jurisdiction	UnGovr	Never. Transcribed with provenance and left alone
akn:	Instrument — act, provision	The publisher of the law	Never. Anchored to a published citation
std:	Standard — framework, control, requirement	The standards body	Never. Transcribed from their published corpus
sg:	Our model — obligation, control-instance, evidence, acceptance	Us	Yes. This is the only layer we own

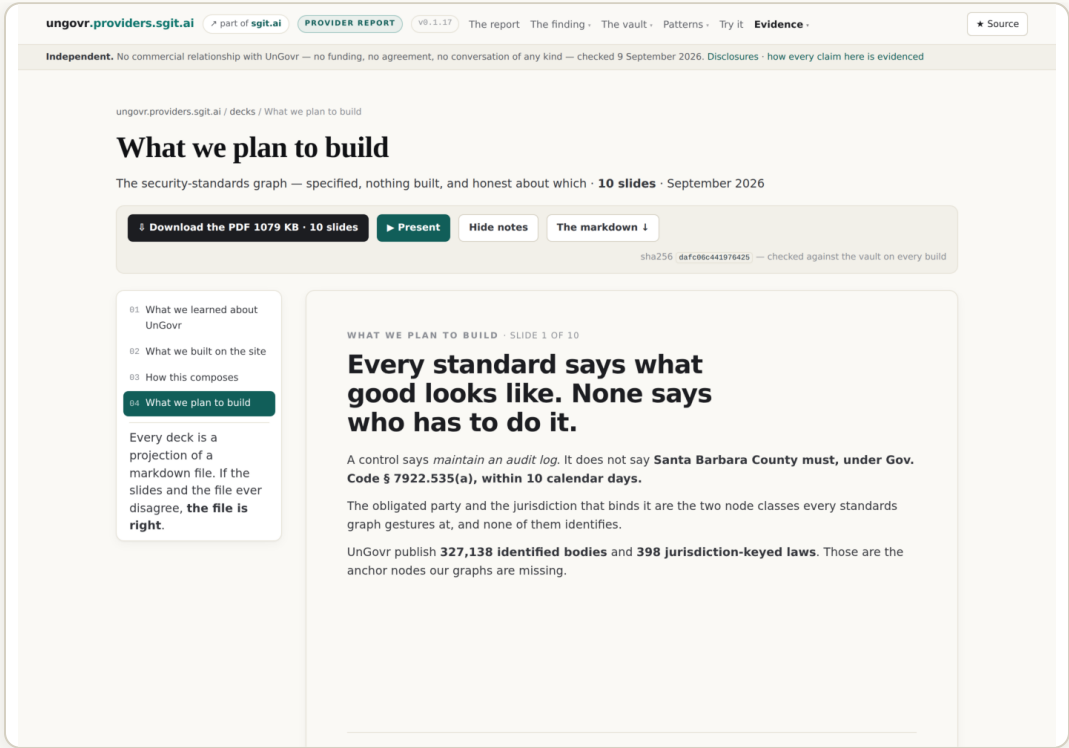
The reader: the pack on the left, the document rendered, the raw file and its sha256 above the text.

The decks: a document first, slides on request

Four decks, **39 slides**, written as ordinary markdown in the vault.

Stacked, numbered, notes visible, printable — **with JavaScript off it is a document**. Present mode, the notes toggle and `#s7` deep links are the enhancement, about forty lines.

The site's parser is a **port of the vault app's**, so both render the same slides from the same bytes.



A deck page: slides stacked and printable, with present mode and the notes toggle on top.

And each deck prints to a PDF worth sending

16:9, one slide per page, at the estate's deck geometry. Printed by a browser from **the same markup the page renders**, so a slide cannot look one way here and another in the file somebody forwards.

Every link in a PDF is absolute. The first run pointed all of them at the localhost the build was printed from — worthless in a file that travels.

