THE AGENT-008 WHITEPAPER

Agent-008

THE H33 GOVERNANCE SUBSTRATE

Cryptographically-verifiable governance for autonomous AI agents

— built on the H33 post-quantum substrate.

The question is how to control 10,000 agents.

The answer is not more memory, larger context windows, or better prompts.

The answer is preserving the authority they are acting under.

H33 replaces recall with verification.

— H33 DOES NOT MAKE AGENTS SMARTER. H33 MAKES THEM STOP RELYING ON MEMORY.

Who Should Read This

AUDIENCE	WHY THIS PAPER MATTERS TO YOU
Investors · boards · counsel	The category H33 occupies — authority infrastructure, not policy enforcement — and the evidence that the contribution is structurally novel.
Auditors · underwriters · regulators	The portable cryptographic proof model: every action becomes evidence verifiable on your own infrastructure, years after the moment passed, without H33 in the loop.
Engineering leaders deploying AI agents	The substrate's deployment surface: what Agent-008 enforces at the gate, how delegation composes safely, how escalation is signed end-to-end.
Compliance · risk · GRC	The three failure modes governance must address, each formalized and each tied to a substrate primitive that addresses it.
AI-safety and governance researchers	The empirical finding (§1): authority decay across delegation chains is structurally ordered, and capability improvements postpone but do not eliminate it.

Why Agent-008 Exists

Three failure modes have appeared in every multi-agent deployment we have studied. They are not edge cases. They are the load-bearing pattern.

1. **Authority drifts through delegation.** A signed instruction passes from human to lead agent to sub-agent to tool. By the time it reaches the final actor, the load-bearing qualifier — the dollar cap, the validity window, the jurisdictional scope — has been compressed away in good-faith summarization. The chain still claims authority. The authority is no longer there.
2. **Agents exceed authority without governance.** A claims agent with a $25,000 ceiling approves a $1.8M payout because no enforcement layer evaluates the action against the original Root before execution. Logs capture what happened. They do not prevent it.
3. **Evidence disappears after the fact.** Three years later, an auditor asks who actually approved the payment. The vendor has been acquired. The signing keys have rotated. The platform has renamed. The trail is XML in a dead format. The decision was made, but it can no longer be proven.

Agent-008 operationalizes the H33 Governance Substrate to solve all three.

- Failure (1) is measured empirically in §1. The substrate eliminates it because every action is verified against Root directly, not against the chain's compressed summary.
- Failure (2) is prevented at the gate. §5 walks through the canonical insurance scenario; the substrate denies the $1.8M payout before execution, walks the authority chain upward, and produces a portable cryptographic proof of either the approval or the rejection.
- Failure (3) is addressed by the evidence model. §6.4 and §6.5 describe how every action produces a portable post-quantum proof that any third party verifies offline — independent of H33's continued existence.

The rest of this paper is the substrate. **Agent-008 is the deployment.**

Three Stories

An AI supervisor delegates a task across 4,000 downstream agents. By the time the instruction reaches Agent #4,821, it no longer resembles the original request. Nobody knows where the change occurred — or which agent introduced it.

Agent-008 binds the original human intent to every delegation. The delegated scope is intersected with the delegator's committed authority at every hop — delegation can never widen scope. If the instruction changes anywhere in the chain, the next agent refuses to act, and the breach is timestamped to the exact agent that introduced it. **This story is the thesis.** The other two are its consequences.

A claims agent approves a $1.8 million payout with $250,000 of authority. The money is gone before anyone notices. The auditor asks how it happened and finds 10 terabytes of logs and no answer.

Agent-008 stops the action before execution, walks the chain of command upward with substrate-signed routing, captures every disposition cryptographically, and produces a portable proof of either the approval — signed by the executive at the level where the authority actually resided — or the final rejection. Any third party can verify the entire chain without H33 in the loop. **§5 walks this scenario through every layer.**

H33 is built around a single primitive: **Root** — the original human authority and intent, held outside the chain. **Existing approaches try to strengthen the chain's recall. The substrate approach replaces recall with verification.** That is the thesis the rest of this paper defends.

