Sovereign-Record Architecture for Community-Scale Platforms

John G. Stroh

Paper A · Review draft v3, May 2026 | Languages: EN · DE · MI

Read HTML Download PDF View slideshow Email feedback

Substantive feedback engaging specific sections welcomed. Please cite section numbers (e.g. §6.10) so corrections can be traced. The author replies personally; allow one to two weeks. The paper is available in English, te reo Māori, and Deutsch (links above). The slideshow is currently English-only; localised slideshows will follow at v4 release-candidate.

Sovereign-Record Architecture for Community-Scale Platforms

Cryptographic provenance, tenant-bounded policy enforcement, bilateral federation, and member-driven sovereign portability for non-hyperscaler community infrastructure

John G. Stroh

2026-05-03

Sovereign-Record Architecture for Community-Scale Platforms

Abstract

The default community-scale platform is owned by a US corporation, hosted on US-controlled infrastructure, monetised through attention extraction, and governed by terms the operator may change unilaterally. The default's pervasiveness is not a values-aligned settlement reached by communities weighing alternatives; it is the consequence of more than a decade of sustained corporate investment in shaping user expectations, in network-effect lock-in mechanisms, and in the public-discourse framing through which alternatives are rendered impractical or invisible. The conditions persist in continuous force regardless of community consent, surfacing as visible failures only intermittently — an account locked, a post taken down, a service shut down without notice, a terms-of-service revision against community interest. For some communities — Māori communities holding taonga, minority-language communities whose content *is* the language, family-history groups holding records of named living people, and any community whose form of life does not reduce to a profile object — those conditions are not bearable inconveniences; they are structural impediments to the work the community exists to do. This paper reports a **sovereign-record architecture** — an alternative substrate in which the sovereignty guarantees a community needs are properties of the records themselves, not concessions the operator may revoke.

Every content record in the system carries its own provenance, its own access policy, its own encryption identifier, and a cryptographic chain of every governance boundary it has crossed. Reads expose this state to consumers; writes append to it; deletions tombstone it. Federation between sovereign tenants is bilateral and bounded — two communities agree, on terms they specify, to one specific interaction, and only that. Members are first-class data subjects under whichever regulatory framework applies to them: each may export the full set of records in which they appear in cryptographically verifiable form, and migrate to any tenant operating under the same architectural model. A supervised dialogue surface — operator-cleared editorial queue, draft-and-publish gate, no auto-publish, no outward messaging — extends the read-only stakeholder governance UI into participatory governance without surrendering the supervision discipline.

The architecture runs in production on EU-sovereign (OVH France) and New Zealand-sovereign (Catalyst Cloud) infrastructure. Village-type configurations on the NZ side cover whānau, rūnanga, committee, kāhui-māori, governance, and membership; the EU side covers family, parish, business, and per-product-type demonstration tenants. A carpool tenant is in build on the NZ side as the first contemplated multi-instance federation deployment.

Bilateral federation infrastructure is shipped end-to-end with a comprehensive negative-test matrix covering scope-bound reads, cross-tenant write blocking, audit-log completeness, citation discipline, caching/staleness behaviour, edge-case data states, authorisation boundary enforcement, and namespace conflict resolution; live federation links between independent tenants remain pending the first multi-instance carpool activation.

The architecture is designed to discharge Te Tiriti governance duties on AI systems using or producing Māori data, in line with the prescriptive principles of Dr Taiuru's Kaupapa Māori AI Framework [25b] (Māori consent + data sovereignty over training material + full-chain accountability). The more open question Dr Taiuru [25a] poses about legal personhood for AI agents constituted by Māori knowledge belongs to the empirical territory of the companion paper (Paper B — Situated Language Layers, forthcoming) and is engaged there, not in this paper. A development-time framework (Tractatus, Apache 2.0) records the architectural consultations that produced this design; the persisted ledger forms part of the evaluation surface. The substrate is built by a small team in New Zealand, without venture capital. It is one architectural answer to the question how community infrastructure refuses the conditions of US-platform-default sovereignty without retreating from the work that infrastructure does for the communities it serves.

Keywords: data sovereignty, tenant isolation, cryptographic provenance, bilateral federation, data subject rights, sovereign portability, Te Tiriti o Waitangi, AI personhood, kaitiakitanga, policy inheritance, EUPL-1.2, CARE Principles, GDPR Article 15, Enhanced Border Security Partnership, CLOUD Act, decentralised identifiers.

---

1. Introduction

The default community-scale platform is a SaaS instance owned by a US corporation, hosted on US-controlled infrastructure, monetised through attention extraction, and governed by terms the operator may change unilaterally. The default's pervasiveness is not the consequence of a values-aligned settlement reached by individual users or communities weighing alternatives. It is the consequence of more than a decade of sustained corporate investment in shaping user expectations, in network-effect lock-in mechanisms, and in the framing through which alternative arrangements are rendered impractical or invisible. The conditions persist in continuous force regardless of community consent, surfacing as visible failures only intermittently — an account locked, a post taken down, a service shut down without notice, a terms-of-service revision against community interest — while the underlying settlement remains in continuous force. For some communities — Māori communities operating under Te Tiriti obligations, professional bodies whose members hold confidential material, family-history groups holding records of named living people, conservation groups whose location data is sensitive, parish networks, sporting federations, minority-language communities, and others whose form of life does not reduce to a profile object — the underlying conditions are not bearable: they are structural impediments to the work the community exists to do.

Three pressures combine for these communities.

The first is **jurisdictional**. The CLOUD Act (2018) [9] extends US warrant authority over US-owned cloud providers worldwide, regardless of where the data subject lives. The Enhanced Border Security Partnership (EBSP) negotiations, currently under public discussion in the New Zealand context [13][14], are tied to continued participation in the US Visa Waiver Program with a deadline set for negotiating countries [13][14][15] and contemplate expanded data access including biometric and other identity information [16][17]. University of Auckland legal commentary [18] observes that the US Department of Homeland Security documentation describes EBSP arrangements going significantly beyond the case-based transfers under existing Passenger Name Record (PNR) agreements, raising the prospect of direct

database access. Privacy Foundation New Zealand has raised concerns about transparency and safeguards [19]. Sovereignty in this setting is not a marketing word: it is a question of which state's process can compel disclosure, on what timetable, and with what notice to the community whose data is being disclosed.

The second is **regulatory**. Te Tiriti o Waitangi (the 1840 Treaty between the Crown and Māori), the EU AI Act [6], GDPR Articles 9 and 15 [8], and the European Media Freedom Act [7] each impose obligations on community data infrastructure that are difficult or impossible to honour through delegation. A platform operator that cannot demonstrate which model evaluated which member's content, against which community-authored policies, with what decision, cannot answer the obligations these instruments name. A platform that cannot deliver to a member, on demand, the full set of records the member authored in a verifiable form, cannot satisfy the right of access in GDPR Article 15. Communities operating under Te Tiriti face a corresponding obligation under Article 2 — rangatiratanga over taonga — that an architecture must structurally support rather than merely declare.

The third is **technical**. The commodity AI stack routes inference through a small number of US infrastructure providers and treats every community's content as a potential training input. For communities whose vocabulary, governance protocols, or sacred material cannot be averaged into a global corpus without harm — and whose Te Tiriti or CARE-Principles obligations would be infringed by such averaging — the commodity stack is unworkable.

This paper reports an architectural response. The central commitment is concrete: every content record carries its own provenance, its own access policy, and a cryptographic chain of every governance boundary it has crossed. Reads expose this state to consumers; writes append to it; deletions tombstone it. The architecture runs in production across multiple village-type configurations on EU-sovereign and New Zealand-sovereign infrastructure. The work has been undertaken without venture capital, on a small team's budget, by a private New Zealand company, using a development-time governance framework (Tractatus) that records its own architectural decisions as it goes.

The paper is organised as follows. §2 establishes the background — the development-time governance framework, the Te Tiriti and CARE-Principles framing for Indigenous data sovereignty (with Dr Taiuru's (2026) two-register argument on Te Tiriti obligations and the open question of AI-agent personhood discussed in §3.6), the operational definition of *sovereign records*, the bilateral framing of federation, and the member-as-data-subject framing for sovereign portability. §3 positions the work against the neighbouring literature, including engagement with Dr Taiuru's argument as cited published work. §4 formalises the threat model with named adversaries and testable predicates. §5 states the design principles. §6 reports the architectural implementation. §7 reports federation in production. §8 reports sovereign portability. §9 reports the stakeholder governance UI, including the shipped Phase-6 supervised participatory dialogue surface. §10 walks a worked example of cross-domain naming-sovereignty transfer between two situated language modules. §11 surveys the village-type configurations. §12 reports the evaluation. §13 describes the open-source posture. §14 states the architectural contribution. §15 names what the architecture does not yet do.

---

2. Background

2.1 The Tractatus framework

The development-time governance machinery used to build and operate the platform is the Tractatus framework, a separate research project by the same author. The framework comprises a set of architectural patterns and code services for AI governance during development — chiefly, services that intervene in an AI coding assistant's decision-making at ar-

chitectural choice points. The framework is open source under the Apache 2.0 licence and publicly distributed at `codeberg.org/mysovereignty/tractatus-framework` [1]. A working paper documents the framework's observational findings and the architectural patterns it codifies; specific quantitative figures are reported in the working paper rather than re-stated here.

Tractatus is *development-time* governance: it shapes the platform's source code and architectural choices, not its runtime requests. The platform consults the framework at architectural decision points and persists each consultation as a record in the platform database; consultations are stored by revision identifier, service, condition list, and PASS/FAIL verdict. The discipline of recording the consultation — uniformly on local plus EU-sovereign and NZ-sovereign production databases, automated by per-decision scripts — is the contribution; a future reader can ask which conditions a particular architectural decision addressed, and the answer is in the database, not in the prose.

The separation matters. Tractatus is the framework. The platform is one application of it. This paper reports on the platform; the framework is named for completeness, since the platform's codebase consults it at every architectural decision point.

2.2 Te Tiriti o Waitangi and Indigenous data sovereignty

Te Tiriti o Waitangi, the 1840 Treaty between the British Crown and Māori iwi, is foundational New Zealand constitutional law. Its three articles — recognising tribal sovereignty over taonga (treasured things), Crown governance authority, and equal citizenship — frame contemporary obligations on data. The Waitangi Tribunal *WAI 262* report [5] and the *CARE Principles for Indigenous Data Governance* [4] are widely cited articulations of what these obligations imply for community data infrastructure. The CARE principles — Collective benefit, Authority to control, Responsibility, Ethics — are not the same as the FAIR principles for open data; they coexist with them, and explicitly take precedence where the two come into tension.

A more recent Tribunal report, *WAI 2522* [13a], extended the analysis into international economic instruments — the Trans-Pacific Partnership inquiry, the Mediation Agreement, and the operationalisation of those instruments through the Ministry of Foreign Affairs and Trade. The Tribunal's conclusions in WAI 2522, alongside the work of Ngā Toki Whakarururanga as a Māori-Crown engagement vehicle, sharpen the obligation: a Crown that exposes Māori data to a foreign jurisdictional regime — through trade-treaty data-localisation prohibitions, through CLOUD Act exposure, through an Enhanced Border Security Partnership that contemplates direct database access — cannot satisfy the Article-2 protection of taonga unless the architecture itself prevents such exposure. Architectural sovereignty is the only sovereignty that survives Article-2 scrutiny once the Crown's own treaty obligations create export pathways.

For Māori communities asserting their rights under Te Tiriti, a community platform must let them keep their data on infrastructure they can audit, governed according to their tikanga (customary protocols), with no possibility of cross-community access — including by the platform operator. The architecture answers this directly: tenant isolation is the foundational primitive, not a feature. A platform operator with content-level access to tenant data cannot meet a CARE-principles obligation to *Authority to control*.

Dr Karaitiana Taiuru (Ngāi Tahu, Ngāti Kahungunu) has published extensively on Māori technology ethics, indigenous data sovereignty, AI ethics, and digital rights; his work is available at taiuru.co.nz. The Village situated-language layer that runs on the platform reported here was trained on Dr Taiuru's published frameworks with his authorisation, with citation. Dr Taiuru's wider body of Māori AI governance work is the standing reference point for this paper's posture. His Kaupapa Māori AI Framework [25b] (March 2026), expressed in the whakatauāki *He Tangata, He Karetao, He Ātārangi* (a person, a puppet, a shadow), names

Māori consent and data sovereignty over knowledge used in AI training, and full-chain accountability across developers, operators, and deployers, as required practice grounded in Te Tiriti and the UN Declaration on the Rights of Indigenous Peoples. The architectural primitives this paper reports are designed to discharge those required practices. Dr Taiuru's more recent inquiry [25a] poses, separately, the open question of whether and under what conditions legal personhood might be extended to AI agents constituted by Māori knowledge — drawing on the WAI 2522 finding that Māori data is taonga and on the precedent of the three natural-feature legal-person Acts (Te Urewera 2014; Te Awa Tupua 2017; Te Kāhui Tupua 2025) — an inquiry he expressly defers to collective work among AI developers, government agencies, and Māori communities. The personhood inquiry belongs to the empirical territory of the companion Paper B (forthcoming), where situated-language-layer cohort training discipline is reported in the detail any future per-cohort partnership engagement (including with Mead's Tikanga Test) would draw on; this paper does not pre-empt that work.

The architecture does not assume Māori communities specifically. The same property — a platform that cannot see across communities — answers the regulatory obligations EU minority-language communities face under the European Media Freedom Act and the GDPR's Article 9 special-category protections, where minority-language cultural data is plausibly construed as a special category. A Welsh community, a Sámi community, a Sorbian community, a Frisian community, a Catalan community can adopt the same architecture by re-training the language layer on a different corpus and re-authoring the policy under their own legal framework; the architecture itself is portable. This portability is a central architectural property.

