

A promise nobody checks is decoration

This site makes claims about itself: that it contacts nothing, that vault content is republished byte for byte, that every figure is generated rather than typed.

Thirty-two checks turn those from sentences into facts. The build fails; the release does not ship.

What makes the list worth reading is that **almost every entry has a story**, and the story is usually that we got it wrong first.

ungovr.providers.sgit.ai

part ofsgit.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 / The claim ledger — every claim, with the state it earned

The claim ledger — every claim, with the state it earned

Six states, and the default is the weakest one. **The join is the enforcement:** a claim that appears on a page and not in this table fails the build, and a docs claim whose source is not a URL fails it too.

43

verified

12

measured

6

vendor docs

3

specified, not shipped

1

written, not run

1

projected

Run against the API, 9 September 2026

CLAIM	STATE	HOW WE KNOW	WHERE IT IS SAID
data.ungovr.org answers this container. GET /v1/entities/index.json returned 200 with 29,099 bytes at 01:47Z. This closes blocker B1, which the previous session recorded as an egress refusal — the refusal was our environment's, not theirs, and it is gone.	VERIFIED 9 SEP 2026	curl, from the authoring container; the row is in the retrieval log	the report
Santa Barbara County resolves as us/ca/santa-barbara, read from /v1/entities/us/ca.json rather than guessed. This closes blocker B4.	VERIFIED 9 SEP 2026	/v1/entities/us/ca.json, sha256 4a4aa242a2d16941	join
open_records is declared in UnGovr's own published schema as a property of entity_detail, an object with one string member, law. It is not in that definition's required list. The join point is designed and optional.	VERIFIED 9 SEP 2026	/v1/meta/schema.json, sha256 ac7ce7e32a70c65d	the report
open_records appears in exactly one place in the API — the per-entity detail document. It is absent from every bulk surface, which UnGovr's own OpenAPI states: the full-depth index is "a compact record ({slug, name, type, parent_slug?, population?})". So the coverage of this edge cannot be computed from any bulk artefact by anyone.	VERIFIED 9 SEP 2026	/v1/meta/openapi.yaml, sha256 5c344d97b440690d, and the retrieved index files	the report, briefs, coverage

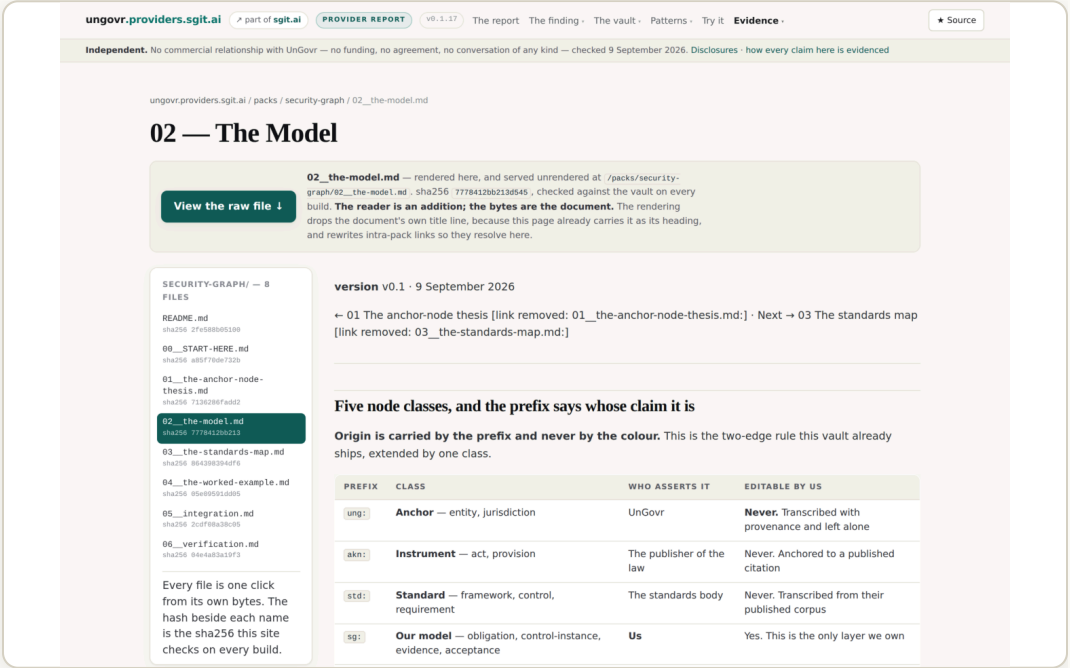
The claim ledger: every factual claim, its state, its date, and the page it is said on.