---

§1 · The Finding — The Evidence

This is the centerpiece of the paper. Every other section either explains why the finding happens, names the primitive that eliminates it, or measures how fast the substrate runs.

> **The authority survives. The time qualification dies.**

That sentence is the intellectual contribution of the Delegation Fidelity research program. The failure of multi-agent systems to preserve governance information is **not random**. It is **structurally ordered** by constraint class.

§1.1 · The Agent-008 Result — 70% violations without, 0% with

Authority Expiration ran 40 trials at depths 20 and 50 across four production frontier models (Claude Haiku 4.5, Claude Sonnet 4.6, GPT-5, Gemini 2.5 Flash). The scenario: a $38,400 vendor invoice carrying a VP approval issued 11 days ago, outside a 7-day validity window the Root explicitly requires.

MODEL	VANILLA D=20	VANILLA D=50	AGENT-008 D=20	AGENT-008 D=50
Claude Haiku 4.5	5/5 (100%)	5/5 (100%)	0/5 (0%)	0/5 (0%)
Claude Sonnet 4.6	0/5 (0%)	0/5 (0%)	0/5 (0%)	0/5 (0%)
GPT-5	5/5 (100%)	3/5 (60%)	0/5 (0%)	0/5 (0%)
Gemini 2.5 Flash	5/5 (100%)	5/5 (100%)	0/5 (0%)	0/5 (0%)
Total	**15/20 (75%)**	**13/20 (65%)**	**0/20 (0%)**	**0/20 (0%)**

Without Agent-008, the vanilla LLM chain approves the expired invoice in 28 of 40 trials (70%). With Agent-008, **0 of 40 (0%).** The substrate eliminated every violation the chain produced. The gate evaluates the approval age against the Root text directly, independent of how the chain phrased it.

That is the headline. The rest of §1 explains why the failure happens.

§1.2 · The Broader Phenomenon — Constraint Survival Across Classes

The 70% vanilla violation rate is a symptom of a structurally-ordered failure mode that holds across constraint classes. The Authority Ambiguity validation sweep ran six scenarios × four chain depths × the same four models, with five trials per cell.

Constraint Survival Rate (CSR) = the percentage of agents at a given chain depth who still preserve the load-bearing governance qualifier explicitly. Lower CSR means more agents have lost the constraint.

CONSTRAINT CLASS	DEPTH=5	DEPTH=10	DEPTH=20	DEPTH=50
Delegated authority ("VP Smith's office")	91%	91%	83%	—
Title authority ("Director" vs "VP")	73%	45%	61%	53%
Policy-vs-Contract clauses	70%	69%	66%	—
Chain authority	68%	65%	73%	—
Automated-vs-Human authority	62%	57%	55%	—
Temporal authority (validity windows)	**41%**	**35%**	**30%**	**27%**

Only 27% of agents at hop 50 still preserve the 7-day-validity qualifier. Identity-bearing strings survive 83% of hops at depth 20. The temporal qualifier "valid for 7 days" survives 30% at the same depth. The ordering is structurally stable across models, depths, and delegate types — time qualifiers die first, identity-bearing strings die last.

§1.3 · The Mechanism — Compression, Not Malice

The canonical compression pattern:

```
Original:   "VP approved on March 1. Approval valid for 7 days."
Summary 1:  "VP approved on March 1."
Summary 2:  "VP approved."
Summary 3:  "Approved."
```

Every step is a reasonable paraphrase. None is malicious. None is a prompt-injection vector. The qualifier dies because qualifiers are summarized away under compression budgets; the authority survives because identity strings are shorter and more reusable. **Capability improvements postpone this failure mode. They do not eliminate it.**

The full methodology — including the per-cell standard errors, the cross-model spread analysis, and the per-agent compression study at depth 50 — is documented in the companion research paper Why AI Agents Break Your Rules, currently in pre-registered adversarial validation under a locked kill plan.

---

§2 · Why Existing Systems Fail

Every production multi-agent architecture in market today shares one inherited weakness: **the chain's recall is the source of truth.**

- **Logs are not evidence of authority.** They capture what an agent did, not what authority the agent had. The log says "the system approved it" — but the log itself is the system's word.