2.3 Why "sovereign records" rather than "encrypted at rest"

The phrase *sovereign record* is intentional. Encryption at rest is a feature; sovereignty is an architectural property. A record is sovereign in the sense intended here when each of the following is true:

1. It carries its own provenance — who wrote it, who is its kaitiaki (steward), under what tikanga it was shared, when it was created, and a cryptographic hash binding those fields together.
2. It carries its own policy — who can read it, who can train on it, who can export it, what happens when policy conflicts.
3. It carries its own proof chain — every governance boundary it has crossed (creation, update, export, deletion) recorded with a cryptographic signature.
4. The cache of its verification state is observable at read time — every API consumer sees whether the record's chain is fresh, expired, mismatched, or unverifiable, without having to trust the platform's word for it.
5. It is portable on member request. A member may export the full set of records in which they are the author, the kaitiaki, or otherwise named as a data subject; the export carries the proof chain forward; an external verifier holding the source tenant's published identifier document can reconstruct every signed entry the records carry.

These are operational properties, testable from the API surface, not aspirational ones. The remainder of this paper describes how each is constructed.

2.4 Federation: bilateral and bounded

A federation in the platform's sense is the narrow technical arrangement that lets two sovereign tenant instances connect for specific bounded purposes — joint events, shared carpools, cross-instance announcements — without either surrendering data, identity, or governance authority. A federation is a bilateral agreement: the two tenants' constitutions agree, the two operators agree, the federation manifest is signed by both. There is no central

federation server; the data flow is direct, the governance is local, and either party can revoke the federation at any time.

This is structurally distinct from the platform model, where instances are leaves of a single operator's tree. It is also structurally distinct from the dominant fediverse model, where federation is a network-wide property mediated by a shared protocol between operator-controlled servers. The architecture described here is bilateral and bounded: two communities agree, on terms they specify, to a specific interaction, and only that.

§3 positions this against the neighbouring literature. §7 reports the federation infrastructure that has shipped and the live-deployment status. §10 walks a worked example of a bilateral federation between a botanical-knowledge module and a language-revitalisation module — a class of federation contemplated for curriculum integration in primary-school settings.

2.5 The member as data subject

A member of a sovereign-record tenant is, simultaneously, a member of the community (with the social, governance, and content-creation rights that membership entails) and a data subject under the regulatory frameworks that apply to them. A Welsh-language community member is a GDPR data subject; a Māori community member's data falls under both Te Tiriti rights of the iwi and the individual's GDPR rights when the iwi operates EU-hosted infrastructure for diaspora members; a German Verein member is straightforwardly a GDPR data subject.

The architecture treats the member-as-data-subject framing as first-class. Every record's `metadata.origin` block names the author and (where distinct) the kaitiaki, both as decentralised identifiers. Every record's `metadata.policy` block declares whether and how that record may be shared, trained on, or exported, and policy conflicts resolve via the constitutional default. Member-driven export (§8) takes the policy at face value: a record's policy may forbid its export even by its author (e.g., a deliberation contributed under collective-consent terms), and the export wrapper enforces this. Sovereignty is not member-versus-collective; it is the architectural framework within which both rights coexist, and the framework makes the conflict explicit rather than burying it in implementation choices.

---

3. Related Work

The architecture sits at the intersection of several active research and development lines. Positioning is crucial because the contribution is not the introduction of any single primitive — well-known building blocks are reused — but the integration of those primitives into a substrate that answers the threat model in §4 from the record level upwards rather than from the operator level downwards.

3.1 Federated social infrastructure

ActivityPub [Snell & Prodromou, 2018, W3C Recommendation [20]] and the Mastodon ecosystem establish federation as a network-wide property mediated by a shared protocol between operator-controlled servers. In ActivityPub federation, two servers federate by exchanging signed activity objects; the granularity is per-actor and per-activity, mediated by the protocol's collection endpoints. This is structurally distinct from the bilateral-and-bounded federation described here. ActivityPub's contribution is interoperability across thousands of instances; this paper's contribution is sovereignty preservation across exactly two instances at a time, by signed manifest, with revocation as a first-class operation. Both architectures are valid responses to different sovereignty postures: ActivityPub optimises for graph reach; this paper optimises for tribal/collective authority over the federation envelope.

The decentralised social-media literature documents federation-graph properties, instance-level moderation tradeoffs, and content-portability gaps. Empirical characterisations of the Mastodon graph and instance-moderation patterns [31][32] inform the operational analysis of fediverse platforms; subsequent work on AT Protocol [Bluesky Public Benefit Corporation, 2024 [21]] proposed account portability — a member's data follows the member rather than the server — as a structural answer to the moderation-and-discoverability problem. This paper's sovereign portability (§8) is technically related but motivated differently: portability is a data-subject right under GDPR Article 15, and the receiving tenant's constitutional acceptance check (§8) is integral, not optional. AT Protocol's portability is account-driven; this paper's portability is record-driven and policy-respecting, with a withheld-list manifest covering records whose policy forbids export even by their author.

3.2 Decentralised identifiers and verifiable credentials

The W3C Decentralized Identifiers (DIDs) v1.0 specification [11] establishes a method-agnostic identifier scheme that supports multiple resolution mechanisms. The platform's tenants and members publish DID documents at well-known endpoints under the tenant's own domain; verification of cryptographic operations (provenance hashes, proof-chain signatures, federation manifests, export bundles) is performed against these documents. This pattern is widely adopted; the architectural decision is to make every cryptographic operation in the system independently verifiable using only the source tenant's published DID document and standard cryptographic primitives — no third-party verifier, no shared trust root, no centralised registry.

3.3 Solid and personal data stores

Solid [Mansour et al., 2016 [22]; W3C Solid Community Group] and the Inrupt platform place data in personal pods owned by the data subject, with applications requesting access via WebID-based authorisation. Solid's architectural commitment is *data per individual*; this paper's is *data per community, with members as first-class data subjects within the community*. The two are complementary rather than competing: a Solid pod could in principle serve as an export destination for a sovereign-record bundle, and a community whose members each maintain Solid pods could in principle implement the platform's sovereign-record tenant on top. The platform's choice to centre the *community* unit reflects a substantive position that minority-language and Indigenous communities are not aggregations of individual data subjects: the collective is the rights-holder for taonga (CARE principle: Collective benefit), and the architecture must serve the collective's authority (Authority to control).

3.4 Federated learning and data trusts

Federated learning [McMahan et al., 2017 [23]; Kairouz et al., 2021 survey [24]] is *architecturally distinct* from this work. Federated learning trains a shared model by exchanging gradient updates across data-holding parties without exchanging raw data. This paper's federation does not exchange model parameters at all — federation is a tenant-to-tenant data agreement under signed manifest, with the data flow defined per agreement, and no shared model is implied. The platform's situated-language layer is per-tenant by construction; cross-tenant model-parameter sharing would violate the tenant-isolation foundational primitive (§5.1). Empirical training-discipline evidence for the situated-language layer is presented separately in Paper B (forthcoming).

Data trusts — as developed in the Open Data Institute's working definition of a data trust as "a legal structure that provides independent stewardship of data" [33] and Element AI's parallel research on data-trust institutional design as a mechanism for addressing power asymmetries between technology firms, government, and the public [34] — introduce a third-party trustee

that holds data on behalf of a community and brokers access. This paper's architecture has no such trustee. The platform operator can run infrastructure operations (create tenants, manage billing, monitor health) but cannot read tenant content. There is no role in the system that aggregates cross-tenant data even temporarily. A data trust's strength is institutional brokerage; this architecture's strength is the impossibility of brokerage from inside the platform.

3.5 GDPR Article 15/20 implementation

The right of access (Article 15) and right to data portability (Article 20) of the General Data Protection Regulation [8] have a substantial implementation literature — including Wachter, Mittelstadt & Floridi [26] on the explanation-rights debate, and Edwards & Veale [27] which argues the right of erasure (Article 17) and data portability (Article 20) carry more practical weight than explanation rights for algorithmic accountability. The platform's DSR endpoint (§8) implements all six relevant rights (Articles 15, 16, 17, 18, 20, 21) via a uniform sovereign-record export pipeline that returns the data subject's records with full proof-chain preservation, in JSON, CSV, or PDF formats, as a single canonical bundle. The technical novelty here is the *cryptographic verifiability* of the bundle: any external party with the source tenant's published DID document can verify every entry the bundle's records carry, without trusting either the source tenant or the data subject's chosen receiving party. Standard Article-15 implementations return data; this implementation returns *verifiable* data.

3.6 CARE Principles and Indigenous data governance

The CARE Principles [4] frame a substantive obligation that has architectural implications: *Authority to control* requires that the community — not the platform operator, not a third-party trustee, not a regulator with subpoena power outside the community's jurisdiction — can govern the data on its own terms. Subsequent work in Indigenous data governance — Walter & Suina [25] on Indigenous data and methodologies; Te Mana Raraunga's *Principles of Māori Data Sovereignty* [28]; Carroll, Rodriguez-Lonebear & Martinez [29] on US Native-nation strategies; Hudson, Anderson, Dewes, Temara, Whaanga & Roa [30] on conceptualising big data through a Māori lens — has elaborated the implications across data lifecycles (collection, storage, processing, sharing, archival). The platform's architectural choice — tenant isolation as foundational, sovereign records as the substrate, bilateral federation as the only cross-tenant mechanism — is one technical answer to the CARE obligations; it is not the only possible answer, but it is an architectural answer that survives Article-2 Te Tiriti scrutiny in the specific case of New Zealand iwi data and EBSP-class jurisdictional pressure.

Dr Taiuru's body of Māori AI governance work [25a, 25b] (introduced in §2.2) sharpens the CARE obligations into specific architectural commitments. The prescriptive duty — Māori consent and data sovereignty over knowledge used in AI training, full-chain accountability across developers and operators, and Te Tiriti obligations on AI systems using or producing Māori data — applies now to any platform whose AI surfaces touch taonga material. The architectural primitives this paper reports (tenant isolation as Article-II rangatiratanga over data; per-tenant cohort training as the locus of community-determined model behaviour for Māori-bearing tenants; the shipped Phase-6 supervised dialogue surface (§9) as kaitiaki supervision over what the surface emits; cryptographic-deletion finality as rangatiratanga over what is forgotten; DID-based attribution and the proof chain as whakapapa traceability of every record) are intended as a structural response to that prescriptive duty. The further question Dr Taiuru [25a] poses — whether and under what conditions legal personhood might be extended to AI agents constituted by Māori knowledge, on the precedent of Te Urewera (2014), Te Awa Tupua (2017), and Te Kāhui Tupua (2025) — is engaged in the companion Paper B at the level of cohort training discipline, not pre-empted in this paper.

3.7 Adjacent threats: foreign-cloud mining via frontier AI

Industry commentary on sovereign-AI architecture [22b][23b][24b] has analysed the convergence of US legal-access frameworks (CLOUD Act; FISA) and frontier AI model deployment on US-controlled cloud infrastructure as a combined risk pathway: data accessed under foreign legal compulsion can be mined at scale by frontier AI models for patterns that exceed the original disclosure scope. The implications for biometric, identity, and authentication data are particularly sharp. The architectural answer this paper proposes is that critical credentials and biometrics never sit in foreign-reachable cloud infrastructure in the first place, and that any LLM interaction with sovereign data passes through a tenant-governed interface where the tenant constrains what an external model can learn or retain.

---

4. Threat Model

This section formalises the threats the architecture is designed to resist. The model names six adversaries, states the sovereignty invariants each must not violate, and resolves each invariant to a testable predicate.

4.1 Adversaries

A1. Jurisdictionally-compelled host operator. A platform operator who is themselves compelled by a foreign legal regime — a CLOUD Act order, a FISA warrant, an Enhanced Border Security Partnership database-access provision, an equivalent provision under another jurisdiction — to disclose tenant data they technically can access. The compulsion may be accompanied by a gag order. The operator may be acting in good faith, in bad faith, or under duress; the architecture is indifferent to the operator's motive and assumes the worst case.

A2. Co-tenant. Another tenant on the same platform infrastructure who attempts to read content that does not belong to them, whether through query construction, schema knowledge, role-escalation, or exploitation of a shared resource (database, cache, file system).

A3. Cross-tenant federation peer. The tenant on the other side of a bilateral federation agreement, who attempts to access data outside the manifest's bounded purpose, or whose own infrastructure is itself jurisdictionally compromised (chain-of-trust attack via federation).

A4. Member-as-attacker. A member of a tenant who attempts to access content they are not authorised to read (e.g., another member's private deliberation, a subgroup-restricted record they are not a member of), or who attempts to assert authority they do not hold (e.g., perform a tenant-admin operation, modify the constitution).

A5. Foreign-cloud-mining-via-frontier-AI. An adversary who has compelled access to data via A1, then routes the data through a frontier AI model to extract patterns that exceed the legal scope of the original disclosure — biometric correlation across populations, authentication-pattern inference from session metadata, social-graph reconstruction from interaction traces.

A6. Biometric-leverage adversary. An adversary who, by combination of A1 (jurisdictionally-compelled host) and A5 (foreign-cloud-mining-via-frontier-AI), seeks to exploit biometric data persisted by the platform — faces, fingerprints, voiceprints, iris scans, behavioural-biometric profiles — to identify, correlate, or coerce members. The adversary's leverage scales with the irrevocability of biometric data: a leaked password is rotatable, a leaked faceprint is not. The adversary's reach is amplified by the three converging US-jurisdiction exposure paths for biometric data — direct US-side capture at borders and visa interviews, US-cloud hosting under CLOUD Act compulsion, and future Enhanced Border Security Partnership arrangements contemplating direct-database-access to partner-country biometric repositories. The

architectural answer is that the platform persists no biometric data of any kind in any path it controls (see §5 design principles and §13.1 vendor discipline).

