Village Governance API: what has changed for an integrator since 11 July 2026

Written 27 October 2026 by an AI agent from the source code and the commit history. It has not had human review. Dates are the commit dates in the repository; where a change was deployed on a different day, the deploy day is not stated here because it was not checked.

If your last look was 11 July 2026, you saw receipt version 3, a signed `attestation_authority`, the sandbox signing identity and the verdict, record and verify routes. Everything below is after that.

Five changes can make a call that worked in July fail or answer differently. They come first. The additions follow, newest first in each part.

Part A. Changes that can break an existing client

A1. `subject_ref` must be at least 16 characters with at least 8 distinct (15 September 2026)

Applies to `POST /governance/attestations`.

- **Old:** any `subject_ref` up to 128 characters without an `@` or whitespace was accepted. `batch-1` returned `200`.
- **New:** below 16 characters, or fewer than 8 distinct characters among them, returns `400`:

```
{ "error": { "code": "invalid_request", "message": "The request did not match
↪  the required schema." } }
```

- **Why:** `request_sha256` in the receipt is an unsalted hash over `{object_sha256, object_kind, subject_ref, idempotency_key}`, and it travels in every bundle you forward. A short label could be recovered by guessing. The floor is not a guarantee: someone who knows your naming scheme is a different case, and salting the hash at mint time, which changes what is signed, has not been done.
- **What to do:** send a UUID, or a SHA-256 of whatever you were sending. Check any code path that passes a row id, a counter or a short label. The same floor applies to `agent_ref` and `identity_binding` on the mandate and inventory routes (see B6).

A2. Every receipt now carries a signed `time_attestation` block (12 and 16 September 2026)

Applies to every receipt: verdict, gate resolution, record seal, attestation, mandate, inventory entry.

- **Old (receipt version 3):** no `time_attestation`. `decided_at` was the only time on the receipt, and it is our clock.
- **New (receipt version 4; version 5 on an attestation):** a `time_attestation` object inside the signed payload. `grade` is `tsa_rfc3161` when an outside RFC 3161 authority stamped the receipt's digest and `operator_attested` when none answered in the window (with an `unavailable_reason`; a nightly sweep retries). From 16 September the block also carries `authorities[]` (one entry for each authority asked), `authority_count`, `jurisdictions`, `authorities_distinct_jurisdictions`, `mutual_assistance_independent` (never `yes`) and `token_authority_id`.
- **Does it break you?** Not if you verify by rebuilding the canonical payload from the receipt as you received it: that rule picks up new fields on its own. Yes if your verifier

rebuilds the payload from a fixed field list, or from the OpenAPI schema, because the block is signed and a list without it fails every version 4 signature. Also yes if a parser of yours rejects unknown fields.
- **A default receipt lists one authority, Certum (Poland, free, not eIDAS-qualified).** A second, qualified authority (Aruba PEC, Italy) is Available on request as an opt-in and has not been used on a live customer engagement. With it, `authorities[]` has two entries and the headline fields name the qualified one. Examples of both shapes are in `USER_GUIDE.md`, built from the unit-test inputs; they are not issued receipts.
- **What to do:** change any fixed-field-list verifier to rebuild from the received receipt, or use `verify-record.js` (https://mysovereignty.digital/downloads/verify-record.js). Read `grade` before you rely on the time.

A3. `action.type` is capped at 128 characters (9 October 2026)

Applies to `POST /governance/verdicts`.

- **Old:** no limit on `action.type`. A 512 KB value was accepted, and when `summary` was absent it became the text the engine scanned.
- **New:** more than 128 characters returns `400 invalid_request`. The engine also reads at most the first 2,000 characters of any text field it is handed, so a long `summary` (limit 500) or `content` (limit 8 KB) is cut for scanning only; the stored and signed values are not cut.
- **What to do:** none, if `type` is a verb such as `email.send`. Check any code that builds `type` from free text.

A4. An expired gate now always resolves to `deny` (13 September 2026)

Applies to `GET /governance/verdicts/{decision_id}` on a `gate` decision whose `gate.expires_at` has passed.