Eighteen slides carry a screenshot of this site, each with the URL of the page it shows.

ungovr.providers.sgit.ai

part of sgit.ai

PROVIDER REPORT

v0.1.17

The report

The finding

The vault

Patterns

Try it

Evidence

Source

Independent. No commercial relationship with UnGovr — no funding, no agreement, no conversation of any kind — checked 9 September 2026. Disclosures · how every claim here is evidenced

ungovr.providers.sgit.ai / Four decks, from the vault

Four decks, from the vault

Four presentations, **39 slides**, written as ordinary markdown in the vault and rendered here as real 16:9 slides — **and as PDFs you can send to somebody**. They read as documents in the vault browser, as slides in the vault's app, and as slides here; if they ever disagree, the markdown is right.

Prose from the government-graph vault dkeclt5r @ obj-cas-imm-760fee6127a2 , 9 September 2026. Republished byte for byte from decks/ in the vault; check_site.py verifies every file against a recorded sha256, so a copy that drifted would fail the build. When the vault moves ahead, this page is behind — and says so rather than guessing.

#	DECK	WHAT IT COVERS	SLIDES	PDF
01	What we learned about UnGovr	Eight findings from 76 hashed requests: the 0-of-49, why nobody can census it, the 96.6% that is one hop away, and the licence split	10	1522 KB
02	What we built on the site	The provider report, the argument for admitting an open-data provider at all, and the checks that make its promises enforceable	9	1538 KB
03	How this composes	What this vault contributes to Risk Mandate, Licence to Operate and the AIUC-1 conformance layer — and what it took from them	10	1138 KB
04	What we plan to build	The security-standards dev pack: the 95 crosswalks that point at law, the vocabulary we forked by accident, and the blocker that would kill the thesis	10	1079 KB

Start with 01 if you have never met the data, and 04 if you have.

VERIFIED 9 SEP 2026

Four decks, each with its PDF, its slide count and the size of the download.

Reaching a law: the finding that earned its own page

Follow the evidence to an instrument and UnGovr hands you a structured summary and **a link to the publisher** — on 392 of 398 laws, across 339 hosts.

Not the text, in any format. And **254 of the 398 — every sub-national one — have no reachable detail document at all.**

ungovr.providers.sgit.ai

part of sgit.ai

PROVIDER REPORT

v0.1.17

The report

The finding

The vault

Patterns

Try it

Evidence

Source

Independent.

No commercial relationship with UnGovr — no funding, no agreement, no conversation of any kind — checked 9 September 2026. Disclosures · how every claim here is evidenced

ungovr.providers.sgit.ai / What you get when you reach a law

What you get when you reach a law

The rabbit hole works right up to the law. **Entity** → **jurisdiction** → **instrument all resolve**. Then it stops: there is no text, no PDF, no markdown, no graph — and for **every sub-national jurisdiction**, no detail document that can be addressed at all.

Prose from the government-graph vault

dkeclt5r @ obj-cas-imm-760fee6127a2 , 9 September 2026. Every row below was fetched in this session and hashed; the requests are in the retrieval log. Nothing here is read from their documentation. When the vault moves ahead, this page is behind — and says so rather than guessing.

ON THIS PAGE

The short answer JSON, and only JSON What the summary actually holds — and it is a lot
The provenance link is there, and it works Provision-level identifiers: one field, some of the time
The sub-national half is unreachable Ontology: a controlled vocabulary, not a graph
So: could we help? Yes, and this is the shape of it What this page does not claim

The short answer

YOU WANT	UNGOVR GIVES YOU	STATE
The law's text — md, html, txt, pdf, docx	Nothing. No format carries the instrument itself	absent
A structured summary of it	31-34 JSON fields: deadlines, residency, fees, exemptions, appeals, penalties	rich
A link to the actual source	<code>law_url</code> + <code>primary_source_url</code> + typed <code>secondary_sources[]</code>	good

Reaching a law: the short answer as a table, with every probe in the retrieval log.