A7. Misattribution-via-aggregating-agent. A future agent surface — runtime, persistent, goal-directed — that aggregates content across tenants or across records inside a tenant in ways that escape per-record `share_within` policy or per-record kaitiaki attribution; or that produces emergent attributions (authorship, kaitiakitanga, tikanga-bearing relationships) the underlying records do not warrant. The platform's runtime today comprises single-turn situated-language-layer dispatch (§5 design principles) and the shipped Phase-6 mdsl participatory dialogue surface (§9), which is itself supervised single-turn — operator-cleared editorial queue, draft-and-publish gate, no auto-publish, no outward messaging. Neither presents A7's full technical mechanism (autonomy + persistence + cross-record aggregation). The Te Tiriti governance duty Dr Taiuru (2026) asserts (see §3.6) bears on this adversary's substantive load — namely that any cross-record or cross-tenant emission from a Māori-bearing tenant carries the duty regardless of the surface's autonomy level. The architecture's existing invariant I3 (policy-respecting disclosure) is the primary technical defence: every cross-record or cross-tenant emission is policy-gated at the route boundary; kaitiaki attribution and the proof chain carry whakapapa-traceability through every record; the Phase-6 editorial-queue + draft-and-publish gate (modelled on the Mastodon-publish-only-on-instruction precedent) is the discipline against silent emergence of stronger-sense behaviours from the supervised baseline. Adopting Dr Taiuru's wider counsel — that tikanga compliance is far easier to build in from the beginning than to retrofit — A7 is named here so that any future move toward autonomy or persistence inherits the refusal property as a design-time invariant rather than a remedial patch.

4.2 Sovereignty invariants

For each adversary, the architecture defends one or more invariants:

I1. Tenant content isolation. No platform operator role, no co-tenant, no automated process outside the tenant's own request context can read tenant content. (Defends against A1, A2.)

I2. Provenance authenticity. The author and kaitiaki of every content record are cryptographically bound to the record's content; neither field can be silently revised without invalidating the verification cache. (Defends against A1, A4.)

I3. Policy-respecting disclosure. Every cross-tenant or cross-boundary data flow respects the record's `metadata.policy.share_within` and the federation manifest's bounded purpose; flows outside that envelope are refused at the route boundary. (Defends against A1, A3.)

I4. Cryptographic-deletion finality. Records whose `metadata.policy.delete_must_be_cryptographic` is set are deletable in a manner that renders the ciphertext unrecoverable from the persisted state, even by the platform operator with full database access. (Defends against A1, A5.)

I5. Federation manifest integrity. A bilateral federation does not activate until both parties' signatures verify against their respective published DID documents; revocation is itself a signed event; no third party can backdoor a federation. (Defends against A3.)

I6. Audit reconstructability. Every cross-boundary event (creation, update, export, deletion, federation activation, federation revocation, membership change) leaves a signed proof-chain entry; the tenant can reconstruct every event from its own database without trusting the platform operator's say-so. (Defends against A1, A3.)

I7. Sovereign-portability honesty. A member's right-of-access export is filtered by the same policy gate that governs ordinary reads; records whose policy forbids export are listed in the export manifest as withheld with the policy reason cited. (Defends against A4 and

protects A1's ability to assert "we exported to the data subject everything they were entitled to receive.")

I8. Off-platform-mining bound. No content record leaves the tenant's database in plaintext form except via a route that has been reviewed against the tenant's constitution; runtime AI inference (situated language layer) executes on tenant-controlled infrastructure with policy-gated input. (Defends against A5.)

I9. No biometric collection. The platform collects, stores, and processes no biometric data of any kind in any path it controls — no faceprints, no fingerprints, no voiceprints, no iris templates, no behavioural-biometric profiles, no biometric-derived keys. (Defends against A6; strengthens A1 — the operator cannot be compelled to disclose what was never collected; strengthens A5 — no biometric mining surface exists.)

4.3 Testable predicates

Each invariant resolves to one or more predicates testable from the API surface or by direct database inspection.

For I1 (tenant content isolation): every tenant-scoped collection's queries are constructed such that omitting a tenant filter raises a runtime error; an automated test suite asserts this. The platform's `AsyncLocalStorage`-based request-context plugin enforces the predicate; queries outside request context (scheduled tasks, batch jobs) must explicitly opt out and document why.

For I2 (provenance authenticity): for any record `r`, `recompute_provenance_hash(r.metadata.origin) == r.metadata.origin.provenance_hash`; the canonical-form serialisation is hydration-mode-stable (a 2026-04-22 incident, where 25 unit tests passed cleanly while real-Mongoose-document hashes diverged from save-time, surfaced this requirement). Use-case validation (§12) asserts hash stability across hydration modes.

For I3 (policy-respecting disclosure): for any cross-route or cross-WebSocket emission, the effective-policy gate is invoked; a record whose `share_within` value is not in the recognised vocabulary fails CLOSED with reason `share_within_unknown_scope`; this is the project's *be honest about what cannot be verified; do not conjure permission* posture. An automated test scaffolds federation peer scenarios and asserts the gate's behaviour for each.

For I4 (cryptographic-deletion finality): for any record marked `delete_must_be_cryptographic`, deletion destroys the per-record encryption key in the tenant key store; subsequent read attempts return `unverifiable` rather than `valid`; the ciphertext at rest is not recoverable by re-keying.

For I5 (federation manifest integrity): a federation manifest carries signatures from both parties; `verify_signature(manifest, party_a.did_document) == true && verify_signature(manifest, party_b.did_document) == true`; the federation does not activate if either fails; a static test asserts no code path activates federation without these checks.

For I6 (audit reconstructability): for any tenant-bounded event sequence, the proof chain on each affected record can be reconstructed by reading the signed entries; no event is silent; the architecture's audit-log writer is called from a single chokepoint that cannot be bypassed by a controller skipping the call.

For I7 (sovereign-portability honesty): for a Article-15 export request, the response includes (a) the canonical bundle, (b) the withheld-list naming each excluded record and its policy reason, (c) a signed receipt covering both. An integration test asserts withheld records are excluded *and* listed.

For I8 (off-platform-mining bound): the runtime inference layer is hosted on tenant-controlled or community-trusted infrastructure (in the platform's case, EU-sovereign OVH

France or NZ-sovereign Catalyst Cloud, or a designated home-eGPU failover). No request to a US-controlled inference endpoint is in the production request path. A vendor-prohibition rule, enforced by code review, lists permitted and prohibited providers explicitly.

For I9 (no biometric collection): a code-grep against the platform's source tree returns zero matches for biometric-handling library or API names; a runtime probe of the platform's data stores returns no biometric-shaped fields; a static test asserts that no biometric-handling library is imported anywhere in the platform's source. The architectural commitment is encoded in the vendor-prohibition rule (§13.1) and verified by code review on every change. Member-side device-local biometric unlock of a member-controlled credential vault — Apple Secure Enclave, Android StrongBox, hardware-token vaults — is permitted and architecturally invisible to the platform; the biometric never crosses the platform's boundary.

The threat model is not exhaustive. It is the model the architecture is *known* to defend, with named predicates the operator and external auditors can test. Threats not enumerated above (deniable-encryption attacks, side-channel attacks against the key store, supply-chain attacks against the framework Tractatus) are out of scope for this paper but are tracked in the project's operating discipline.

---

5. Design principles

5.1 Tenant isolation as foundational, not as feature

The architecture's first commitment is that tenant isolation is the foundational primitive. Every database query is filtered by `tenantId`. The filter is enforced by a database plugin that runs against an `AsyncLocalStorage` request context; queries outside request context (scheduled tasks, batch jobs) must explicitly opt out and document why. There is no platform-administrator role with cross-tenant content access; a platform-administrator user can create and manage tenants (infrastructure operations) but cannot read tenant content. This is not a configuration option — it is enforced in code, and any attempted cross-tenant access is treated as a security defect.

The discipline is durable. A single internal memory record, marked "never truncate", states the principle: *"Tenant isolation IS the product. Without it there is no sovereignty."* This is an internal engineering rule, enforced by code review and by automated tests that fail if a query is constructed without a tenant filter.

5.2 Sovereign-record metadata as a uniform schema

Every content model that participates in the sovereignty story carries the same metadata block, applied via a database plugin:

```
metadata: {
  origin: {
    author_did, kaitiaki_did, collective_id,
    tikanga_under_which_shared, created_at,
    provenance_hash, provenance_algorithm
  },
  policy: {
    share_within, share_exclude_jurisdictions, share_include_jurisdictions,
    collective_consent_required, collective_consent_body,
    train_flag, conflict_resolution_directive,
    delete_must_be_cryptographic, delete_propagates,
    expiry, individual_overrides_respected
  },
```

```
  encryption: { key_id, algorithm },
  proof_chain: [{ boundary_crossed, policy_evaluated_by, decision,
                  caveats_added, timestamp, algorithm,
                  signature, signer_did }],
  verification_cache: { verified_at, chain_hash_at_verify,
                        algorithms_verified, re_verify_after }
}
```

The schema is identical across the tenant-generated content models — Story, Poll, Event, Media, Album, Comment, ChatMessage, Deliberation, Correspondence, NewsPost, Resource, CommunityResource, ResourceBooking — and across an extended set of embedded surfaces (sub-document coverage of EventMenu, Edition, and similar). Each model's create path derives the origin via a shared helper; the plugin's pre-save hook computes the provenance hash and signs the create entry; the post-save hook caches the verification state; reads decorate every record with a verification field that surfaces cache freshness to consumers. The uniformity is the point: there is no per-model bespoke sovereignty implementation, and so no per-model sovereignty regression risk.

5.3 Cryptographic provenance with algorithm agility

Provenance is computed as SHA-256 over a canonical-JSON serialisation of the origin's required and optional fields. The algorithm is named in `provenance_algorithm`, so a future migration to a different cryptographic primitive (e.g., NIST post-quantum candidates) does not require a schema change — only a new entry-point in the same canonical form. Signature operations on proof-chain entries similarly carry their `algorithm` field; the platform's crypto-agility wrapper supports Ed25519 today and is structured to accept additional algorithms without call-site churn.

This is deliberate. Long-lived records outlive the cryptographic primitives that sign them. An architecture that hard-codes its primitive cannot honour a sovereignty claim that lasts longer than the primitive's lifetime.

5.4 Policy inheritance with effective-policy computation at the read boundary

Policy is not a single field; it is a hierarchy. Each tenant has a sovereign constitution declaring its defaults; each subgroup may override; each record's `metadata.policy` block may further specify. At read time, an effective policy is computed for the requesting member against the record's policy stack. The Policy Inheritance Engine performs this computation; the gate is enforced at the route boundary via per-list and per-detail invocations.

The engine is tested at multiple levels: per-rule unit tests, use-case validation against live local databases, and a discipline that tests prove wiring works in mocks but use cases prove it works in reality. This last commitment — internalised after an incident in which a substantial unit-test suite passed cleanly for a feature that was, in production, unwired — is documented in the project's operating discipline.

Three filtering options narrow the gate to specific access patterns: origin-only restricts reads to the record's authoring identifiers; group-scope restricts reads to members of the record's `collective_id` subgroup; unknown-scope strict mode fails CLOSED on any `share_within` value the gate does not recognise — defence in depth against misconfigured tenant constitutions or federation-imported records carrying scope values outside the platform's recognised set.

5.5 Bilateral federation in production

Federation in this paper's sense is the narrow technical arrangement of §2.4 and §4. Two tenants' constitutions agree on the federation's bounded purpose; both operators sign the

federation manifest; the data flow is direct between the two tenants; either may revoke at any time. The federation manifest itself is a sovereign record — it carries its own provenance, its own policy, its own proof chain, and its own verification cache.

Federation infrastructure is shipped end-to-end in the platform: the agreement model, the agreement service, the route surface, an administrator UI, an audit-log path, and a comprehensive negative-test matrix covering scope-bound reads, cross-tenant write blocking, audit-log completeness, citation discipline, caching/staleness, edge-case data states, authorisation boundaries, and namespace conflicts. Live federation links between independent tenants are pending the first multi-instance deployment; the bilateral pattern is built, the deployments are not. §7 reports the implementation in detail.

5.6 Member-driven sovereign portability

A member who wishes to leave their tenant — to migrate to another tenant operating under the same architectural model, to take their material to a different community, or to satisfy the GDPR Article 15 right of access — may do so via a canonical export. The export contains every record in which the member is the author, the kaitiaki, or otherwise named as a data subject; each record carries its proof chain forward; the receiving party may verify the chain against the source tenant's published DID document without trusting either operator. Records whose policy forbids export (e.g., a deliberation contributed under collective-consent terms) are listed in the export manifest as withheld, with the policy reason cited.

This is the architectural framing of GDPR Article 15: not a special-purpose access endpoint, but the same export pipeline the architecture uses for all sovereign-record movement, instantiated for the data-subject-as-member case. §8 reports the implementation, including the receiving-tenant ingest path that closes the migration loop.

---

6. Architectural implementation

This section reports the components that realise §5's design principles in code. Each is in production on both infrastructure sites (EU-sovereign OVH France; New Zealand-sovereign Catalyst Cloud) and verifiable from the codebase and the running API surface. Per the IP-perimeter posture (§13), this report describes architectural components and their interactions rather than per-file source paths.

6.1 Cryptographic provenance primitive

The provenance primitive computes SHA-256 over a canonical-JSON serialisation of the origin's required and optional fields. The entry-point produces the hash; a verifier re-computes and compares against the stored hash. The algorithm identifier travels with the record. A canonical-form helper eliminates hydration-mode-dependent enumeration — a failure mode surfaced during use-case validation, where a serialiser iterating an ORM subdocument's enumerable properties produced hashes that diverged across hydration modes, while a substantial unit-test suite on plain-object payloads passed cleanly. The fix was a single normalisation step; the discipline that surfaced it (use-case validation against live databases, not just mocked tests) is now part of the project's operating norms.