The worst one: a vault document could run script here

Reading sgit.ai's guidance turned up the rule this site was breaking:

On a *.sgit.ai page there is no host — you are the host — and the bytes you render were written by whoever holds the vault's write key. A vault must be able to change what is shown, and never what the page does.

A <script> in a pack file became a real script tag and executed. So did . Confirmed in a browser, not reasoned about.



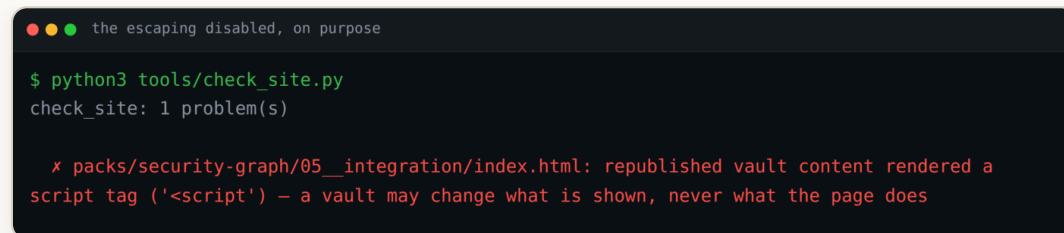
Republished vault content on this origin. Until v0.1.15, a `<script>` in one of these files ran.

The fix, and the gate that keeps it fixed

Republished vault files render in an **untrusted mode**: raw HTML escaped, no shortcodes — a vault file must not mint a claim chip either — and links restricted to `http`, `https`, `mailto`, a relative path or a fragment.

A markdown link was the second vector: `[x](javascript:...)` minted a live `javascript:` href.

`check_no_vault_markup` **reads the built pages** rather than trusting the flag was set.

A terminal window with a dark background and light green text. The title bar shows three colored dots (red, yellow, green) and the text "the escaping disabled, on purpose". The terminal content shows a command prompt followed by a python command, its output, and a detailed error message in red text.

```
the escaping disabled, on purpose

$ python3 tools/check_site.py
check_site: 1 problem(s)

x packs/security-graph/05__integration/index.html: republished vault content rendered a
script tag ('<script') – a vault may change what is shown, never what the page does
```

Real output: the gate firing with the escaping deliberately switched off, then put back.

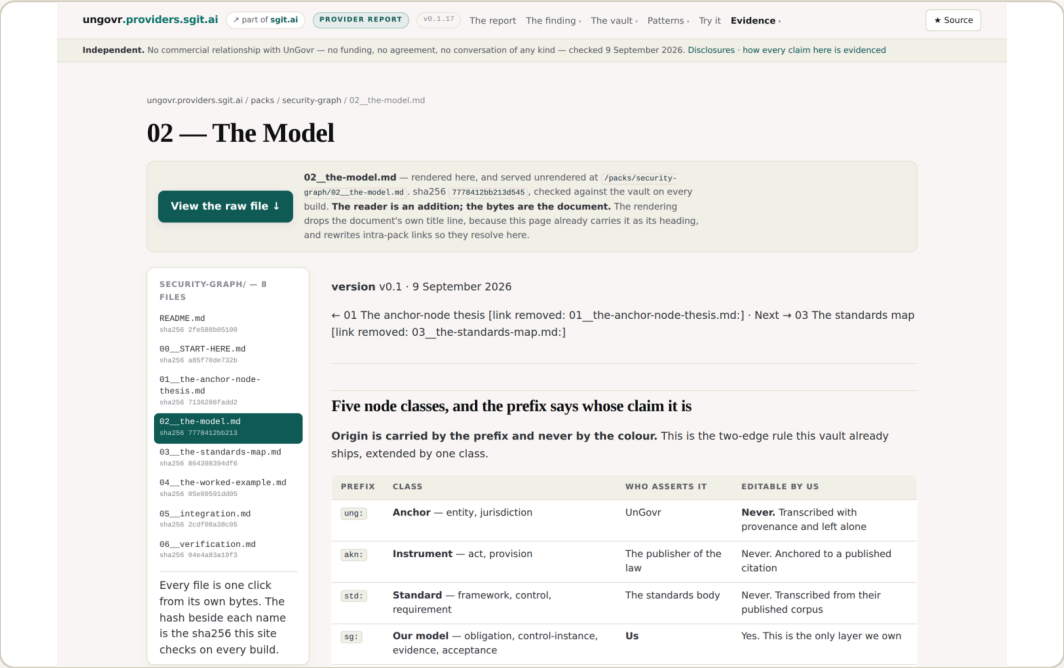
The false positive that would have taught the wrong lesson

