

# Village Governance API — Versioning & Deprecation Policy

*Publishable. A commitment to consumers so you can build with confidence.*

## Version surface

- **API version** in the path: `/api/v1/...`
- **reason\_code\_catalogue\_version** — returned by `GET /governance/policy`. Reason codes are **additive-only within v1**: new codes may be added; existing codes are never removed or repurposed. `UNSPECIFIED` always exists as the catch-all, so we are never forced to mint a revealing code under pressure.
- **receipt\_version** — currently 2. A receipt-format change bumps this; both prior and current versions are listed in `receipt_versions_supported`. Verifiers should branch on `receipt_version`.
- **policy\_epoch** — an opaque handle that changes when a tenant’s effective policy changes. It is not a version of the API; it lets you detect that cached assumptions should be re-checked.
- **canonicalisation\_spec\_version** — the receipt/kōrero canonicalisation used for offline verification. A change here bumps this and is documented in the changelog with a migration note.

## Compatibility commitments (within v1)

- **No breaking change** to a v1 request or response shape without a new major version path (`/api/v2/...`).
- **Additive only** within v1: new optional request fields, new response fields, new reason codes. Your parsers should ignore unknown fields.
- **12-month deprecation notice** on any breaking change or version retirement, published on the docs portal changelog and, for identified integrators, by direct notice.
- **The canonicalisation spec is frozen within a canonicalisation\_spec\_version** — a receipt you verified offline today verifies the same way for the life of that spec version.

## Changelog

Every change to the spec, the reason-code catalogue, or the canonicalisation is recorded on the public docs portal with a date and a plain-language note. The OpenAPI spec is the source of truth and is generated from the same schemas the API uses to build responses, so the docs cannot drift from what the wire actually emits.

## Deprecation mechanics

- A deprecated field/endpoint is marked in the OpenAPI spec (`deprecated: true`) and the changelog, and continues to function for the full notice period.
- Retirement removes it only after the notice period, in a new major version.