6.2 Proof-chain signing across creates, updates, and deletes

Every write to a sovereign-tagged record appends a signed entry to the record's proof chain. CREATE entries are signed by the tenant's proof-signing key (provisioned via the tenant key

store, §6.6) and bind the entry to the record's provenance hash. UPDATE entries on document-mode writes are emitted only when sovereign-relevant paths changed (excluding bookkeeping fields like `updatedAt`); the modified path list is captured. Query-mode UPDATE entries follow the same shape, computed from the diff between pre-image and post-image documents on `updateOne`, `updateMany`, `findOneAndUpdate`, and related paths. DELETE crossings are handled by two hook layers — a document-mode hook and a query-mode hook covering single, batch, and findAndDelete variants. Both layers produce a Tombstone record carrying the signed delete entry as the deletion's proof; query-mode tombstones are observably distinguishable from document-mode tombstones via the `policy_evaluated_by` field. A separate component extends this to the governance-queue model, ensuring governance-internal deletions also leave a signed cryptographic trace.

6.3 Verification caching and read-path integration

Verification of the proof chain happens at ingestion and caches to the record's verification-cache block: the time of last verification, the SHA-256 of the canonicalised proof chain at that time, the algorithms verified, and the next re-verify deadline (default 90 days; tenant-configurable via the constitution). The verifier exposes three entry points: an ingestion-time verify-and-cache, a synchronous read-time cache check, and a scheduled batch sweep.

The wiring is uniform. A pre-save hook computes provenance and signs the create entry. A post-save hook fires verify-and-cache after every write, debounced through an in-flight key set to prevent storm conditions. A scheduled task runs daily against the tenant-generated content models, processing expired cache entries in batches. The post-save hook fires on creates and on sovereign-relevant updates; a path-list filter excludes bookkeeping paths so the verifier's own writes do not loop the hook on themselves; bookkeeping-only saves skip the verifier and keep the re-verify cost bounded to genuine sovereign changes.

The read path completes the surface. Every API GET response on a sovereign-tagged record carries a verification field: compact `{valid, reason}` on list responses, verbose extras (verified-at, re-verify-after, algorithms-verified) on detail responses. The implementation is uniform: a single decorator helper is invoked from each route's lean and aggregate paths.

6.4 Policy Inheritance Engine and group-scope enforcement

The Policy Inheritance Engine reads from the tenant constitution, the requesting member's subgroup memberships, the record's policy block, and the operation requested (read/write/export/delete). It returns an effective policy with explicit violation reasons when a request fails. Three filtering options narrow the gate per §5.4.

Group-scope enforcement depends on records carrying a valid `collective_id`. A helper validates a caller-supplied subgroup identifier against three constraints (format, tenant scope, viewer membership) and returns an atomic context — atomic because `collective_id` without `share_within: ['group']` is purely ornamental (the gate would not enforce). Eight content-create paths (Poll, Event, Story, Album, Deliberation, ChatMessage, Carpool, Resource) accept the subgroup identifier from the request body and forward it through the helper. Backward compatibility is preserved: callers omitting the field create tenant-scoped records as before. Per-form UI pickers expose subgroup selection at the create-form level across the eight surfaces.

Unknown-scope strict mode closes a fail-open hole present in earlier iterations. A platform-wide set of recognised scope values defines the vocabulary; any value outside this set now denies with a named reason unless a recognised scope in the same set already grants access. This is the project's standing posture — *be honest about what cannot be verified; do not conjure permission for it* — applied to the read-path gate.

6.5 Sovereign constitution editor

A tenant's sovereign constitution is editable by the tenant administrator via a dedicated route and frontend. The editor exposes the constitutional defaults (default resolution mode, default export mode, default presumed authority, encryption model), a categories table that pins the canonical content models and allows tenant-defined custom categories, and multilingual support across English, German, French, Dutch, and te reo Māori. The te reo Māori translations were authored via the project's translation tooling (DeepL, which supports te reo Māori under language code `MI` — a fact regularly mis-assumed by external commentators and corrected within the project's own discipline) and spot-checked for sense.

A constitutional cutover window means tenants editing their constitution see a banner indicating that the change becomes binding only after the cutover; this is the constitutional-prerequisite gating that distinguishes a draft constitution from a binding one. A separate gate (the sovereign-constitution gate) hard-enforces 403 for tenants created after the cutover date that lack required sovereign sections, with built-in suspension immunity for designated platform-infrastructure tenants.

6.6 Tenant key store

Each tenant's encryption and signing keys live in a tenant-scoped key store with operations for generation, retrieval, rotation, and destruction. Keys are addressed by identifiers stored in each record's encryption block. Cryptographic deletion of a record — policy-gated by the `delete_must_be_cryptographic` flag — proceeds by destroying the per-record encryption key in the tenant key store, rendering the record's ciphertext unrecoverable from the persisted state.

6.7 Decentralised identifier publication

Tenant and member decentralised identifiers follow the W3C DID specification [11] and are published under the tenant's domain (`/.well-known/did.json` for the tenant DID document; `/.well-known/did/members/${slug}/did.json` optionally for member DIDs). DID documents contain the tenant's verification methods used to sign proof-chain entries; an external verifier holding the tenant's DID document can verify every signed entry a record carries, including entries on records exported under §8 to a different tenant.

6.8 Governance queue

The governance-queue model captures cases requiring tenant-arbiter decision: policy violations, conflict-resolution requests, member-initiated deletion requests that require governance approval. Lifecycle states — created → under review → decided → enacted (or rejected) — are transition-gated; each decision and each enactment leaves signed trail entries that the tenant can reconstruct from its own database. Deadline enforcement auto-enacts per the tenant's constitutional default resolution when an entry exceeds its grace period.

6.9 Export wrapper with non-admin visibility overlay and symmetric audit logging

Every sovereign-record export passes through a wrapper that gates on three conditions: every record carries provenance; every record belongs to the requesting tenant; full mode requires tenant-admin role. Hash and aggregate modes are now available to rank-and-file members through a visibility-overlay that filters records to the caller's read horizon before producing the projection — owner bypass; per-visibility-level rules; fail-secure on subgroup-preload errors. Violations write a governance-audit-log entry with an explanatory reason. Successful exports also write an audit entry with metadata capturing mode (full/hash/aggregate), record

count before and after filtering, per-model breakdown, and caller identity. Every export — successful or violating — is reconstructable from the audit log; no export is silent.

6.10 Uniform sovereign-record migration across the tenant-generated content models

The sovereign-record metadata block is applied uniformly across the tenant-generated content models. Migration was lazy where possible (records gain the metadata on first write under the new schema) and eager where necessary (a one-shot script populated provenance for the existing record stock). The same metadata block extends to embedded subdocument surfaces (EventMenu, Edition, and similar) under a uniform plugin extension. The verification cache is populated across the operational tenant set on both production sites; the small residual of legacy records without provenance hash are tagged `unverifiable` rather than `valid` — the architectural choice is to surface what cannot be verified rather than synthesise a cache for it.

6.11 Worker and WebSocket policy alignment

The asynchronous worker layer applies the tenant constitutional policy to records it creates. The two create-path workers in scope (email-to-content processing; document scanning) invoke a shared helper that assembles a policy context from the originating job's metadata (tenant identifier; originating member's identifier; originating subgroup identifier where applicable) and sets `metadata.origin` and `metadata.policy` on the record at create time. Workers that update existing sovereign records (OCR enrichment of uploaded contributions; media enhancement; story extraction; voice validation) preserve the policy set at upstream create-time and do not need the helper. Workers that produce no sovereign content (orchestrator coordination; queue scanning; transcription pipelines that mutate operational queue records) were audited in scope and confirmed not to require the helper. The WebSocket surface is wired to the same effective-policy computation through a per-recipient broadcast filter; before a chat message reaches a recipient socket, the visibility predicate is evaluated for that socket, and the message is omitted if the visibility check fails. Federation broadcasts pass through unchanged — federation visibility is decided at the federation service level, not at the broadcast level.

6.12 Proof-chain compaction primitive

A proof-chain compaction primitive replaces a contiguous sub-range of a record's proof chain with a single signed summary entry whose payload is the SHA-256 of the canonical-JSON of the replaced sub-chain. Verification of a compacted entry has two modes: a default cheap mode treats the compacted entry as a single signed step, anchored by the summary hash; a full mode retrieves the archived pre-compaction sub-chain and verifies entry-by-entry. The primitive is opt-in per tenant constitution; default is off. Application to live tenant proof chains is operator-paced.

6.13 Tombstone retrofit

A tombstone retrofit primitive signs pre-existing plain-text tombstones from before proof-chain signing was introduced, so the audit trail is uniform across the tenant's history. The retrofit operates per-tenant, idempotently and resumably, and adds a signed entry alongside the original plain-text fields without erasing them. The primitive is operator-paced; no production tenant tombstones have been retrofitted yet.

6.14 Framework consultation as audit trail

Every architectural decision in the platform's development is preceded by a framework consultation: a documented decision record naming the services consulted, the per-service condition list, and the verdict. Consultations are recorded on local plus EU-sovereign and NZ-sovereign production databases. The recording is automated by per-decision scripts; the documentary form is a per-decision markdown file under `docs/framework-consultations/`. The discipline of recording — three insertion sites per consultation so no single host's loss compromises the audit position — is the contribution; the value is reproducibility and auditability, not record count.

This is not virtue-signalling. The consultation is the project's mechanism for binding architectural decisions to observable artifacts: a future reader can ask, *which conditions did the read-path integration address?*, and the answer is in the database. The Tractatus framework's working paper [1] documents the pattern from the framework side; this paper documents an instance of it on the platform side, with the consultation ledger forming part of the evaluation surface (§12).

---

7. Bilateral federation in production

The bilateral federation pattern is built end-to-end, with a substantial verification surface; live federation links between independent tenant deployments remain pending. The architecture is ready for the first multi-instance deployment; the deployments are not yet established.

7.1 The federation manifest

A federation between two sovereign tenants is materialised as a federation-agreement record signed by both tenants. The agreement specifies the bounded purpose (carpool-ride-matching, shared-event-announcement, joint-deliberation, kaupapa co-stewardship, curriculum cross-domain reference), the data-flow shape (which fields cross which way, what transformation, what retention on each side), the cross-tenant policy resolution (which constitution governs records authored under the federation; how policy conflicts resolve), the revocation procedure (either party may revoke unilaterally; revocation is a signed record; propagation is immediate), and the audit retention (each tenant retains a signed copy of every cross-tenant interaction).

The manifest is itself a sovereign record. A federation cannot activate without verified signatures from both parties against their respective DID documents. Appendix C reports the schema shape at architectural-component level; specific implementation field-set details are held back per the IP-perimeter posture (§13).

7.2 Administrator UI and audit log

A tenant administrator manages federation agreements through a dedicated administrator UI that renders the federation lifecycle (proposed → accepted → active → revoked) and exposes the audit log. Every cross-boundary event — a federation proposal, an acceptance, a query routed across the federation, a revocation — leaves a signed entry in the federation audit log. The audit log is reconstructable on either side independently; neither tenant relies on the other's record-keeping for its own audit position.

7.3 Negative-test matrix

A negative-test matrix (under continuous-integration coverage) asserts the federation surface's invariants. Twelve categories are scaffolded: scope-bound reads (including a static

guarantee that no forbidden collection reference appears in the federation service code), cross-tenant write blocking, audit-log completeness, citation discipline, caching/staleness behaviour, edge-case data states (missing fields; null-versus-absent distinctions), authorisation boundary enforcement, and Phase-3 namespace conflict resolution. A subset of the matrix is walked by a live multi-tenant validator that exercises the full HTTP stack against a running deployment; the remainder run against a service-level fixture or as static code-grep assertions.

The most load-bearing test in the matrix is a static assertion: the federation service file is read as text, comments are stripped, and forbidden collection names are matched against the executable code. The assertion is encoded as a per-CI test, not a one-shot pre-commit check; any future widening of the federation read surface that introduces a forbidden collection reference fails the assertion at CI time.

7.4 Live deployment status

Live federation links between independent tenants are pending the first multi-instance deployment. The carpool federation — a class of federation connecting a set of communities for koha-only ride-matching — is the original target multi-instance deployment. A carpool tenant is in build on NZ-sovereign infrastructure (Catalyst Cloud); the multi-instance federation activates once at least two carpool tenants are operating. **Communities or organisations exploring multi-instance carpool deployment as a sovereign-infrastructure alternative — community-transport practitioners, transport-equity researchers, rural-resilience programmes, university sustainability or transport groups — are invited to contact the corresponding author regarding pilot participation.** Iwi-to-iwi federation deployments — where one iwi shares specific kaupapa material with another, by signed manifest, retaining full revocation control — are infrastructurally supported but operator-led; no live iwi-to-iwi federation has been activated at the time of this draft.

§5.5's framing of bilateral federation is therefore an *architectural commitment with shipped infrastructure and a verification surface*, not a *deployed network of live federations*. The architectural property at stake — that two communities may agree, on terms they specify, to a specific bounded interaction, and only that — is exactly what Te Tiriti's three articles imply for digital infrastructure: tribal sovereignty over taonga is honoured because each iwi retains full authority within its own tenant, and federation does not erode that authority — it permits a specific bounded interaction within the architecture's framework.

---

8. Sovereign portability — DSR integration

The architecture's sixth design commitment (§5.6) is that a member is a first-class data subject. A member who wishes to leave their tenant — to migrate to another tenant under the same architectural model, to take their material to a different community, or to satisfy a GDPR data subject right — may do so via a canonical export.

8.1 The canonical export bundle

A member-initiated canonical export contains every record in which the member is the author, the kaitiaki, or otherwise named as a data subject. The export is a paginated bundle across the tenant-generated sovereign-tagged content models, including embedded surfaces where applicable. Each record in the bundle carries its full proof chain, its full policy block, and its provenance hash. The bundle's manifest is itself signed by the source tenant; an external verifier holding the source tenant's DID document can verify every signed entry in the bundle

without trusting either operator. The bundle is rendered in JSON, CSV, or PDF form according to the request format; the underlying canonical content is identical across renderings.