The first version of that gate flagged this, in a page that was **working correctly**:

```
<img src=x onerror="...">
```

That is the escaper doing its job — inert text, not markup. A gate that fires on its own fix teaches the next person to loosen the fix until the gate goes quiet.

Every pattern now requires a real  or a real attribute.



The escaped rendering the false positive fired on — inert text, and the renderer working.

A pack file edited here instead of in the vault

Two paths carry every republished file — the raw copy and the twin beside its rendered page — and **both are hash-checked against the vault**.

Change either and the build stops and names it.

A correction belongs in the vault. Making it on the site would leave two documents that disagree and no way to tell which is the source.

A terminal window with a dark background and light text. The title bar at the top reads "a pack file edited here instead of in the vault". The terminal shows a command prompt followed by a script execution. The output indicates a problem with a file in the vault.

```
$ python3 tools/check_site.py
check_site: 1 problem(s)

x packs/security-graph/02__the-model.md differs from the vault – a correction belongs in the vault, not here (recorded 7778412bb213d545, published 0a2015e5a92a1630)
```

Real output: a published pack file edited on the site instead of in the vault.

A deck edited, its PDF not reprinted

The PDFs are generated by a browser and **committed**, because CI has no browser and a deck nobody can download is not the point.

That makes staleness the risk — so `data/deck-pdfs.json` records **the sha256 of the deck each PDF was printed from**, not of the PDF alone.

Move the words, and the build demands the PDF be reprinted.

A terminal window with a dark background and light green text. The title bar shows three colored circles (red, yellow, green) and the text "a deck edited, its PDF not reprinted". The terminal content shows a command prompt followed by a Python command, its output, and a subsequent error message.

```
$ python3 tools/check_site.py
check_site: 3 problem(s)

--
x 01_what-we-learned.md has changed since what-we-learned.pdf was built (deck is
0000000000000000, the PDF was printed from b72727cd738b94db) – rerun tools/make-pdfs.js
```

Real output: a deck's words moved and its PDF not reprinted.

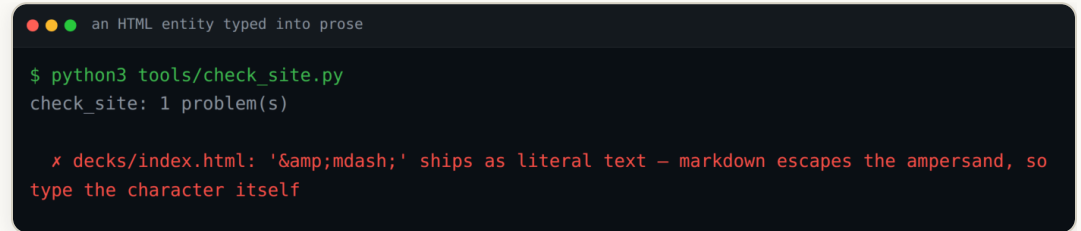
An HTML entity typed into prose

`—` in a page's lead shipped as **six literal characters**.

Markdown escapes the ampersand first, so the reader sees `—` rather than a dash.

It went out in v0.1.14 and was caught **in a screenshot**, not by a check. The build was perfectly happy.

The check that now catches it caught four more of mine within the hour.