- **Old:** the first poll after expiry resolved the gate to `allow` or `deny` on a hash of the decision id, so about half of expired gates became `allow`. The published guide said `deny`.
- **New:** `deny`, every time, with `reason_codes: ["HUMAN_REVIEW_REQUIRED"]`, `gate.status: "resolved"`, and a new signed receipt appended. It counts against the gate-resolution quota.
- **What to do:** if you hold receipts for gates that expired before 13 September, some record an `allow` that the guide would have told you to expect as a `deny`. Do not treat an old `allow` on an expired gate as a human approval. Treat anything but an explicit human verdict as `deny`.

A5. `GET /governance/committed-seeds/{decision_id}` is gone (21 October 2026)

- **Old:** a key with the `governance:record` scope could fetch a committed-seed commitment or fulfilment by id. The route was documented nowhere.
- **New:** the route is not served. Use `GET /governance/examination-periods/{audit_period_id}` (B1) for the same records.
- **What to do:** if you called it, move to the examination-period route. If you never heard of it, nothing changes for you.

Part B. Additions, newest first

B11. Anchor proof for Governance API receipts on Midnight (27 October 2026). Built and deployed

Numbered B11 so that the earlier B-numbers above and in other documents keep their meaning. It is the newest entry.

`GET /governance/records/{record_id}/anchor-proof` (scope `governance:record`, 30 a minute) returns, for each covered receipt of a record, the path from the receipt's hash to your tenant's day root, the path from that day root to the signed estate root, and the outside anchors of that estate root. The checker `governance-receipt-verifier.js` takes the body with `--anchor-proof` and the text printed by `verify-anchor.mjs` with `--anchor-check`, and adds a supplementary line, `anchored` or `anchor_not_checked`.

- **Nothing existing changes.** A receipt's verdict never depends on Midnight. With the options absent, the checker's output and exit code are as before. A sandbox receipt keeps `sandbox_receipt`; the anchor does not upgrade it. A new error code, `not_yet_anchored`, is added to the closed set; only the new route returns it. A client that rejects an unknown code in the set should add it.
- **When a receipt is covered.** Only after the UTC day closes, the day root is built and the daily job anchors that day's estate root. Until then the route answers `404 not_yet_anchored`, or lists the receipt in `pending_receipt_ids`. A receipt sealed after a day's root is built joins the next day's range. `404 decision_not_found` is unchanged and identical for an unknown id and another tenant's id.
- **State.** Built and deployed, running on Midnight's Preview test network only, which can be reset. On the OVH server the setting that adds Governance receipts to the daily root is on, and the daily anchoring job runs every 30 minutes (a systemd timer). Until a receipt's UTC day has closed and the daily root has been built and anchored, the anchor-proof route answers `404 not_yet_anchored`. The real network can be selected only by an operator through a two-key setting, and it is off. Anchoring is per account: it is on for the Attune sandbox account and off for every other account until MDSL switches it on. A receipt's proof is returned only after its day has been anchored.
- **What an anchor shows.** That a value was recorded under a date key on the ledger. It does not show that the records are true or complete. It is an attestation by our service key. Midnight is run by a federated set of named operators, not a decentralised one. Only hashes go on the ledger. `estate_path` length can indicate roughly how many leaves the day's estate root has.
- **Coverage limits.** The OVH estate only: Catalyst-hosted tenants are not covered. Late receipts join the next day's range. A tenant with a dedicated database is not covered.
- **What to do:** nothing, if you do not need the anchor. To use it, see `USER_GUIDE.md`, "Anchoring on Midnight".

B1. Examination periods (27 October 2026). Built; running for Email Decision Record receipts only

`GET /governance/examination-periods/wk-YYYY-MM-DD` (scope `governance:record`, 30 a minute) returns what your tenant may see of one weekly examination period under MIO-STD-01 section 7: the commitment, the fulfilment, a `late` record and your tenant's selection. `late` carries `late_after_seconds` and `late_last_reason` (at most 64 characters, the last reason the round could not be fetched). The route exists and is mounted. The weekly examination sweep has run on the OVH server since 27 October 2026 for Email Decision Record receipts. It does not yet examine Governance API receipts, so expect `404 decision_not_found` for every period on your account. Nothing breaks for you either way.