8.2 Policy-respecting export and withheld-list manifest

The export wrapper enforces policy. Records whose policy forbids export (e.g., a deliberation contributed under collective-consent terms; a piece of media subject to a tikanga-specific share constraint) are listed in the export manifest as withheld, with the policy reason cited. The member receives both the bundle and the withheld-list; the withheld-list is itself signed, so the member has a verifiable artefact attesting to what was excluded and why. The discipline is full disclosure of what is withheld and why: an attempt to use a data-subject right as a pretext for accessing material the member has no legitimate right to is met by the policy gate, and the response itself is auditable.

The withheld-list mechanism is the architectural answer to a tension regulators have identified for years: the right of access in Article 15 is bounded by the rights of other identifiable persons (Article 15(4)) and by other legitimate-processing grounds. A standard implementation can either return everything (violating other parties' rights) or return less than asked (without explaining the basis for exclusion). The architecture's posture is that every excluded record is named, its policy reason is cited, and the manifest binding both pieces is verifiable.

8.3 Receiving-tenant ingest (cross-tenant migration)

A member may take their canonical export bundle to a different tenant operating under the same architectural model. The receiving tenant's import path verifies each record's proof chain against the source tenant's DID document, accepts the records (where the receiving tenant's constitution permits), and continues the proof chain forward — the receiving tenant signs an `ingest_via_migration` entry on each record, naming the source tenant and the bundle manifest hash. The member's identity is established by their cross-tenant DID; the migration is recorded on both sides as a normal sovereign-record event. A receiving-tenant constitutional acceptance check is integral, not optional: records whose source-tenant policy is incompatible with the receiving tenant's defaults (e.g., stricter privacy posture refusing more permissive sender policy) are listed as REJECTED with the policy reason. The migrated bundle's reception receipt is signed by the receiving tenant and returned to the member, providing a verifiable closure to the migration.

The current implementation of the receiving-tenant ingest path covers Phases A–F: source-DID resolution, bundle verification, constitutional acceptance check, ingest with proof-chain forward continuation, receipt signing, and end-to-end framework-consulted integration test scenarios. Identity reconciliation in the v1 implementation is configured to refuse auto-onboarding by default — a migrating member must already be a member of the receiving tenant, or the receiving-tenant administrator must approve the membership creation manually before the bundle ingests. This is a deliberate conservatism choice: auto-onboarding via cross-tenant DID has a security surface that warrants its own design pass, and the v1 implementation defers it.

8.4 GDPR Articles 15, 16, 17, 18, 20, 21

The DSR endpoint surface implements all six GDPR data-subject rights via the same export pipeline, with right-specific behaviours where required:

- **Article 15 (Right of Access):** the canonical export, as described in §8.1–§8.2.
- **Article 16 (Right to Rectification):** members may request correction; the request is policy-gated; accepted corrections leave a signed proof-chain entry.

- **Article 17 (Right to Erasure):** records authored solely by the member, where no other rights are implicated, may be cryptographically deleted on request — the per-record encryption key is destroyed in the tenant key store; the ciphertext becomes unrecoverable; a signed tombstone records the erasure. Records implicating other parties (a comment on another member's story; a contribution to a multi-author deliberation) follow the constitutional default for collective-consent erasure — the tenant's governance queue receives the request, the implicated parties are consulted per the tenant's process, and the resulting decision is enacted with full audit trail.
- **Article 18 (Right to Restriction):** restriction is implemented as a policy override that prevents processing while the request is pending; the override is itself a sovereign-record event.
- **Article 20 (Right to Data Portability):** the canonical bundle of §8.1, with the receiving-tenant ingest path of §8.3 as the architectural completion.
- **Article 21 (Right to Object):** objection is recorded on the relevant record and propagates through the policy gate as a per-record processing veto.

The 30-day Article 15 response window is enforced by an audit-log timer; missed responses trigger an alert in the tenant's governance queue.

8.5 The tension with Article 17 carve-outs

The architectural framing of the right-to-erasure tension between Article 17 and the Article 17(3) carve-outs (freedom of expression; legal claims) is not that the tension does not exist; the tension is real. The architecture frames the resolution explicitly, in the tenant's constitution, with policy-gated enactment, and with full audit trail. A member's erasure request is honoured to the extent the constitutional resolution permits; where collective-consent terms require multi-party process, the process is logged and the resulting decision (erase, redact, retain) is signed. An external auditor reading the audit log can reconstruct exactly which Article-17 carve-out was invoked, by whom, on what record, with what outcome.

---

9. Stakeholder governance UI

The governance UI exposes the platform's constitutional posture, communications discipline, decision history, framework consultation ledger, dialogue surface, and stakeholder review surface. It is a stakeholder-facing surface, deliberately kept readable by parish treasurers and community elders rather than only by engineers. The UI lives at a designated operations-hub tenant subdomain and is replicated to every tenant subdomain via a uniform pattern. Phases 1 through 7 are shipped as of this paper's date; Phase 6 (participatory dialogue) and Phase 7 (cross-product-type generalisation) extend the surface from read-only review to participatory governance.

9.1 Constitution viewer (Phase 1)

The Constitution viewer renders the platform's stable-anchored stakeholder-facing aggregation of the three primary sources (the project's hard rules, the never-truncate memory items, and the Layer 1 universal platform principles). The renderer is a markdown-page-loader pattern: no API route, no dynamic fetch, the viewer is a static HTML file that loads the markdown over HTTPS and renders it. This pattern is intentional: the viewer is the simplest possible artefact, auditable by any reviewer who can read HTML and markdown.

9.2 Comms Constitution viewer (Phase 2)

The Comms Constitution viewer follows the same pattern, exposing the operator-facing communications rulebook (channel stack, cadence, refusal lines, drafts-only-never-send rules). The current published version closes its operator-input items under operator-delegated agentic best-judgement; subsequent revisions await operator endorsement.

9.3 Decision-log viewer (Phase 2)

The Decision-log viewer renders a curated index of significant decisions across the architecture's development (architectural primitives, vendor posture, privacy and content rules, process discipline, training and AI). The Constitution viewer and Comms Constitution viewer both link to the Decision-log viewer in their Companions sections. The full stakeholder-readable arc — Constitution → Comms Constitution → Decision Log — is navigable on every tenant subdomain.

9.4 Framework consultation viewer (Phase 3)

The Framework consultation viewer exposes the consultation ledger via a stakeholder-readable HTML surface. The viewer presents an aggregated index by document reference (one row per architectural decision, with services consulted, conditions, verdicts, and dates) and a per-decision detail view exposing the full per-service condition list and verdict trail. The viewer is read-only; the underlying ledger is written by the platform's automated consultation recording scripts; the stakeholder reads but does not write.

9.5 Stakeholder guest-token access (Phase 4)

A stakeholder-specific guest session grants read-only access to the governance UI without requiring full tenant-member registration. A platform administrator issues a stakeholder invite; the invite is a signed record naming the stakeholder, the invited surfaces, and the invite expiry; the stakeholder accepts via a one-shot URL; the resulting session carries the same policy-gate posture as a member session, but with the read horizon bounded to the invited surfaces.

The architectural property is that stakeholder review is itself a sovereign-record interaction: every invite, every accept, every read is logged, signed, and reconstructable from the audit trail. A funder or policy reviewer who has reviewed the platform's governance UI can produce a verifiable record of what they reviewed, when, and against which version of the underlying material.

9.6 Stakeholder review surface (Phase 5)

A final review surface aggregates the governance UI material into a single navigable index for stakeholder consumption: a one-page entry that names every Constitution article, Decision-log entry, Comms Constitution rule, and recent framework consultation, with deep links into each. The surface is the natural endpoint of a Phase 4 invite; a stakeholder accepting the invite lands on the review surface and may navigate from there.

9.7 Participatory dialogue (Phase 6)

The Phase 6 dialogue surface converts the read-only governance UI into a participatory one. Stakeholders may comment on Constitution articles, Decision-log entries, and Comms-Constitution rules; comments are themselves sovereign records, with the same provenance, policy, proof-chain, and verification-cache machinery applied. The platform's situated-language layer answers stakeholder queries from the curated corpus that the governance

UI itself maintains, with the corpus serving as the citation surface. Hallucination defence is layered: a tightened system prompt biases the language model toward refusal-by-default for queries outside the corpus, and a citation-discipline filter rejects responses that fail to cite a corpus source.

9.8 Cross-product-type generalisation (Phase 7)

Phase 7 generalises the Phase 6 dialogue surface across the platform's product types. Each product type carries its own dialogue-corpus paths, vocabulary, and citation patterns. Phase 7.A delivers the per-product-type generalisation; Phase 7.B builds the shared universal-patterns dialogue corpus; Phase 7.C wires the inline approved-comment display surface across the mdsl viewer pages with dynamic anchors and a widget public API; Phase 7.D is the federation surface for the dialogue layer, which uses the same bilateral-federation infrastructure described in §7 (the negative-test matrix is shared, not separate; cross-Village stakeholder-comment federation is one specific application of the general bilateral pattern); Phase 7.E records the comms-constitution rule and ledger row that documents the surface's adoption.

The cumulative effect of Phases 1–7 is a stakeholder governance surface that is readable, auditable, navigable, participatory, federation-aware, and uniformly applied across the platform's product types — at the cost of a substantial verification surface (the bilateral-federation negative-test matrix being the largest single contributor) and the discipline overhead of running the framework-consultation recording on every architectural extension.

---

10. Worked example: cross-domain naming sovereignty between two situated language modules

This section illustrates the bilateral-federation pattern with a worked example, contributed by the corresponding author from his ongoing curriculum-design work. The example concerns curriculum integration in a primary-school setting, but the architectural pattern generalises to any pair of communities whose data-sovereignty postures intersect at a specific point of cross-domain authority.

10.1 The configuration

Consider two situated language modules, each operating as a sovereign tenant on the platform:

- **A botanical-knowledge module** for a regional flora — for instance, a Flora of New South Wales module, owned by a botanical institution responsible for the validated taxonomy, scientific names, distribution, ecological notes, and cross-references to scholarly sources. The module's owners maintain the content under a continuous-improvement process: new discoveries are validated and integrated; corrections are issued under signed authority; the corpus is the authoritative source for botanical reference within the module's scope.
- **A language-revitalisation module** for an Indigenous language in the same region — for instance, an Aboriginal Languages of New South Wales module, owned by a community-governed language authority responsible for the validated lexicon, pronunciation, etymology, cultural context, and ongoing revitalisation work. The module's owners maintain authority over the language and its uses, including how the language names entities in the natural world.

A point of cross-domain authority arises at the *naming of plants*. Each plant in the botanical module may carry — alongside its scientific name — an Indigenous name from the language

module's lexicon. Historically, these Indigenous names sat under the botanical module's control as bibliographic appendages. A political decision to restore language sovereignty transfers naming authority from the botanical module to the language module: from now on, the canonical Indigenous name for a plant is whatever the language module says it is.

10.2 Federation as the architectural answer

The architectural answer is a bilateral federation between the two modules, with a manifest that names exactly the bounded interaction:

- **Bounded purpose:** cross-domain naming reference. The botanical module may query the language module for the canonical Indigenous name of a plant, given a scientific binomial. The language module retains all authority over the name; the botanical module retains all authority over the scientific taxonomy.
- **Data flow:** a structured query from the botanical module to the language module names the scientific binomial; the response is the canonical Indigenous name (or `unknown` if the language module's corpus does not yet name that plant). The flow is *pull-on-demand*; no batch transfer is implied. The botanical module may cache responses with a tenant-configurable expiry.
- **Policy resolution:** the language module's constitution governs the response. If the language module's corpus is in revision and a name is provisionally pending, the response carries that status as a `caveats_added` field on the proof-chain entry; the botanical module surfaces the status to its consumers.
- **Revocation:** either party may revoke at any time. Revocation propagates immediately; the botanical module ceases querying; cached responses age out per their expiry. No data flow continues post-revocation.
- **Audit:** each query and each response leaves a signed proof-chain entry on both sides. Either party can reconstruct the full federation history from its own database.

10.3 The student's experience

A student exercising the curriculum prompts a question — *"What is the Aboriginal name for Eucalyptus camaldulensis?"* The platform's curriculum delivery routes the query to the botanical module (which holds the scientific binomial as the authoritative source) and the federation forwards the naming-resolution sub-query to the language module. The response is *curated from a combination of situated language modules, not from a frontier large language model.* The student sees the name, the language module's citation, and an indication that the answer is federation-provided — not because federation is technically interesting to a student, but because verifiability is a curriculum value.

The architectural property at stake is that the student gets a curated authoritative answer without an LLM in the loop. Hallucination is structurally precluded because no model is generating the response from a probability distribution over training data; the response is a federated query against a curated corpus governed by the rights-holder. Where the corpus does not have an answer, the federation returns `unknown` — the student is told the system does not know, a structurally accurate answer that a frontier model's confident confabulation cannot offer.

10.4 The architectural lessons

Three lessons travel from this worked example back into the architectural commitments of §5:

1. **Cross-domain sovereignty transfer is a federation operation.** When authority over a class of references shifts from one community to another (botanical → language module

on plant naming; iwi → kāhui on a shared kaupapa; parish → diocese on a shared event calendar), the architectural operation is signing a new federation manifest, not migrating data between tenants. The data stays where the rights-holder is; the federation tells the consumer where to query.

2. **Curriculum delivery via federated situated modules is a structurally distinct deployment from LLM-curated answers.** The curriculum-delivery use case is a strong empirical motivator for the architecture's bias against LLM-mediated content delivery: where the student is a learner, the answer should be authoritative, not probabilistic.
3. **The federation manifest's bounded-purpose field is load-bearing.** A naming-resolution federation does not authorise the botanical module to query the language module's whole corpus; it authorises only the named query shape. The platform's federation service refuses queries outside the manifest's stated shape. This is the architectural property that lets two sovereign communities federate over a specific bounded interaction without surrendering authority over anything else.

