ROOT ZERO VAULT

Structural Trust as a Decidable System

Profile B: Root Zero–Aligned Governance Specification

Hosameldeen Saleh
Root Zero Vault (RZV) / RSBIS Research & Specifications

Abstract

We define Structural Trust as a formally decidable property of governance systems: within a specified governed domain, every high-risk decision is (i) accepted or rejected by a terminating mediator, (ii) verifiable by independent parties, and (iii) recomputable offline from a finite continuity bundle without reliance on human discretion, vendor services, or mutable databases. We model Structural Trust using five jointly necessary components—Root, Authenticity, Ledger, Continuity, and Mediation—and state a conditional completeness theorem under an explicit enforcement boundary (the Governed Domain Assumption). We then specify a Root Zero profile with normative requirements for canonicalization, content identifiers (CVIDs), injective conversion rules for global Unicode and tenant-local coordinates, vault-and-deed authority resolution, deterministic reason codes, policy evolution via governed upgrades, and cryptographic agility via explicitly sealed transitions. A normative Continuity Bundle schema and worked policy examples demonstrate expressiveness without Turing completeness. Finally, we provide an evaluation plan and analytical size/time estimates that can be validated by a reference implementation. The contribution is a systems-level guarantee: enforceability and long-horizon auditability by construction, conditional on binding in-domain execution to mediation.

Keywords

decidable governance; auditability; canonicalization; continuity bundles; hash-chained journals; enforcement boundary; Root Zero; RSBIS

1. Introduction

Governance systems frequently record what should happen while remaining unable to prevent silent bypass, post-hoc reinterpretation, or dependence on live institutional authority. In high-risk domains—AI deployment, issuance of identity, regulated approvals—this gap converts governance into advisory policy.

Structural Trust targets a stronger outcome: within a governed domain, decisions are structurally mediated and independently verifiable from artifacts.

This document is a specification-grade research note. It does not claim novel cryptographic primitives or new string mathematics. Its novelty is compositional and operational: a terminating governance stack with offline recomputation as a first-class requirement.

2. Definitions and Enforcement Boundary

Definition 2.1 (Decision).

A decision is a request to perform a governed, high-risk digital action (e.g., deployment approval, issuance, transfer, policy upgrade).

Definition 2.2 (Structural Trust).

Structural Trust (ST) is the property that a decision's validity is decidable (ACCEPT/REJECT) by a terminating mediator, and that any party can recompute the same outcome offline from a finite continuity bundle containing the rules and evidence in force at the time of the decision.

Definition 2.3 (Governed Domain $\mathbb{D}$).

The governed domain $\mathbb{D}$ is the set of actions and interfaces that the institution binds to mediation. A system may have ST for $\mathbb{D}$ while leaving other actions out of scope.

Assumption 2.4 (Governed Domain Assumption, GDA).

GDA: Any action executed inside $\mathbb{D}$ MUST be routed through the mediator and MUST commit the mediator's outcome to the journal. Actions executed outside $\mathbb{D}$ are out of scope for ST claims.

2.1 Threat Model and Non-Goals

Threat model (in scope): (i) tampering with historical records, (ii) non-canonical encodings, (iii) replay/backdating/sequence manipulation, (iv) unauthorized actions attempted via in-domain interfaces, (v) discretionary reinterpretation of rules by operators.

Non-goals (out of scope): (i) preventing a malicious operator from executing an out-of-band action outside $\mathbb{D}$, (ii) guaranteeing moral correctness of policy, (iii) proving internal alignment of an AI model. Structural Trust is a governance and control claim conditional on enforcement boundary.

3. The Structural Trust Components

We define five components. Each is necessary; together they are sufficient for ST in $\mathbb{D}$ under GDA.

3.1 Root ($\mathcal{R}$) — Sovereign Origin and Scope Truth

Root defines the sovereign origin from which authority derives. In Root Zero, scope truth is resolved from vault YAML path and deed chain. Numeric substrings in display identifiers are explicitly treated as opaque labels and MUST NOT be used to infer governance ancestry or authority.