B2. Post-quantum second signature (20 October built; 26 October enabled per tenant)

An ML-DSA-65 signature beside the Ed25519 one. It is not inside the receipt, so a verifier that ignores it still checks the receipt exactly as before. It arrives as `pq_attestations` and `pq_policy` beside `receipts` in a bundle, with the public key in `verification_keys`, made afterwards by a background worker, and only for receipts your tenant's key signs (never a sandbox verdict). MDSL switches it on per tenant and tells you in writing; it runs only while a process-wide switch on the host is on (`src/services/governanceApi/governanceReceipt.js:228`), and the state of that switch on production was not read on 27 October 2026. `verify-record.js` reports a bundle without one as "post-quantum NOT PRESENT" and fails it only if you pass `--require-pq`. No requirement window (the point at which a missing second signature becomes a failure) is open for your tenant (read 27 October), and one needs at least 14 days' notice. A sealed record carrying a second signature has not yet been observed; the code is MDSL's own and has had no independent audit.

B3. Time-stamp tokens by hash (21 October 2026). Built

`GET /governance/records/{record_id}/time-stamp-tokens/{token_sha256}` returns the raw RFC 3161 token (`application/timestamp-reply`) so `openssl ts -reply -in file.tsr -token_in -text` can read it. It serves the stored token for a mandate, an inventory entry, an attestation, a verdict or a finance record of your own tenant, and for a record opened through `/governance/records`; every other case returns the same `404`. On 27 October 2026 the route was widened to those record kinds, and a defect was fixed that had made it answer `404` for a hash taken from a real Governance API receipt. Full verification also needs the authority's own root certificate, which we do not ship.

B4. Served verifier: numeric DID hosts refused, `--require-pq` (20 October 2026). Live

`verify-record.js` as served from both domains (checked 27 October, same bytes as the repository) refuses a DID host written as a number and accepts `--require-pq`. It also checks `did:x509` signers against a trust anchor you supply with `--trust-anchor`.

B5. Mandates and the agent inventory (23 September 2026). Live; documented 20 October

Four routes: `POST /governance/mandates` (scope `governance:mandate`), `GET /governance/mandates/in-force?agent_ref=…&at=…`, `POST /governance/agent-inventory` (scope `governance:agent_inventory`), `GET /governance/agent-inventory` (newest 200). Reads need `governance:record`. A key issued before these scopes existed does not carry them. Each write returns a signed, timestamped receipt and counts against the records quota. These routes record; nothing checks a mandate before an agent acts. The in-force answer carries a `caveat` about ordering.

B6. `agent_ref` and `identity_binding` entropy floor (23 and 29 September 2026)

Same floor as A1 (16 or more characters, 8 or more distinct), plus a refusal of anything shaped like an email address or a bare personal name (`400 invalid_request`). A short label such as `agent-svc-7f3a` (14 characters) is refused.

B7. Proof-chain `prev_sha256` (entries written from 24 September 2026, 11:30 NZST)

Each proof-chain entry carries the SHA-256 of the entry before it. Earlier entries have no field and are not linked after the fact. A removed or reordered entry in the middle of a chain is detectable. Removal of the newest entries is not. `verify-record.js` walks the links. An

existing signature check is unaffected; a strict parser of the entry may need to allow the new field.

B8. Attesting a digest (14 September 2026), and reading it back

`POST /governance/attestations` (scope `governance:attest`, 60 a minute) seals a SHA-256 you compute without sending us the content. Receipt version 5. `GET /governance/attestations/{id}` (scope `governance:record`) returns it with the verification keys. See A1 for the `subject_ref` floor.

B9. Record-only routes documented (13 September 2026)

`POST /governance/records`, `…/segments`, `…/seal` and `GET /governance/records/{id}` were live in July and are documented from 13 September. They return `200`, not `201`.

B10. Sandbox caps, restated (no change in the code since July)

Per calendar month (UTC), across all of a tenant's keys: 2,000 verdicts, 100 gate resolutions, 500 record writes. A record write is now any of open, segment, seal, attest, mandate or inventory write. At a cap the call returns `429 rate_limited`. The limiter is set per route and per key (60 for writes, 120 to poll a verdict or verify a receipt, 30 for reads).

Every verdict today is a stub verdict signed by `did:web:agenticgovernance.digital:sandbox`, on every tier. Record seals, attestations, mandates and inventory entries are signed by the tenant's key.