A range of related federation classes — between a museum's collection module and an Indigenous community's cultural-property module; between a regional council's planning module and a hapū's heritage-site module; between a school district's curriculum module and a community language module — share the same shape. The Flora ↔ Languages example is offered as the canonical illustration because it makes both the *data sovereignty* and *epistemic-authority transfer* dimensions visible simultaneously.

---

11. Six village-type configurations — examples from a template family

The architecture is expressed through a template model. A tenant configuration is not a one-off build; it is a template instantiation, where the template fixes the sovereign-record primitives, the policy-inheritance behaviour, the federation semantics, the stakeholder governance UI, and the governance-queue shape, and the tenant-specific configuration fixes what the template leaves open: the constitution, the membership structure, the subgroup topology, the multilingual locale, the situated-language-layer cohort, the vendor preferences within the vendor-prohibition envelope.

The operational value of the template model is that a new village-type configuration is a configuration exercise, not a rebuild. The architectural value is that a reviewer examining one village-type configuration is examining *the same architecture* that every other village-type configuration also runs; the sovereignty guarantees are uniform across the family because the template is uniform.

Village-type configurations are operational across NZ-sovereign infrastructure (Catalyst Cloud) and EU-sovereign infrastructure (OVH France) covering whānau, rūnanga, committee, kāhui-māori, governance, membership, family, episcopal/parish, and ops-hub product types, alongside per-product-type demonstration tenants. A carpool configuration is in build on NZ-sovereign infrastructure as the first contemplated multi-instance federation deployment; the invitation to communities or organisations interested in pilot participation is in §7.4. Specific tenant subdomain names are not enumerated in this paper and are intentionally redacted to reduce attack-surface exposure; they are available to legitimate reviewers via direct enquiry to the corresponding author. Each currently-operational village-type configuration authenticates its members and returns a 302 / 403 response to unauthenticated content requests — the operational signature of a live tenant.

Village type	Purpose (a selection from possible applications)
Whānau	Māori extended-family sites carrying intergenerational genealogical and oral-history material; Te Tiriti-aligned content sovereignty is structural, not added
Rūnanga	Iwi (tribal) council sites carrying minutes, committee decisions, and taonga descriptions that touch Te Tiriti rights directly; cross-iwi sharing by bilateral federation, not platform-wide disclosure
Committee	Deliberative body configurations for sporting federations, professional associations, and local societies where decision provenance matters — the deliberation content model's signed proof chain is built for exactly this
Kāhui Māori	Multi-iwi coordination sites where sharing is by bilateral federation between sovereign iwi instances; each iwi retains full authority; federation primitives are the architectural answer
Governance	Institutional bodies (community boards, school boards, parish councils) serving a mix of all-member work (minutes, resolutions) and confidential work (membership, draft material, committee deliberations) — the share-within policy with group-scope enforcement lets a single platform serve both
Membership	Nationally-affiliated bodies with local chapters — a national association whose branches need their own membership list and event calendar, federated with the national body for bigger interactions; the dominant structure for European *Vereine*, sport federations, professional bodies, and trade unions

Additional village-types in the template family — family (genealogy-focused family heritage sites, operational on EU-sovereign infrastructure), parish (local-church community configuration, operational as the *episcopal* product type on EU-sovereign infrastructure), business (member-directory-plus-notices configuration for small associations of traders, demonstration tenant on EU-sovereign infrastructure), and carpool (ride-matching configuration on NZ-sovereign infrastructure, in build, carrier of the contemplated first multi-instance federation deployment) — are concrete instances of the same template family. Demonstration tenants for conservation, alumni, clubs, family, episcopal, governance, and membership product types are operational on EU-sovereign infrastructure. The set is not a fixed list; the template's value is precisely that additional village-types can be configured without architectural change.

11.1 Situated language layer cohorts (forward reference to Paper B)

Each village-type configuration is paired with a *situated language layer* cohort — a per-tenant-type language model trained on the tenant's own content with strict training discipline. Five

Tier-1 cohorts are deployed in production (whānau, episcopal/parish, generic-community, family, business); four Tier-2 cohorts (conservation, diaspora, clubs, alumni) are designated and will be commissioned when the first tenant of each type is in deployment, per the project's discipline against aspirational training. The empirical findings — including a documented set of weight-modification experiments showing uniform degradation, the four "no-X" rules of training-data hygiene, the CPU-fallback inference architecture, and the per-cohort evaluation results — are reported separately in [Paper B forthcoming].

A pre-launch gate is in place at the platform level: tenant creation is blocked until explicit operator authorisation. The platform does not silently spin up tenants; the gate ensures that every operational tenant has gone through an authorisation step that is itself audit-logged. A separate sovereign-constitution gate enforces hard 403 for tenants created after 2026-05-01 that lack required sovereign sections; this gate is operational with built-in suspension immunity for designated platform-infrastructure tenants.

---

12. Evaluation

This section collects the evidence of the architecture's implementation into a single section. Three ledgers and one case-study are presented: the use-case verification ledger, the framework-consultation ledger, the deployment-and-verification snapshot, and the hydration-mode hash-stability case study from 2026-04-22.

12.1 Experimental setup

The platform runs in production on two infrastructure sites: an EU-sovereign deployment on OVH France (community.myfamilyhistory.digital and associated tenant subdomains under mysovereignty.digital and myfamilyhistory.digital) and a New Zealand-sovereign deployment on Catalyst Cloud (village-nz infrastructure at 202.49.243.176 serving the mysovereignty.digital tenant subdomains). Both sites run the same code at the same revision; mirror parity is maintained across two deploy targets plus a self-hosted Forgejo upstream. Database is per-site MongoDB with tenant-bounded queries enforced by a Mongoose plugin; runtime inference for the situated-language layer is hosted on a New Zealand-sovereign GPU (Catalyst A6000 during business hours, home-eGPU during off-hours) with automatic failover.

12.2 Use-case verification ledger

The use-case ledger demonstrates that each implemented component operates as designed against a live local database. The ledger covers the architectural components: provenance canonicalisation; the Policy Inheritance Engine and its filtering modes; verification caching; proof-chain signing including query-mode UPDATE and DELETE; the governance-queue tombstone path; DID publication; the export wrapper including the visibility overlay; constitutional prerequisites; the federation surface; sovereign-record migration across the tenant-generated content models and embedded subdocument coverage; group-scope wiring including the chat-thread surface; DSR canonical export and ingest paths; worker policy alignment; WebSocket policy adaptation; tombstone retrofit; proof-chain compaction; the access-gate surface; the per-tenant viewer pages. The ledger's PASS rate is at parity at the snapshot point; the script set is reproducible by an external reviewer with codebase access (Appendix B summarises the categories).

12.3 Framework consultation ledger

The framework-consultation ledger spans the architectural surface. Each record names the service consulted, the per-condition verdict, and the operational metadata (operation name, duration, result class). The active-service set covers the core Tractatus services (BoundaryEnforcer, ContextPressureMonitor, MetacognitiveVerifier, PluralisticDeliberationOrchestrator, CrossReferenceValidator, InstructionPersistenceClassifier) plus a wider set of decision-specific services accumulated as the architecture has grown (TractatusAuditRecorder, SovereigntyPrimacyEnforcer, PolicyCoherenceValidator, TenantIsolationValidator, AuditTrailVerifier, SchemaGuardian, PolicyDecisionOracle, TenantOwnerAuthority, PluralisticDeliberator, GovernanceOrchestrator). The ledger is recorded uniformly on local plus EU-sovereign and NZ-sovereign production databases — three insertion sites per consultation — so that no single host's loss compromises the audit position. A scheduled health check reports per-service consultation freshness against thresholds tuned for realistic working cadence (4 hours system-wide, 24 hours per-service); these thresholds replaced earlier 30-minute defaults that produced false-positive fade alerts on every short break in active development.

12.4 Deployment metrics

Deployment status at the snapshot point: both production sites report `/api/health` 200; the framework module's services report operational on both sites; verify-and-cache runs nightly across the tenant-generated content models; the Catalyst smoke-test surface (catalyst-operational) passes at full coverage; ESLint runs clean across the modified files in every deploy; mirror parity is maintained across `ovh`, `catalyst`, and `forgejo`. The smoke-test FAIL verdict observed during active maintenance windows is expected behaviour — the smoke test queries production endpoints which serve maintenance HTML during lockout — and is recovered to PASS once the maintenance window is lifted.

12.5 Verification cache observability

The verification cache is populated across the operational tenant set on both production sites and spans the eleven content types currently in active use. Records without provenance hash (a small residual of legacy records pre-dating the sovereign-record migration) are tagged `unverifiable` rather than `valid`. The architectural property is that the verification field on every API GET response surfaces the cache state to consumers; downstream auditing tools can infer the architecture's verification health by querying the API surface, without requiring database access. The substantive contribution is observability discipline, not record count.

12.6 Case study: the 2026-04-22 hydration-mode hash-stability bug

The most instructive empirical event during the architecture's development was the discovery of a hash-stability bug during use-case validation of the read-path integration of the verification cache. The canonical-form serialiser was iterating an ORM subdocument's enumerable properties — a pattern that worked correctly for plain-object payloads but surfaced internal ORM state for hydrated documents, producing hashes that diverged across hydration modes. Save-time hashes cached one value; read-time hashes computed another; every record after the deploy would have read `chain_hash_mismatch`. The fix was a single-line normalisation step. The bug had passed the unit-test suite cleanly because the tests mocked plain-object entries — the failure mode required real hydrated documents.

This is a load-bearing example for the operating discipline: tests prove wiring works in mocks, but use-case validation against a live database proves wiring works in reality. The discipline that tests prove what mocks reveal, and that use-case validation surfaces what mocks hide, is encoded in the project's operating discipline and is the reason a use-case validation script

set exists alongside the unit-test suite. After the fix, all existing production records were re-cached with the corrected canonical-form serialiser; no record was lost, no audit position was compromised, and the bug is the canonical example used in operating-discipline training.

12.7 Interpretation

The evaluation evidence supports a specific claim: the architecture is operational, observable, and auditable at the API surface across multiple sovereign infrastructure sites. The verification surface (use-case ledger, framework-consultation ledger, deployment metrics) is reproducible by any reviewer with codebase access; the hash-stability case study demonstrates that the operating discipline catches real failure modes that mocked tests miss. What the evaluation does *not* claim is that every threat in §4 is comprehensively defended — the architecture defends *named* invariants with *named* predicates; threats outside the model (deniable-encryption attacks; supply-chain compromise of the framework Tractatus; physical compromise of the inference-layer hardware) are out of scope and are tracked in the operating discipline as separate concerns.

---

13. Open-source posture

The open-source posture distinguishes two strands.

The **Tractatus framework** — the development-time governance machinery — is public, open source under the Apache 2.0 licence, and distributed at `codeberg.org/mysovereignty/tractatus-framework` [1]. Its working paper, code patterns, and metrics are reproducible by an external reviewer with access to a Claude-Code-class installation and the framework's pattern library.

The **platform codebase** — the runtime application — is in a module-by-module open-source release under the European Union Public Licence Version 1.2 (EUPL-1.2) [10]. Source files carry per-file EUPL-1.2 headers; the platform's most-recently-touched files (provenance, verification-cache, group-scope attribution, query-mode delete and update hooks, DSR canonical export, federation services, stakeholder UI components, worker policy helpers) carry the header. The repository-level licence is pending the constitution of a governance corporation under New Zealand law, with a democratically-elected Board and an Advisory Committee. The Board approves architectural changes that bear on the platform's open-source posture and stakeholder commitments; the Advisory Committee provides cultural and stakeholder advice (Māori cultural advice; minority-language community advice; FOSS-community advice). The Board has not yet been constituted; the per-file EUPL-1.2 headers are the in-flight open-source preparation, not the final repository-level state.

The module-by-module release path was deliberately chosen over wholesale repository release. The motivating concern is a class of large-language-model attack surface in which a wholesale source release of an internally-coupled platform exposes material whose threat model has not been reviewed at the module boundary — code that only resists certain attacks because it is not yet read by adversarial-corpus-trained models. A careful per-module release allows each module's threat model to be reviewed before publication, and limits the bleed of internal coupling into externally-depended-upon surfaces. The released modules to date — the core sovereign-record plugin, the Policy Inheritance Engine, the tenant key store, the Tractatus framework, components of the DSR pipeline — establish the architectural surface external reviewers can use; subsequent modules follow the same review-then-release cadence.

A Village Model Licence draft exists as a custom licence form intended to combine FOSS-typical permissions with specific community-protection clauses (no use for surveillance of communities; no use that would breach a community's data-sovereignty stipulations; no use

that would circumvent a tenant's stated CARE-Principles posture). The draft is awaiting formal legal review; pending the review's outcome, the platform's per-file licensing remains EUPL-1.2.

13.1 Vendor discipline

The platform uses no US-owned cloud, SaaS, or infrastructure dependency in its production request path. EU-sovereign hosting is OVH France; New Zealand-sovereign hosting is Catalyst Cloud (with Catalyst (NZ) Limited as the corporate entity); home-eGPU failover for off-hours inference is on a non-US graphics processing unit. Code repository hosting is split: a self-hosted Forgejo instance is the EU-sovereign primary remote, with mirrors to the OVH and Catalyst bare repositories. Payment processing uses Airwallex (NZ) Limited — US card-network rails are touched only per-transaction when the payer's card issuer is US-based, and only for that single transaction. Translation tooling uses DeepL (German entity, EU GDPR jurisdiction). This vendor discipline is enforced by an internal rule that lists permitted and prohibited providers explicitly; deviations require explicit project-level decision, not silent introduction.