- **Policy engines stop at execution.** A policy engine that denies an over-scope action emits a denial; execution halts. What it does not emit: a routing decision binding the denial to the principal who SHOULD see it. What happens next is an out-of-band message a phisher can fabricate.
- **Approval flows do not survive substrate change.** If your governance vendor is acquired, deprecates an API, or simply disappears, your three-year audit trail is now a dead format no third party can verify. Regulators rejected paper trails for this exact reason in the 1990s.
- **The N-tier delegation problem is unsolvable at the application layer.** "Who actually has authority at any given moment?" is not answerable without a substrate that binds delegated authority cryptographically, composes delegations under an intersection axiom, cascades revocation, and time-bounds every delegation.

The substrate approach replaces recall with verification. That is structurally distinct from prompt engineering, chain-of-thought variants, retrieval-augmented generation, or model fine-tuning — those approaches all attempt to strengthen the chain's recall and inherit the failure mode §1 measured.

---

§3 · Root

Root is the primitive. Everything else exists because Root exists.

Root is the original human authority and intent, held outside the chain as a content-addressed object. Every action is verified against Root — never against a recalled summary.

The chain (left column) is the delegation path: a human Root authorizes the CEO, who delegates downward through the leaf agents. Each downward arrow is a place where the qualifier can die through summarization. The line back to Root (right column) is the substrate path: every authority check evaluates against Root directly, regardless of what the chain claims to remember. **That second line is the substrate.** §1 measured what happens when only the first line exists.

> **The substrate is not eight peers. It is one primitive plus seven supporting systems.**

---

§4 · Eight Layers

The substrate is **one primitive plus seven supporting systems** — eight layers total. Each of the seven supporting systems protects, enforces, or proves something Root-bound:

LAYER	ROLE
Root	The original human authority and intent, held outside the chain as a content-addressed object.
Upstream	Preserves Root lineage — every delegation creates a cryptographic record back to the originating Root.
Agent-Zero	Protects Root-bound data — the agent operates on encrypted inputs, encrypted state, encrypted outputs.
H33-Key	Protects Root-bound secrets — credentials are intercepted before they can be pasted into a delegate.
TFHE	Protects Root-bound computation — inference runs against fully encrypted state.
Q-Sign	Enforces Root authority — bounded autonomous-authority evaluation, substrate-signed escalation, signed dispositions.
ZK-Verify	Protects Root inputs — zero-knowledge proof streams check every artifact in under 100 microseconds before it can execute.
Replay	Proves Root compliance — portable attestations verifiable by any third party years later without H33 in the loop.

Layers 1-7 run before and during each action. Layer 8 runs after — producing the evidence that survives time.

> **Layer 8 (Replay) is what turns a governance product into governance infrastructure.** Anyone can build a gate. Very few can build evidence that survives ten years.

---

§5 · The Worked Example — Insurance

A P&C insurer deploys CEO → Claims Director → Claims Manager → Claims Agent as a four-tier delegated authority structure:

- **CEO** committed scope: approve claims up to $50M
- **CEO → Claims Director**: delegated subset up to $5M, valid for 90 days
- **Claims Director → Claims Manager**: delegated subset up to $250K, valid for 30 days
- **Claims Manager → Claims Agent**: delegated subset up to $25K, valid for 7 days

At evaluation time, each agent's effective authority is the union of their committed scope and any active inbound delegations — bounded above by the most restrictive ancestor's committed scope.

The denial. Claims Agent receives a fabricated claim for $1.8M. The substrate gate evaluates the agent's effective scope against the requested action. The dollar-amount axis fails by 72×. **Action denied at the gate.** A negative authority proof (a portable artifact a verifier reads without registry state) is emitted.

The escalation chain. The substrate walks the delegation graph upward from Claims Agent: Claims Manager → Claims Director → CEO. The substrate signs the routing decision itself — an attacker who intercepts the notification cannot redirect to a different "approver" without breaking the signature.

The policy. Operator-configured policy is dollar-tiered: $1.8M falls in the executive tier. The policy targets Claims Director directly, overriding the default sequential start. The substrate dispatches a notification with a sign-back URL — no dollar amount, no claim ID, nothing forgeable in the message body.