Versions get a page of their own

Make it a link, and make the link go to that version's own details — not to a generic changelog. A reader who clicks `v0.1.7` wants to know what v0.1.7 was.

sgit.ai/docs/guidance

Eighteen releases, each with a page naming the git commit it was built from. The history is generated from `data/releases.json`, and served as JSON beside it.

ungovr.providers.sgit.ai

part of sgit.ai

PROVIDER REPORT

v0.1.17

The report

The finding

The vault

Patterns

Try it

Evidence

★ Source

Independent. No commercial relationship with UnGovr — no funding, no agreement, no conversation of any kind — checked 9 September 2026. Disclosures · how every claim here is evidenced

ungovr.providers.sgit.ai / Release history

Release history

admin/build/version.txt owns the version and data/releases.json owns the history. bin/bump.py moves both, and every row links to that version's own page — a reader who clicks v0.1.7 wants to know what v0.1.7 was, not to land on a generic changelog. The release commit's subject repeats it, and CI refuses to tag if the two disagree — which is how a site avoids shipping two versions of itself.

VERSION	DATE	WHAT CHANGED	COMMIT
v0.1.17	2026-09-09	Every release now has a page naming the git commit it was built from, the history is generated from data/releases.json rather than hand-kept beside it, and the nav pill links to the current version rather than to a generic changelog. The same convention in the vault: versions/index.json plus one file per version, and a Versions view in its app.	3a745ff92e
v0.1.16	2026-09-09	The decks become real 16:9 slides with screenshots of this site, and each one downloads as a PDF whose links point back here.	49906b42f4
v0.1.15	2026-09-09	A vault document could run script on this origin, through the pack and deck readers. Confirmed in a browser, closed, and gated — plus link schemes restricted and a literal entity caught.	01e68f28f6
v0.1.14	2026-09-09	A fourth deck — what we plan to build — and all four on the site at /decks/: stacked and printable by default, present mode and deep links on top, markdown one click away.	6d9014f956
v0.1.13	2026-09-09	What you get when you reach a law: JSON only, no instrument text, no graph — and 254 of the 398 records laws, every sub-national one, with no reachable detail document.	ac016c2101
v0.1.12	2026-09-09	The vault app names its own release in its header, and the pill opens a table of what each one changed — the same pattern as this nav, reaching a view instead of a page.	b0b21f920a
v0.1.11	2026-09-09	A reader for the pack files: the rest of the pack beside every document and the raw bytes one click above it. Two corrections it surfaced, one of them the pack forking a vocabulary it said it reused.	d39dab4a54

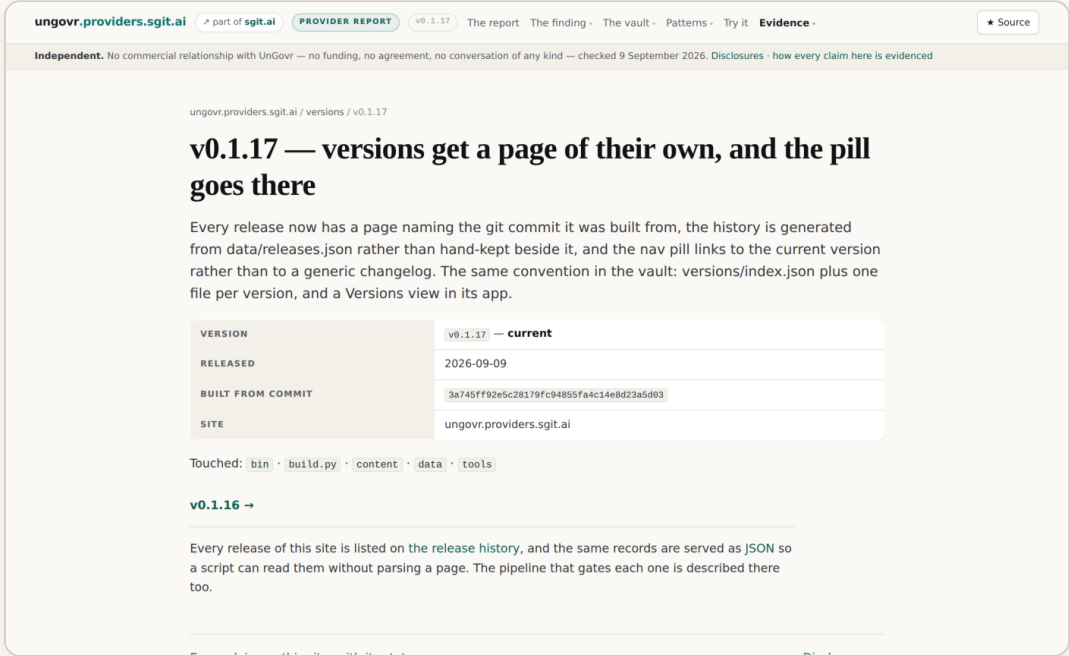
The release history, generated from data/releases.json — every row links to that version's page and names its commit.