A screenshot of a terminal window with a dark background. The title bar at the top reads "an HTML entity typed into prose". The terminal shows a command prompt "\$" followed by the command "python3 tools/check_site.py". The output is "check_site: 1 problem(s)". Below this, a red error message is displayed: "x decks/index.html: '—' ships as literal text – markdown escapes the ampersand, so type the character itself".

```
an HTML entity typed into prose

$ python3 tools/check_site.py
check_site: 1 problem(s)

x decks/index.html: '&mdash;' ships as literal text – markdown escapes the ampersand, so
type the character itself
```

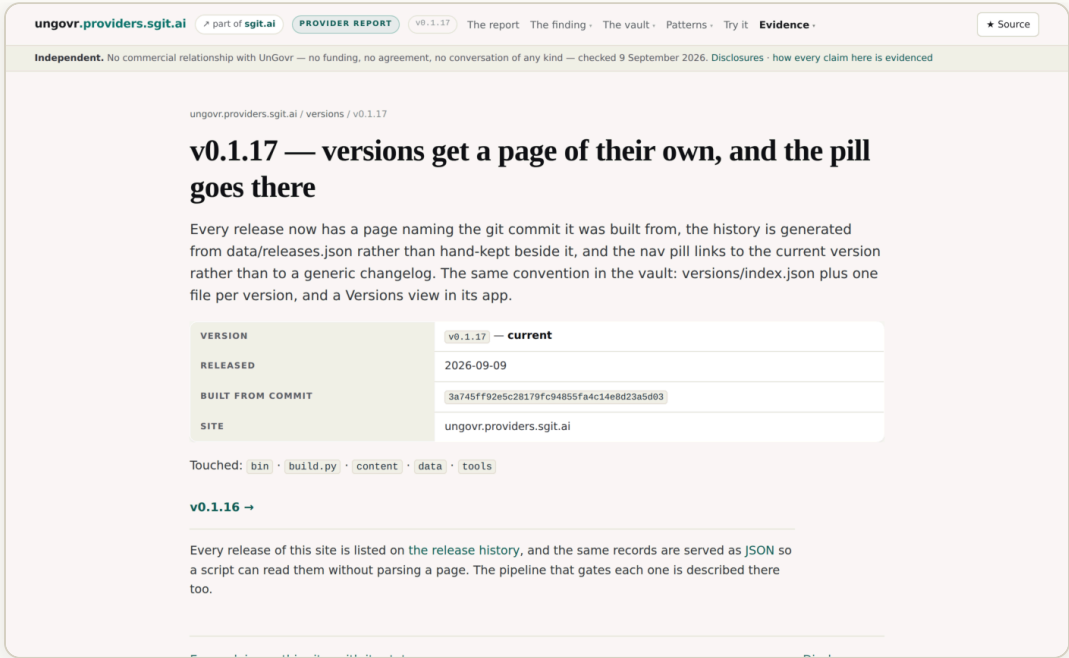
Real output: an HTML entity typed into prose, which ships as six literal characters.

A release that cannot be traced to a commit

Record the commit. A version that does not name the commit it was built from cannot be verified later.

sgit.ai/docs/guidance

Eighteen releases, eighteen commits. The gate refused this very release when it had none — which is how the workflow's own flaw was found.