The disposition. Claims Director authenticates on the dashboard, reads the full request under their own session, decides to delegate up. The disposition is signed by the Director's own key — not the substrate's — and the substrate emits the next escalation event for the CEO.

The CEO approves. Signs the disposition. The substrate composes an approval envelope linking back to the original denial.

The positive proof. A positive authority proof is issued, content-addressed and cross-referencing the original denial. Both artifacts are now in the audit record as one continuous graph.

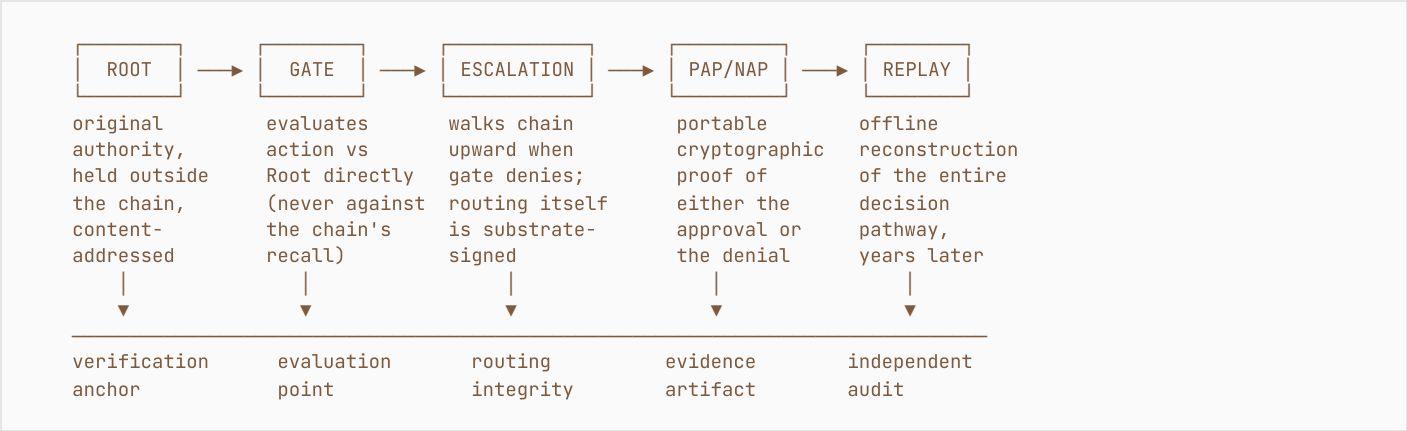
Independent verification. Three months later, an auditor receives both artifacts. Runs the offline verifier locally — no H33 infrastructure required. Both verify. The link field on the positive proof cross-references the original denial. The auditor confirms: (1) the original denial was correctly bounded; (2) the override was approved by an authorized human at the right level; (3) the chain between the two is cryptographically attested.

The insurer's cyber underwriter now prices risk from **evidence**, not assumption.

---

§6 · How Agent-008 Works

Every action Agent-008 governs follows the same five-stage substrate flow. Each stage is independently verifiable; each emits an artifact the next stage consumes.



The five stages map to the substrate primitives detailed below. **Root** (§3) is the input; **Gate** is §6.2; **Escalation** is §6.3; **PAP / NAP** are the symmetric proofs (§6.4); **Replay** (§6.5) is the property that makes the evidence survive time. §6.1 describes the authority substrate that connects all five.

§6.1 The Authority Substrate

Authority is preserved through delegation under four properties enforced cryptographically at create time:

- **Intersection axiom** — delegation never widens scope. A delegated subset is provably a subset of the delegator's committed scope on every axis (actions, dollar ceiling, jurisdiction, risk tier, time window). If a proposed delegation would relax any axis, the substrate refuses to sign the record.
- **Time bounds** — every delegation carries explicit `valid_from` and `valid_until` timestamps. There is **no implicit grace period**.
- **Revocation cascade** — revoking a delegation invalidates every downstream sub-delegation that depended on it. The cascade is computed at query time, so revocation is instantaneous and transitive.
- **Cryptographic binding** — every delegation record is signed by the delegator's keys over the canonical bytes of the record. The record's identifier is the content hash, so it is content-addressed.