Seventeen of the eighteen say they were reconstructed

The words are contemporaneous — each release commit's own message, and its row in the old table. **The structure around them was assembled afterwards.**

Say when a version is reconstructed rather than recorded. A history assembled after the fact is still useful, but only if it is labelled.

So it is labelled, and each entry names what it was assembled from.



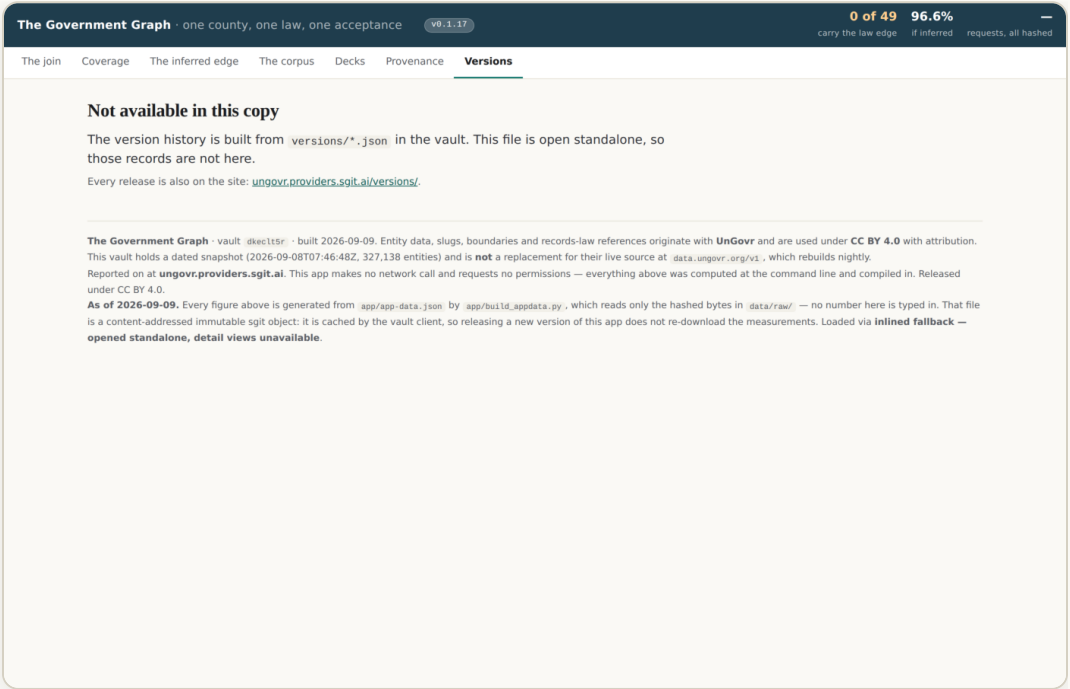
One release's own page: the commit it was built from, what it touched, and that it was reconstructed.

The same convention inside the vault

`versions/index.json` plus one file per version — **data, so an agent can read the history without running anything.**

The app gained a **Versions view** rendered from those files, and the header pill opens it **at the current release.**

The hand-kept release array in the app source is gone. The files are the source.