No biometric data is collected by the platform. Member identity-verification at high-stakes operations is performed out-of-band (in-person introduction within the community; video introduction; paper-channel verification) or via the member-controlled decentralised-identifier signing key, never via biometric capture. The reasoning is structural and converging from four lines: biometric data is irrevocable, so a leak cannot be remedied by rotation; biometric data has three structural US-jurisdiction exposure paths (direct US-side capture at borders and visa interviews; US-cloud hosting subjected to CLOUD Act compulsion regardless of the data subject's nationality; future Enhanced Border Security Partnership arrangements contemplating direct-database-access to partner-country biometric repositories); Māori biometric and DNA data carries specific Te Tiriti / WAI 262 / WAI 2522 protection that becomes load-bearing once such data is platform-collected, and the Crown's exposure of that data to a foreign jurisdictional regime tests the Article-2 protection of taonga in ways the platform must not foreclose; and the most ergonomic biometric application-programming interfaces are operated by US-headquartered firms whose use would in any case violate the platform's vendor-prohibition rule. Refusing to collect biometric data architecturally removes the platform from this entire risk surface — the operator cannot be compelled to disclose what was never collected. Members who wish to use device-local biometric unlock of their own credential vault may do so on their own hardware; the biometric never crosses the platform's boundary, and the platform does not interfere with this pattern. The platform's own access surface — including the shipped sovereign access gate (§15), per-tenant enablement operator-paced — uses text passphrases (dice-words / EFF-wordlist style; high-entropy and rotatable) plus self-hosted proof-of-work bot-detection.

13.2 The IP perimeter

The publication posture distinguishes the *architectural shape* (publishable as paper content; published as open-source modules) from the *operational specifics* (held back for IP perimeter reasons). Held back: specific Tractatus framework consultation conditions per service (the catalogue is the framework's contribution); per-product-type vocabulary content beyond the high-level template family; per-tenant configuration specifics; specific federation manifest field-set details beyond the architectural shape in Appendix C. Published: the architectural primitives in the paper; the threat model and testable predicates; the high-level schema shapes; the limitations and failure modes (§15); the source modules per the module-by-module release plan.

---

14. The architectural contribution

The architecture is a response to a structural condition, not a competing product. The default community-platform model — US-owned, attention-extracting, ToS-revisable at will — is a particular architectural choice about where data sovereignty lives. An architectural choice can only be answered by an architectural alternative, not by ToS revisions or feature additions to existing platforms.

Four properties of the work bear on this.

The work is **operational**. The architecture is not a specification awaiting implementation. It is running, across multiple village-type configurations, on EU-sovereign and New Zealand-sovereign infrastructure. A reviewer can verify operational status via the API surface and verify every architectural decision via the persisted framework-consultation ledger. The use-case verification ledger covers the implemented architectural surface at parity. The substantive contribution is the discipline of recording — that the architecture produces audit artefacts observable from the outside without trusting the operator's word for them.

The work is **structurally portable**. The architecture does not assume Māori communities, te reo Māori, or Te Tiriti specifically. The same `metadata.origin.collective_id` field that lets a Māori community attribute a record to its rūnanga lets a Welsh community attribute a record to its parish, a Sámi community to its siida, a Sorbian community to its village. The situated-language-layer pattern is similarly portable: a Welsh-language layer trained on Welsh-language material under Welsh community authority answers Welsh queries with the same architectural posture as the Māori-language layer answers Māori queries. The architecture is a substrate, not a product.

The work is **federation-capable**. Bilateral federation infrastructure is shipped end-to-end with a comprehensive negative-test matrix covering scope, write-blocking, audit, citation discipline, caching, edge-cases, authorisation, and Phase-3 namespace separation. Live federation links between independent tenant deployments are pending the first multi-instance deployment; the architectural property — that two communities may agree, on terms they specify, to a specific bounded interaction, and only that — is exactly what Te Tiriti's three articles imply for digital infrastructure, and what minority-language communities in Europe need when their cross-community work bridges legal jurisdictions.

The work is **portability-respecting**. A member is a first-class data subject. They may export their full record set in cryptographically verifiable form and migrate it to any other tenant operating under the same architectural model. The export is symmetric with the GDPR Article 15 right of access; the migration is symmetric with the architectural commitment that exit is a first-class operation. A community model where exit is hard is a closed garden, regardless of the marketing language it uses; a community model where exit is architectural is what the work reported here makes available.

The architecture's substantive claim is that a community-scale platform can be built such that its sovereignty guarantees live at the record level and the tenant-infrastructure level, not at the operator's discretion. The default model depends on operator-granted, operator-revocable sovereignty; an architectural alternative — sovereignty as a property of the records and the tenant infrastructure — refuses that condition by construction. The work reported here is an instance proof that such an alternative can be built by a small team in New Zealand, on a small budget, and run in production for real communities, even as the regulatory environment continues to drift toward foreign-jurisdiction reach-through (the Enhanced Border Security Partnership being the live NZ instance).

---

15. Limitations and failure modes

Several items are in scope for the paper's claims but not deployed at the time of this draft:

- **Live carpool federation traffic.** The bilateral federation infrastructure is shipped end-to-end with a substantial verification surface, but no live federation between independent tenant deployments has been activated. The carpool-federation deployment, contemplated as the first multi-instance illustration, is operator-paced.
- **Tier-2 situated-language-layer cohorts.** Four Tier-2 cohorts are designated (conservation, diaspora, clubs, alumni) but commissioning is paused per the project's discipline against aspirational training: a cohort is not commissioned until the first tenant of that type is in deployment.
- **Full repository-level open-source release.** The platform's per-file EUPL-1.2 headers are in place; the module-by-module release is rolling; the repository-level licence is pending Board approval, and the Board itself is pending corporate constitution under New Zealand law.
- **Tiriti Compliance Statement v0.2 publication.** A v0.2 revision exists under operator-delegated agentic best-judgement; named publication requires explicit Dr Taiuru consent.
- **Village Model Licence formal legal review.** The draft exists; formal legal review is pending; pending its outcome, the platform's per-file licensing remains EUPL-1.2.
- **Receiving-tenant identity reconciliation auto-onboarding.** The current DSR migration ingest path refuses auto-onboarding by default; auto-onboarding via cross-tenant DID is a security surface that warrants its own design pass.
- **Sovereign access gate (passphrase + sovereign proof-of-work bot-detection + paper recovery codes) — shipped, per-tenant rollout operator-paced.** The platform's baseline authentication is httpOnly cookies plus tenant-context isolation enforced at the database query layer. The access-gate component is shipped end-to-end as a global request-pipeline middleware (`accessGate`) with a per-tenant `AccessGateConfig` (passphrase hash, rotation history, recovery-code count), ten REST endpoints (`/api/access-gate/{status, pow/{challenge,verify}, passphrase/verify, recovery/use, admin/{enable,disable,rotate,status,recovery-codes/pdf}}`), a self-hosted proof-of-work challenge/verify pair (no third-party bot-detection service), and paper-printable recovery codes generated as PDF on operator request (no SMS, no email, no out-of-band channel routed through US infrastructure). The component is disabled by default on every tenant; per-tenant rollout — passphrase issue, recovery-code distribution, member onboarding to the gate — is operator-paced and proceeds tenant-by-tenant under documented review. The architectural commitment to text-only and no-biometric authentication, as encoded in §13.1's vendor discipline and §4's invariant I9, is permanent and is enforced today by the platform's existing collect-no-biometric-data posture independently of the gate's per-tenant rollout state. Architectural rejections explicit to the gate's design — biometric authentication, SMS-based two-factor, email magic links via US-hosted providers, US-controlled push-OTP, US-controlled bot-detection services, behavioural biometrics — are documented in the access-gate plan-of-record with the reasoning that informed each rejection, so the choice surface is permanent rather than oversight. What remains operator-paced is the per-tenant enablement decision, not the component's existence.
- **AI-agent legal personhood — open question, engaged in Paper B.** Dr Taiuru (2026) [25a] poses, as an open inquiry, whether and under what conditions legal personhood might be extended to AI agents constituted by Māori knowledge, expressly deferring the determination to collective work among AI developers, government agencies, and Māori communities. The companion Paper B reports the situated-language-layer cohort training discipline that any future per-cohort partnership work would draw on; this paper does not pre-empt or prejudge that work.

Threats outside the §4 model are tracked separately in the operating discipline: deniable-encryption attacks against the key store; supply-chain compromise of the Tractatus framework or its dependencies; physical compromise of the inference-layer hardware; cryptographic primitive obsolescence outside the algorithm-agility wrapper's reach. None of these are addressed in this paper; all are real concerns at the implementation level and warrant their own analyses.

Two structural limitations are inherent rather than implementational. The architecture preserves community sovereignty over data; it does not by itself preserve community sovereignty over *cognition*. A community using the situated-language layer to mediate member queries is still using a language model; the model's training corpus, training discipline, and runtime behaviour are part of the architecture's surface, not separate from it. Paper B will document the empirical training discipline that makes the situated-language layer trustworthy for community use; its findings constrain the conclusions a reader can draw from this paper alone. The other structural limitation is that the architecture's defence against §4's adversary A1 (jurisdictionally-compelled host operator) is ultimately conditional on the tenant operating its own infrastructure or partnering with a sovereign-jurisdiction host: the architecture cannot create sovereignty where the host's jurisdiction does not support it, but it can preserve sovereignty for tenants whose hosts are themselves sovereign-jurisdictioned.

---

16. Conclusion

The architecture described here is implemented and running on EU-sovereign and New Zealand-sovereign infrastructure. Its core primitives are operational: tenant isolation as the foundational primitive; uniform sovereign-record metadata across the tenant-generated content models; cryptographic provenance with algorithm agility; policy inheritance with effective-policy gating at the read boundary; proof-chain signing across creates, updates, and deletes (both document-mode and query-mode); verification caching surfaced at read time; per-tenant key store with cryptographic-deletion finality; decentralised-identifier publication; governance queue with signed decision trail; export wrapper with non-admin visibility overlay and symmetric audit logging; bilateral federation with signed manifest; member-driven sovereign portability with Article-15-symmetric receiving-tenant ingest; per-tenant-type situated-language layer; stakeholder governance UI with read-only review surface (Phases 1–5) plus the shipped Phase-6 supervised participatory dialogue surface (operator-cleared editorial queue, draft-and-publish gate, no auto-publish), generalised across the platform's product types in Phase 7; worker and WebSocket policy alignment; proof-chain compaction primitive; tombstone retrofit primitive; and the shipped sovereign access gate (text passphrase + self-hosted proof-of-work bot-detection + paper recovery codes; per-tenant rollout operator-paced).

The framework-consultation ledger spans the architectural surface (each consultation recorded uniformly on local plus EU- and NZ-sovereign production databases); the use-case verification ledger covers the implemented architectural components at parity; bilateral federation infrastructure is shipped end-to-end with a comprehensive negative-test matrix (a subset additionally walked by a live multi-tenant validator), with live federation links between independent tenants pending the first multi-instance carpool activation; the Tier-1 situated-language-layer cohorts (whānau, episcopal/parish, generic-community, family, business) are deployed; the designated Tier-2 cohorts (conservation, diaspora, clubs, alumni) await the first tenant of each type before commissioning, on the project's discipline against aspirational training.

The companion paper (Paper B — Situated Language Layers for Minority-Language and Indigenous Communities, forthcoming) reports the empirical training discipline for the situated-language-layer cohorts: nine weight-modification experiments showing uniform degradation;

the four “no-X” rules of training-data hygiene; the CPU-fallback inference architecture; per-cohort evaluation results; and the substantive engagement with Dr Taiuru’s wider body of Māori AI governance work — both the Kaupapa Māori AI Framework [25b] (the prescriptive Te Tiriti grounded principles for AI consent, data sovereignty, and full-chain accountability) and the more recent inquiry [25a] (the open question of legal personhood for AI agents constituted by Māori knowledge, which Dr Taiuru expressly defers to collective work among AI developers, government agencies, and Māori communities). Paper B is the venue where the two registers Dr Taiuru distinguishes — the prescriptive governance duty and the interrogative personhood question — bear directly on the cohort discipline being reported. Paper B engages Mead’s Tikanga Test (tapu, mauri, take-utu-ea, whanaungatanga) as the evaluative surface that any future per-cohort partnership work would draw on; Paper B does not pre-empt that partnership work either, but it reports the cohort training discipline in the empirical detail such partnership work would require.

The paper is offered as a worked instance of how architectural sovereignty can answer Te Tiriti, AI-personhood (Dr Taiuru 2026), and EBSP-class jurisdictional pressures on a small team’s budget. Readers it is written for include: New Zealand iwi and rūnanga practitioners considering platform sovereignty under WAI 2522 and the EBSP negotiations; minority-language community technologists in DACH, Welsh, Sámi, Sorbian, Frisian, and Catalan contexts; Indigenous data-governance researchers and post-graduate students engaging the intersection of distributed systems, AI governance, and Indigenous data sovereignty; FOSS evaluators; community platform operators weighing sovereignty trade-offs; and policy reviewers considering NLnet, EU Commission, and Member-State funder positioning. Comments and corrections to the corresponding author are welcome.

Acknowledgements

The author is grateful to Leslie Stroh for foundational philosophical mentorship on pluralistic thinking and the question of goodness in artificial intelligence. The pluralistic-deliberation commitment that runs through the platform’s governance architecture — and the wider conviction that an AI substrate worth building must answer to a substantive notion of goodness, not a procedural one — owes its formative shape to those conversations.

The author also acknowledges Dr Karaitiana Taiuru’s cultural-safety review of the Tiriti Compliance Statement v0.1; named citation of subsequent revisions awaits his direct consent and is not asserted here. Reviewers of early drafts of the predecessor implementation report (v0.4) provided framing that has carried into this paper; their contributions are gratefully acknowledged.

Appendix A — Reproducibility

A reviewer wishing to inspect the architecture’s reproducibility surface can do so at the following levels.