3.2 Authenticity ($\mathcal{A}$) — Canonical Bytes ↔ Content Identity

Authenticity holds when semantic content maps deterministically to canonical bytes, and canonical bytes deterministically map to a content identifier (CVID). Canonicalization profiles and canonical sort specifications eliminate representational ambiguity.

3.3 Ledger ($\mathcal{L}$) — Tamper-Evident Historical Induction

The ledger is an append-only hash-chained journal. Integrity is inductive: altering a prior decision requires rewriting all subsequent hashes.

3.4 Continuity ($\mathcal{C}$) — Offline Recomputation

Continuity holds when all predicates required to verify a decision are available offline in a finite continuity bundle: canonical rules and profiles, the event artifact and its canonical bytes/CVID, vault/deed chain content and CVIDs, coordinate conversion rules used, journal context, and required signatures/public keys.

3.5 Mediation ($\mathcal{M}$) — Terminating Enforcement (Non-Turing)

Mediation is a deterministic logic gate constrained to a terminating, non-Turing subset (finite predicates, acyclic rule dependencies). Mediation yields exactly ACCEPT or REJECT with a canonical reason code.

4. The Structural Trust Theorem (Conditional)

Theorem 4.1 (Structural Trust Completeness under GDA).

Assume GDA holds for governed domain $\mathbb{D}$. If the system implements $\{\mathcal{R}, \mathcal{A}, \mathcal{L}, \mathcal{C}, \mathcal{M}\}$, then for any decision D executed in $\mathbb{D}$: (i) validity is decidable (ACCEPT/REJECT), (ii) validity is recomputable

offline from a finite continuity bundle, (iii) any invalid decision presented to the mediator is rejected, and (iv) any accepted decision produces tamper-evident evidence.

Proof (sketch).

Necessity: Without $\mathscr{R}$, scope truth is undefined; without $\mathscr{A}$, semantics are ambiguous; without $\mathscr{L}$, history is mutable; without $\mathscr{C}$, verification depends on live authority; without $\mathscr{M}$, rules are advisory. Sufficiency follows from canonical bytes/CVID determinism, deterministic authority resolution, terminating mediation, hash-chained records, and offline bundle completeness. ■

Corollary 4.2 (Basement Calculator Criterion).

Given a continuity bundle containing the rules in force, canonical bytes/CVIDs, vault/deed chain context, journal linkage, and public verification material, a verifier can recompute ACCEPT/REJECT offline using only basic computation.

5. Root Zero Profile (Normative Specification)

5.1 Canonicalization and CVIDs

Root Zero defines canonicalization profiles (e.g., ROOTZERO.CANON.YAML.1) and canonical sort specifications. CVIDs are computed from canonical bytes and expressed as cvid:<hashAlg>:<hex> (e.g., cvid:blake3:<hex>).

5.2 Multi-Coordinate Conversion Rules (Injective, Reversible)

Root Zero uses a multi-coordinate model: global coordinate G_1 (mandatory) for canonical cross-script ordering, and local coordinate G_2 (optional) under a tenant-published alphabet.

Injectivity and reversibility for variable-length strings are obtained by a zero-free digit encoding (shift digits by +1; expand base by +1).

Global G_1: Normalize to NFC. Let B=1,114,112 (0x110000). For each code point cp_i set $d_i=cp_i+1 \in \{1,\ldots,B\}$. Base b=B+1. For $cp_0 \ldots cp_{n-1}$ in logical order: $G_1(ID)=\Sigma\, d_i \cdot b^{(n-1-i)}$. Decode by repeated div/mod by b and subtract 1.

Local G_2: Require published Σ and bijection $\mu:\Sigma\to\{0,\ldots,k-1\}$, $k=|\Sigma|$. Use digits $\mu(s_i)+1$ and base (k+1). Decode analogously. Local coordinates are valid only if Σ and μ are published and fixed at issuance.

5.3 Authority Resolution (Digits are Opaque)

Authority truth is resolved from vault path and deed chain. Display identifiers (including numeric blocks) are labels; they carry no legal authority by themselves.