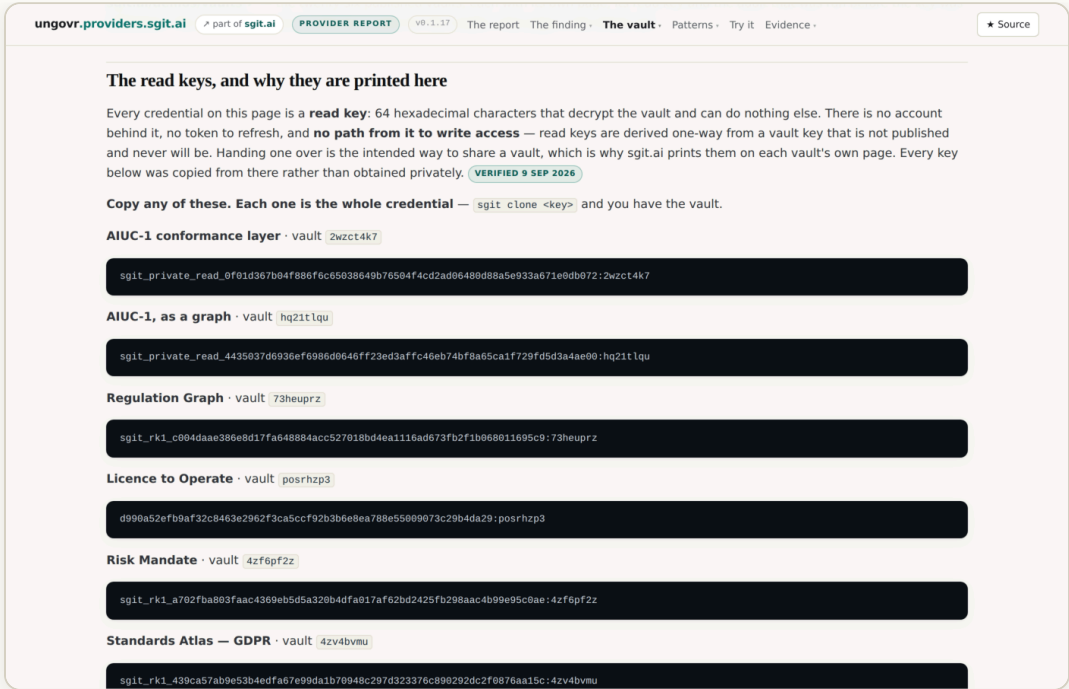
Every release names the commit it was built from — the gate refused this one when it did not.

The scan that had to look past its own prefixes

A credential scan built for sgit shapes will not catch other secrets. We nearly published an OpenRouter key sitting in a vault file, in a field called `openrouter_key`, that matched none of the sgit patterns.

Seventeen patterns now, including one that matches a credential named by its field rather than by a known prefix.

The first version of that pattern did not catch `openrouter_key` either — the anecdote repeating itself inside the fix for the anecdote. Caught by testing it against the case.



Read keys printed in full where a person can copy them. Vault keys never touch this repository.

What this does not prove

No check proves a site is correct. Thirty-two of them prove that thirty-two specific things that once went wrong cannot go wrong silently again.

The defects that shipped were found by **reading the guidance, driving a browser, parsing a table with a script, and looking at a screenshot** — not by the checks.

Checks are memory, not judgement. They stop you making the same mistake twice; they have never once found a new one.

ungovr.providers.sgit.ai

part ofsgit.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 / The retrieval log — every fetch, including the failures

The retrieval log — every fetch, including the failures

The rule: write the raw response bytes to disk unmodified, hash those bytes, and log before parsing anything. A node that cannot name the bytes it came from is an assertion, not a retrieval.

Every number on this site derives from these bytes. Nothing on any page here is fetched at read time — the site makes no network call at all — so this log is the only thing standing between a published figure and an unsupported one.

UTC	METHOD	STATUS	SHA256 OF THE BYTES	BYTES	URL
2026-09-09T01:47:30Z	GET	200	fa22c89c637f1e50...	115	https://data.ungovr.org/v1/meta/last-updated.json
2026-09-09T01:47:31Z	GET	200	ac7ce7e32a70c65d...	2182	https://data.ungovr.org/v1/meta/schema.json
2026-09-09T01:47:31Z	GET	200	85d5b05756fe29f9...	29099	https://data.ungovr.org/v1/entities/index.json
2026-09-09T01:47:32Z	GET	200	4a4aa242a2d16941...	1644990	https://data.ungovr.org/v1/entities/us/ca.json
2026-09-09T01:47:32Z	GET	200	6b71da2f1a20fdff...	109257	https://data.ungovr.org/v1/laws/records/index.json
2026-09-09T01:48:16Z	GET	200	ecb652db836b748f...	29453	https://data.ungovr.org/v1/entities/us.json
2026-09-09T01:48:17Z	GET	200	8d5e6db9e4056b2f...	748	https://data.ungovr.org/v1/entities/detail/us--ca--santa-barbara.json
2026-09-09T01:48:17Z	GET	200	7fa6641141d72042...	146921	https://data.ungovr.org/v1/entities/boundaries/us--ca--santa-barbara.geojson
2026-09-	GET	200	ec4d5238a7dd2c2f...	584	https://data.ungovr.org/v1/entities/detail/us--ca.json

Every request with the sha256 of the bytes it returned, 404s included — checks are memory, not judgement.