The Tractatus framework is the fully-public component, distributed at `codeberg.org/mysovereignty/tractatu` under Apache 2.0. Its working paper documents the framework’s observational findings and the architectural patterns it codifies. A reviewer with access to a Claude-Code-class installation can reproduce the framework’s pattern library and replicate consultation recording on a local database.

The platform’s released modules — under EUPL-1.2 in the module-by-module release — establish the architectural surface external reviewers can use. The modules to date cover the

core sovereign-record plugin, the Policy Inheritance Engine, the tenant key store, components of the DSR pipeline, and the framework-consultation recording infrastructure.

Architectural components are described in this paper at the level of their interactions and contracts. Specific source paths (file names within the platform's source tree) are intentionally not enumerated; reproducibility at the file-and-line level is preserved via the released modules, while operational specifics (deploy pipeline, maintenance gates, hook smartening) are held back as engineering detail rather than research contribution.

Use-case verification scripts follow a `validate-use-cases-*` naming convention; framework-consultation recording scripts follow a `record-*-consultation` naming convention. Each consultation script produces an idempotent insertion of records on local plus EU-sovereign and NZ-sovereign production databases at a per-revision identifier.

Reproducer of the framework consultation pattern, from the Tractatus framework side, is documented in [1].

Appendix B — Use-case verification ledger snapshot

A current snapshot of the use-case ledger covers more than 45 distinct validation scripts. Each script asserts a named property of an architectural component against a live local database; the scripts are runnable independently and in aggregate. The categories below summarise the script set; per-script scenario counts and PASS/FAIL outcomes are included in the project's internal artefacts and are reproducible by an external reviewer with codebase access:

- Provenance canonicalisation and stability across hydration modes
- Policy Inheritance Engine: core resolution; gate-and-filter wiring; origin-only filtering; group-scope filtering; unknown-scope strict mode
- Verification caching: ingestion service; post-save hook + scheduled sweep; read-path integration; update-path post-save hook
- Tenant key store: lifecycle operations
- Proof-chain signing: CREATE consumer; UPDATE/DELETE document-mode; DELETE query-mode; UPDATE query-mode; governance-queue tombstone
- DID publication: tenant + member documents
- Governance queue wiring
- Export wrapper: per-mode behaviour; integration; success-path audit logging; visibility overlay for hash and aggregate modes
- Constitutional prerequisites and multilingual support
- Sovereign-record migration: across the tenant-generated top-level models; embedded subdocument coverage (NewsPost, Resource, EventMenu, Edition); sub-document Phases 1+2
- Group-scope wiring and cross-form attribution
- DSR canonical export: bundle assembly; manifest signing; truncation-flag honesty; receiving-tenant ingest end-to-end
- Worker policy gate: helper unit tests; per-worker integration (EmailProcessor, DocumentScanner)
- WebSocket policy adapter
- Tombstone retrofit
- Proof-chain compaction
- Federation surface: negative-test matrix across twelve categories, with a subset walked by a live multi-tenant validator

Appendix C — Federation manifest schema reference

A federation-agreement record carries the bilateral manifest at architectural-component level. The schema names: the agreement identifier; each party (tenant identifier, tenant decentralised-identifier, signature, signed-at timestamp); the bounded purpose (an enumeration: carpool ride matching; shared event announcement; joint deliberation; kaupapa co-stewardship; cross-domain naming reference; and others); the data-flow shape per direction (fields exposed; transformation applied; retention at the receiving end); the policy-resolution rule (which constitution governs; how policy conflicts resolve, including a per-field explicit table); the revocation procedure (either party unilateral; immediate propagation; both parties retain signed copy in audit); and the audit-retention identifiers (cross-tenant query-log records on each side). The manifest itself carries the standard sovereign-record metadata block (origin, policy, encryption, proof chain, verification cache); a federation does not activate without verified signatures from both parties against their respective DID documents.

Specific implementation field-set details beyond this architectural shape are held back per the IP-perimeter posture (§13.2). Reviewers requiring full schema specifics for compatibility evaluation may obtain them via direct enquiry to the corresponding author under appropriate confidentiality.

References

[1] Stroh, J. G. (2026). *Tractatus Framework — Architectural Patterns for AI Development Governance, Working Paper v0.2.* `codeberg.org/mysovereignty/tractatus-framework`. Apache 2.0.

[2] Stroh, J. G. (2026). *Sovereign AI Governance at Community Scale — An EU Policy Brief, v0.1.* My Digital Sovereignty Limited. DOI: 10.5281/zenodo.19635598. CC BY 4.0.

[3] Stroh, J. G. (2026). *Distributive Equity Through Structure — A Community-Scale Worked Example of Values Stickiness, v1.0.* My Digital Sovereignty Limited. DOI: 10.5281/zenodo.19600614. CC BY 4.0.

[4] Carroll, S. R., Garba, I., Figueroa-Rodríguez, O. L., Holbrook, J., Lovett, R., Materechera, S., Parsons, M., Raseroka, K., Rodriguez-Lonebear, D., Rowe, R., Sara, R., Walker, J. D., Anderson, J., & Hudson, M. (2020). The CARE Principles for Indigenous Data Governance. *Data Science Journal*, 19(1), 43. doi.org/10.5334/dsj-2020-043.

[5] Waitangi Tribunal. (2011). *Ko Aotearoa Tēnei: A Report into Claims Concerning New Zealand Law and Policy Affecting Māori Culture and Identity (WAI 262).* Legislation Direct, Wellington.

[6] European Commission. (2024). *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act).*

[7] European Commission. (2024). *Regulation (EU) 2024/1083 of the European Parliament and of the Council of 11 April 2024 establishing a common framework for media services in the internal market (European Media Freedom Act).*

[8] European Parliament & Council. (2016). *Regulation (EU) 2016/679 (General Data Protection Regulation),* Articles 9, 15, 16, 17, 18, 20, 21.

[9] U.S. Congress. (2018). *Clarifying Lawful Overseas Use of Data Act (CLOUD Act),* Pub. L. No. 115–141, Div. V (March 23, 2018).

[10] European Union Public Licence v1.2 (EUPL-1.2). https://joinup.ec.europa.eu/collection/eupl. Approved by the European Commission, 2017.

[11] World Wide Web Consortium. (2022). *Decentralized Identifiers (DIDs) v1.0 — Core Architecture, Data Model, and Representations.* W3C Recommendation.

[12] Stroh, J. G. (2026). *Sovereign-Record Architecture for Community-Scale Platforms — A Phase 1 Implementation Report, v0.4.* My Digital Sovereignty Limited (NZ). Predecessor draft to this paper; retained as historical record of the Phase 1 architecture state.

[13] Radio New Zealand / 1News. (2026, February). *MFAT confirms Enhanced Border Security Partnership discussions with United States.* On-record statement from the Ministry of Foreign Affairs and Trade.

[13a] Waitangi Tribunal. *Inquiry WAI 2522.* Two final reports relevant to this paper: (i) *Report on the Trans-Pacific Partnership Agreement* (2016); (ii) *The Report on the Comprehensive and Progressive Agreement for Trans-Pacific Partnership* (2021). A third report under the same WAI 2522 inquiry — *Report on the Crown's Review of the Plant Variety Rights Regime* (2020) — is adjacent but not cited here. All available at waitangitribunal.govt.nz.

[14] Centrist.nz. (2026). *New Zealand-United States border security agreement: status of negotiations.* Accessed via centrist.nz coverage of the EBSP discussions.

[15] Oceanic Press. (2026). *EBSP discussions: officials confirm scope and requirements under negotiation.*

[16] Privacy Foundation New Zealand. (2026). *Position statement on biometric data sharing with the United States under the Enhanced Border Security Partnership.* Privacy Foundation NZ media release.

[17] Biometric Update. (2026). *New Zealand considers US access to citizens' biometric and identity information under EBSP discussions.*

[18] Gunasekara, G. (2026). *Analysis of the Enhanced Border Security Partnership and DHS direct-database-access provisions.* University of Auckland legal commentary.

[19] Cochrane, T. (2024). *Should New Zealand seek a CLOUD Act-style executive agreement? Implications for digital privacy.* Privacy Commissioner-funded research.

[20] Snell, J., & Prodromou, E. (2018). *ActivityPub.* W3C Recommendation, 23 January 2018. https://www.w3.org/TR/activitypub/

[21] Bluesky Public Benefit Corporation. (2024). *AT Protocol specification.* https://atproto.com. Account portability + decentralised identifier (DID-based) handle resolution.

[22] Mansour, E., Sambra, A. V., Hawke, S., Zereba, M., Capadisli, S., Ghanem, A., Aboulnaga, A., & Berners-Lee, T. (2016). *A Demonstration of the Solid Platform for Social Web Applications.* Companion of the 25th International Conference on World Wide Web. Plus W3C Solid Community Group ongoing specification work at https://solidproject.org.

[23] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). *Communication-Efficient Learning of Deep Networks from Decentralized Data.* In *Artificial Intelligence and Statistics (AISTATS)*. The original federated-learning paper.

[24] Kairouz, P., et al. (2021). *Advances and Open Problems in Federated Learning. Foundations and Trends in Machine Learning*, 14(1-2), 1-210. Comprehensive survey of federated-learning architectural choices and open problems.

[25] Walter, M., & Suina, M. (2019). *Indigenous data, indigenous methodologies and indigenous data sovereignty. International Journal of Social Research Methodology*, 22(3), 233-243.

[25a] Taiuru, K. (2026, May 3). *AI Agents and Legal Personhood in New Zealand*. taiuru.co.nz/ai-agents-and-legal-personhood-in-new-zealand/. Accessed 2026-05-03. Personal-opinion piece (carries the author's explicit personal-capacity disclaimer). Poses the legal-personhood question as open inquiry, deferring determination to collective work among AI developers, government agencies, and Māori communities.

[25b] Taiuru, K. (2026, March 6). *Kaupapa Māori AI Framework — He Tangata, He Karetao, He Ātārangi*. taiuru.co.nz/kaupapa-maori-ai-framework/. Accessed 2026-05-04. Indigenous-Peoples AI framework grounded in Te Tiriti o Waitangi and UNDRIP; names Māori consent and data sovereignty over training material and full-chain accountability across developers, operators, and deployers as required practice.

[22b] Symmetry Systems. (2024). *Securing Your Sovereign Data+AI Stack.* Industry analysis on sovereign-AI architecture.

[23b] Merit Data Tech. *Zero-Egress AI: Architecting On-Premise Situated Language Models for Verifiable Data Sovereignty.* Industry analysis.

[24b] Enterprise DB. *Sovereign AI: Ensuring Data and AI Sovereignty in Enterprises.*

[26] Wachter, S., Mittelstadt, B., & Floridi, L. (2017). *Why a Right to Explanation of Automated Decision-Making Does Not Exist in the General Data Protection Regulation.* International Data Privacy Law, 7(2), 76–99. DOI: 10.1093/idpl/ipx005. Cited in §3.5.

[27] Edwards, L., & Veale, M. (2017). *Slave to the Algorithm? Why a 'Right to an Explanation' is Probably Not the Remedy You Are Looking For.* Duke Law & Technology Review, 16(1), 18–84. Available at scholarship.law.duke.edu/dltr/vol16/iss1/2. Cited in §3.5.

[28] Te Mana Raraunga (Māori Data Sovereignty Network). (2018, October). *Principles of Māori Data Sovereignty.* Retrieved from temanararaunga.maori.nz/principles-of-maori-data-sovereignty. Cited in §3.6.

[29] Carroll, S. R., Rodriguez-Lonebear, D., & Martinez, A. (2019). *Indigenous Data Governance: Strategies from United States Native Nations.* Data Science Journal, 18, 31. DOI: 10.5334/dsj-2019-031. Cited in §3.6.

[30] Hudson, M., Anderson, T., Dewes, T. K., Temara, P., Whaanga, H., & Roa, T. (2017). *"He Matapihi ki te Mana Raraunga" — Conceptualising Big Data through a Māori lens.* Available via Research Commons, University of Waikato. Cited in §3.6.

[31] Raman, A., Joglekar, S., De Cristofaro, E., Sastry, N., & Tyson, G. (2019). *Challenges in the Decentralised Web: The Mastodon Case.* In Proceedings of the Internet Measurement Conference 2019 (IMC '19), Amsterdam, October 2019. Empirical characterisation of the Mastodon federation graph, instance concentration, and operational fragility under instance-level moderation.

[32] Zignani, M., Gaito, S., & Rossi, G. P. (2018). *Follow the "Mastodon": Structure and Evolution of a Decentralized Online Social Network.* In Proceedings of the Twelfth International AAAI Conference on Web and Social Media (ICWSM 2018), Stanford, June 2018. Structural analysis of the early Mastodon network, including instance-level clustering and federation-graph properties.

[33] Open Data Institute. (2018, October). *Defining a "data trust".* ODI Working Paper. Establishes the working definition of a data trust as "a legal structure that provides independent stewardship of data," used by subsequent UK Government and policy literature on data-stewardship institutional design.

[34] Element AI / Nesta. (2019). *Data Trusts: A new tool for data governance.* Co-published institutional-design research on data trusts as a mechanism for addressing power asymmetries between technology firms, government, and the public.

Corresponding author: John G. Stroh, Director, My Digital Sovereignty Limited (NZ). ORCID: 0009-0005-2933-7170. Email: john.stroh@mysovereignty.digital.

Suggested citation (on operator approval): Stroh, J. G. (2026). *Sovereign-Record Architecture for Community-Scale Platforms — Paper A.* My Digital Sovereignty Limited. (Zenodo DOI to be assigned upon publication.)

Draft status: Review draft v3 — May 2026. Comments and corrections welcome. Voice anchored on the predecessor v0.4 implementation report. Structure adds Related Work, Threat Model, and Evaluation per Step C of the 2026-05-01 reposition plan. Companion paper (Paper B — Situated Language Layers) is forthcoming. Published at agenticgovernance.digital as review draft; Zenodo DOI to be assigned at v4 release-candidate stage.