5.4 Mediation Logic and Policy Complexity

Non-Turing mediation supports complex policies via finite predicates over canonical artifacts and resolved state (e.g., membership sets, signatures present, turn-order monotonicity, stage constraints). Cycles or unbounded computations are rejected at publish time.

5.5 Policy Evolution (Governed Upgrades)

Canonical rules evolve only via a governed policy-upgrade event. The upgrade declares a new profile identifier, includes CVIDs of the new rules, seals prior profiles for historical verification, and records effective-from conditions. Old decisions remain verifiable because bundles include the profile in force.

5.6 Cryptographic Agility (Sealed Transitions)

Migration between signature primitives occurs via sealed transitions recorded as events. Dual-signature transitions require both signatures; omission is rejection. Deprecation windows may require new events to use successor policies while maintaining verification of legacy events.

5.7 Canonical Reject Codes

Validators MUST return canonical reject codes with zero discretion. Example taxonomy: E-PARSE, E-CANON, E-CVID, E-CHAIN, E-SIG, E-CONV, E-STAGE, E-TURN, E-RULE, E-JOURNAL, E-REG.

5.8 Worked Policy Patterns (Non-Turing Expressiveness Examples)

This section provides concrete policy patterns that fit within a terminating, predicate-based mediation model. The examples are intentionally declarative: each rule is a finite predicate over canonical artifacts and resolved state.

Pattern A — Multi-signature + stage gate + strict turn-order

Predicates:
P1: stage == DEPLOY
P2: sig_present(KeyA) ∧ sig_present(KeyB)
P3: turn_is_strictly_increasing(scope=/RootZero/Deployments/PROD)

Decision:
ACCEPT iff P1 ∧ P2 ∧ P3 else REJECT with first failing reason code (E-STAGE/E-SIG/E-TURN).

Pattern B — Dual-signature migration window (sealed transition)

Predicates:
P1: now < transition_deadline
P2: sig_present(Ed25519Key) ∧ sig_present(PQCKey)
P3: policy_profile == TRANSITION.DUALSIG.1
Decision:
During the transition window, ACCEPT iff P2 ∧ P3 else REJECT (E-SIG or E-RULE). After deadline, policy upgrades to PQC-only via a governed upgrade event.

Pattern C — Explicit override allowed only under root-authorized exception

Predicates:
P1: override_requested == true
P2: override_allowed_by_profile == true
P3: sig_present(RootExceptionKey)
P4: override_record_includes_justification_cvid == true
Decision:
If P1, ACCEPT iff P2 ∧ P3 ∧ P4 else REJECT (E-RULE/E-SIG/E-CVID). If not P1, evaluate standard predicates. This preserves decidability while acknowledging real-world exception handling.

6. Continuity Bundle (Normative Schema)

A Continuity Bundle is the minimal finite artifact set sufficient for offline recomputation of ACCEPT/REJECT and reason code. Bundles MUST be self-contained: no external databases, vendor services, or network calls are permitted for verification.

6.1 Schema Overview

Normative requirement: A bundle MUST enable a verifier to (i) reconstruct canonical bytes for the decision artifact, (ii) recompute its CVID, (iii) resolve scope truth from vault/deed chain, (iv) recompute coordinates where required, (v) verify signatures and policy predicates, and (vi) recompute the mediator outcome deterministically.

Field	Type	Required	Purpose / Constraint

bundle_version	string	MUST	Bundle schema identifier (e.g., RZ.CONTINUITY.1).
profile_id	string	MUST	Canonical rule profile in force for the decision.
profiles	object	MUST	CVIDs (and optionally embedded texts) for canonicalization profile, sort spec, conversion rules, and validator constraints.
event_artifact	bytes or text	MUST	Decision/event artifact in verifier-readable form (e.g., YAML/JSON).
event_canon_bytes	bytes	MUST	Canonical bytes of event_artifact under the stated profile.
event_cvid	string	MUST	CVID computed from event_canon_bytes.
vault_path	string	MUST	Resolved legal scope path.
deed_chain	list<object>	MUST	Ordered deeds sufficient to resolve authority; each entry includes its CVID and signatures.
journal_context	object	MUST	At minimum: prev_entry_hash, entry_hash; include