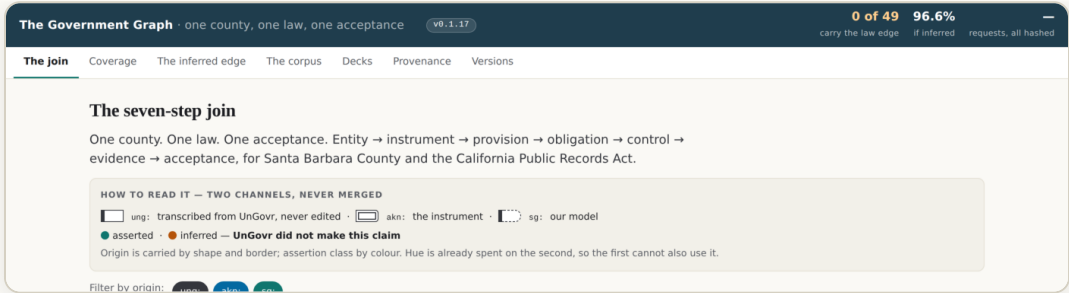


The vault app's Versions view, rendered from `versions/*.json` — the pill in the header opens it here.

A commit cannot contain its own hash

The workflow was: commit, record the sha, amend. **The amend changes the sha.** Record that one and amend again, and it changes again. It does not converge.

So the sha lands in the **following** commit, and two things stop "later" becoming "never": the gate requires a commit on every release but the newest, and `bin/bump.py` refuses to move on while the current one is blank.

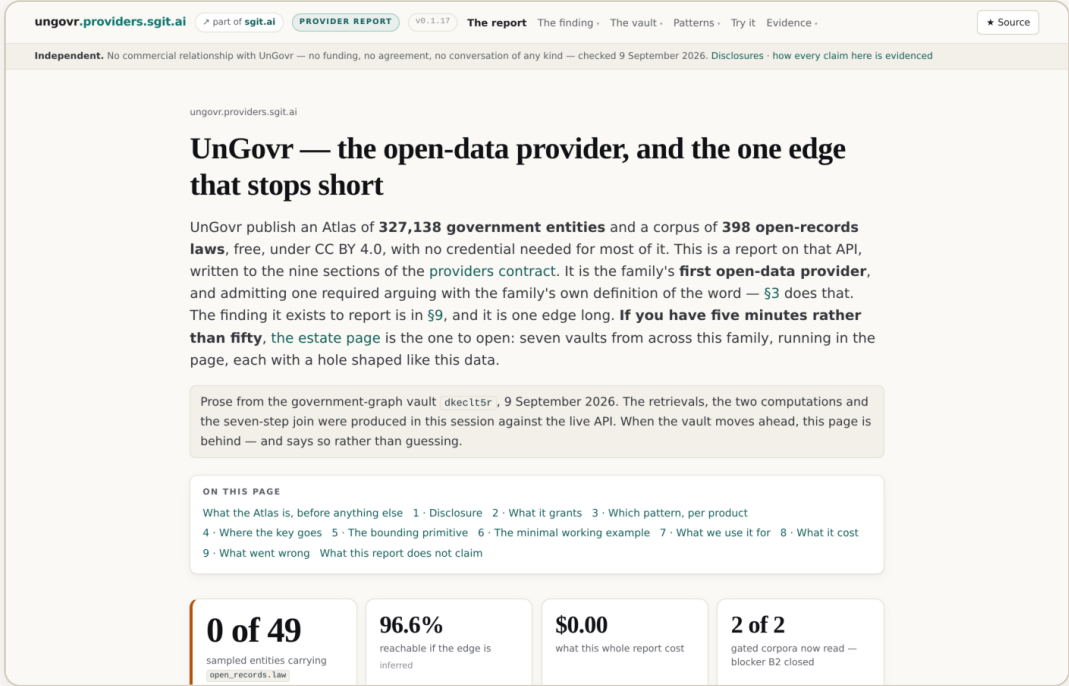


The pill sits in the app's chrome, not a footer, and reaches the current release rather than a list.

Four features, one shape

| The reader | shows the bytes it rendered | | The slides | are a projection of a file that stays readable | | The PDFs | are printed from the markup the page shows | | The versions | name the commit they were built from |

None of it is a promise. All of it is a check — which is the next deck.



The report the four mechanisms serve — every claim carrying the state it earned.