			any turn-order state required.
reason_code	string	MUST	Recorded mediator reason code; offline recomputation MUST match.
outcome	enum	MUST	Recorded outcome ACCEPT/REJECT; offline recomputation MUST match.
public_keys	list<object>	MUST	Public keys required to verify signatures for this decision.
signatures	list<object>	MUST	Signatures required by policy over event_cvid (and other signed components if required).
coordinate_proofs	object	MAY	Optional recomputation traces for G_1/G_2; if present MUST verify.
overrides	object	MAY	If override is used, MUST include explicit override authorization and proof that override is allowed by profile.
implementation_fingerprint	object	SHOULD	Optional: validator build ID/runtime fingerprint for dispute resolution; not

required if profiles suffice.

6.2 Bundle Invariants (Verifier MUST enforce)

Invariant I1 (CVID determinism): event_cvid MUST equal CVID(event_canon_bytes).
Invariant I2 (Profile determinism): profiles MUST identify the exact canonicalization, sort, and conversion rules used.
Invariant I3 (Authority determinism): deed_chain + vault_path MUST be sufficient to resolve authority without external state.
Invariant I4 (Outcome determinism): offline recomputation MUST match outcome and reason_code; mismatch indicates tampering, divergence, or unauthorized override.
Invariant I5 (No hidden dependencies): verification MUST not require network access or mutable databases.

6.3 Bundle Profiles (Minimal vs. Forensic)

To address size concerns, Root Zero supports bundle profiles:
• Minimal Bundle: includes required fields only; embeds profile CVIDs and references profile texts by CVID (retrieved only if already archived).
• Forensic Bundle: embeds the full canonical rule texts, full deed chain artifacts, coordinate proofs, and optional implementation_fingerprint for courtroom-grade replay.
Both profiles remain self-contained relative to the verifier's archive: no network calls are required.

7. Worked Example (End-to-End Recomputation)

This example is illustrative and shows how a verifier recomputes an outcome offline from a continuity bundle.

7.1 Scenario

Decision: approve deployment of model M to environment PROD under scope /RootZero/Deployments/PROD.
Policy excerpt: required signatures {KeyA, KeyB}; stage==DEPLOY; turn strictly increasing; canonicalization profile ROOTZERO.CANON.YAML.1.

7.2 Offline Verification Steps

1) Canonicalize event artifact → event_canon_bytes.
2) Compute event_cvid.
3) Verify required signatures over event_cvid. Missing => REJECT (E-SIG).
4) Resolve authority from deed_chain and vault_path. Invalid => REJECT (E-CHAIN).
5) Verify stage and turn predicates using journal_context. Violations => REJECT (E-STAGE or E-TURN).
6) If all predicates pass => ACCEPT; record entry_hash and signatures.

8. Evaluation Plan and Analytical Benchmarks

This section provides (i) a concrete measurement protocol suitable for a reference implementation and (ii) analytical size/time estimates. These numbers are not claims of observed performance unless a referenced implementation reports them; they are bounded estimates intended to guide engineering.

8.1 Measurement Protocol (Reference Implementation)

A reference implementation SHOULD report:
M1: Canonicalization time and output size for event artifacts across a representative corpus.
M2: CVID hashing time (per hash algorithm).
M3: Signature verification time per signature type and per key count.
M4: Bundle size distribution under Minimal vs Forensic profiles.
M5: End-to-end verification time (ACCEPT path and REJECT path) on a commodity laptop.
All measurements MUST report hardware/OS/build details and corpus characteristics.

8.2 Analytical Bundle Size Estimates

Bundle size is dominated by (i) event_artifact/event_canon_bytes, (ii) deed_chain embeddings, and (iii) signatures/public keys. Let |E| be canonical bytes length, |D| be total embedded deed artifacts, and s be signature count. As a rough bound:
BundleBytes ≈ O(|E| + |D|) + O(s·SigBytes) + O(k·KeyBytes) + overhead.
Where SigBytes is typically tens to a few hundred bytes depending on scheme; KeyBytes is similarly bounded.

Bundle Profile	Typical Contents	Illustrative Size Drivers	Expected Range (Analytical)

Minimal	Required fields only; profile texts by CVID reference	Mostly |E| + signatures + small deed refs	≈ 2–50 KB (depends on |E| and signature count)
Standard	Minimal + embed rule texts; embed minimal deed chain	|E| + |Rules| + subset of |D|	≈ 20–500 KB (depends on embedded artifacts)
Forensic	Standard + full deed chain + coordinate proofs + optional fingerprints	|E| + |Rules| + full |D|	≈ 0.2–10 MB (depends on deed chain depth/artifact size)

Note: these ranges are intentionally broad; a reference implementation should publish real distributions for the targeted deployment domain.

8.3 Analytical Verification Time

End-to-end verification time is the sum of canonicalization + hashing + signature verifications + predicate evaluations. In typical deployments, hashing and signature verification dominate. Predicate evaluation is linear in predicate count under acyclic dependencies.

9. Related Work and Positioning

Root Zero composes known primitives (canonicalization, content addressing, hash chaining, signatures) into a terminating governance stack with offline recomputation as a first-class requirement. The novelty claim is not a new primitive; it is a profile that makes governance outcomes portable and replayable from bundles.

9.1 Transparency Logs (Certificate Transparency)

Certificate Transparency (CT) defines append-only public logs for certificates and proofs of inclusion/consistency. CT improves auditability of issuance but does not by itself define a general governance mediator for arbitrary decisions. Root Zero borrows the audit lens (public verifiability) but targets decision mediation and portable replay for institutional governance rather than certificate issuance alone.

9.2 Blockchains and Consensus Ledgers

Blockchains provide shared state via consensus and typically encode validity as a function of a virtual machine or script system. Root Zero does not require consensus to decide validity: validity is computed deterministically from published profiles and bundle contents. Where blockchains emphasize global shared state, Root Zero emphasizes verifiable local governance with portable proof artifacts.

9.3 Git-Based Governance and Signed Histories

Git's content-addressed objects and signed commits enable traceability, and recent research explores append-only reference logs for policy enforcement in repositories. However, Git workflows often rely on process conventions and tooling rather than a terminating mediator with canonical reject codes and an explicit continuity bundle contract. Root Zero can interoperate with Git-based workflows by treating Git objects as artifacts whose canonical bytes and CVIDs are governed by Root Zero profiles.

10. Guarantees and Limitations

Guarantees (within $\mathbb{D}$ under GDA): deterministic ACCEPT/REJECT with reason codes; tamper-evident historical record; offline recomputation from continuity bundles; structural rejection of rule-violating inputs presented to the mediator.

Limitations: cannot prevent out-of-band actions executed outside $\mathbb{D}$; cannot guarantee moral correctness of policy; does not prove internal AI alignment.

11. Conclusion

Structural Trust is a conditional system property: when high-risk actions are bound to a terminating mediator and recorded in a hash-chained ledger with continuity bundles, validity becomes decidable, non-discretionary, and independently recomputable for decades. Root Zero specifies one such profile, fixing canonicalization, CVIDs, injective Unicode/tenant coordinates, vault/deed authority resolution, policy upgrades, cryptographic agility, and a normative continuity bundle contract.

References

• Unicode Consortium. Unicode Standard Annex #15: Unicode Normalization Forms (NFC).

• IETF. RFC 8032: Edwards-Curve Digital Signature Algorithm (EdDSA).

• IETF. RFC 8785: JSON Canonicalization Scheme (JCS).

• BLAKE3 authors. The BLAKE3 Hashing Framework.

• IETF. RFC 6962: Certificate Transparency (June 2013).

• IETF. RFC 9162: Certificate Transparency Version 2.0 (December 2021).

• Yelgundhalli, S., et al. Rethinking Trust in Forge-Based Git Security (gittuf) (NDSS 2025).

• NIST / FIPS 204: ML-DSA (post-quantum digital signature standard).

• Saleh, H. Root Zero Deed (Complete) — RootZero0200–0220 (internal).

• Saleh, H. Root Zero–Aligned Complete Technical Specification (Master Note) (internal).