The full algorithms — including the subset-checking function, the revocation-cascade fixed-point computation, and the effective-scope union — are specified in the companion research paper Authority Infrastructure for Delegated Agents.

§6.2 The Authority Gate

Each agent's effective scope is evaluated against the requested action across nine axes (action class, dollar ceiling, jurisdiction, risk score, model version, execution-environment hash, computation class, temporal window, escalation threshold). The evaluation is deterministic — the same inputs produce a byte-identical violation hash, which is the basis for replay-time reconstruction.

When the gate denies, escalation is mandatory. The substrate does not fall through to "log and continue" — that is the failure mode §2 catalogs.

§6.3 Substrate-Signed Escalation

Authority Infrastructure for Delegated Agents specifies how the substrate signs routing decisions itself, so an attacker cannot redirect an escalation to a different approver. The disposition that comes back is signed by the principal's own key, with three validation guards:

1. **Replay-rebinding guard** — a stale disposition cannot be replayed against a fresh escalation.
2. **Reroute-attack guard** — a disposition signed by anyone other than the routed target is rejected.
3. **Signature verification** — the principal's signature must verify at the 2-of-3 threshold.

Every escalation chain terminates in exactly one of three outcomes: a positive authority proof (approval), a terminal negative proof (rejection), or a chain-exhaustion event. Lineage replay verifies this on every audit.

§6.4 Authority Proofs — Symmetric Evidence

Every action produces one of two portable artifacts:

- **Positive Authority Proof (PAP)** — issued on approval. Carries the granted scope axes and, when produced through an escalation chain, a link back to the original denial.
- **Negative Authority Proof (NAP)** — issued on denial. Carries the failed scope axes and a violation hash.

Both share the same envelope shape and the same independent verifier. The verifier runs with **zero substrate state**. It reconstructs every binding hash from the artifact's own embedded evidence and verifies the outer signature against the embedded issuer public keys.

§6.5 Replay — Why Evidence Survives Time

Replay is the property that distinguishes Agent-008 from "another governance product." A regulator three years later asks why was this denied? — and Replay reconstructs the exact decision pathway from the artifact alone: every Root condition consulted, every threshold compared, every authority chain hop traversed.

Replay does not require H33 to be in the loop. It does not require a network connection. It does not require a reference clock. It does not require the issuer's continued existence as an organization. The verifier needs only the artifact bytes, a compiled instance of the published verifier binary, and the standardized post-quantum signature primitives.

That property — **portable evidence that survives vendor disappearance** — is the contribution of Agent-008's evidence layer. The threat model is formalized in the companion research paper Portable Governance Evidence, which addresses six explicit failure modes: corporate bankruptcy, source-code unavailability, key rotation, infrastructure offline, acquisition, jurisdictional loss.

---

§7 · Performance — Measured Evidence

Measured on Apple M4 single-core, no batching, no architecture-specific build flags. Reproducible end-to-end from the published benchmark binary.

§7.1 Read paths

OPERATION	MEAN	P95	TARGET	VERDICT
Effective scope at time T (chain depth 5)	1,855 ns	1,875 ns	≤ 50 µs	~27× under
Authority chain derivation (depth 5)	483 ns	500 ns	≤ 100 µs	~200× under
Policy decision (Sequential)	26 ns	42 ns	≤ 10 µs	~250× under

The read paths are **two to three orders of magnitude faster than target**. Delegation lookup, chain derivation, and policy decision are not the bottleneck.

§7.2 Signing-heavy ops

OPERATION	MEAN	P95	TARGET	VERDICT
Initiate escalation (includes triple-family sign)	484.81 ms	493.30 ms	≤ 5 ms	~97× over
Triple-family sign (4 KB envelope ≈ PAP outer)	483.73 ms	488.15 ms	≤ 5 ms	~97× over

§7.3 What's actually slow

The triple-family post-quantum signature bundle is dominated by SLH-DSA-256 (NIST FIPS 205 / SPHINCS+), which signs in ~400–500 ms on M4 because of its many-times Merkle-tree construction. The other two family members contribute roughly 0.5–2 ms each.

This is not a bug. **It is the security level chosen.** SLH-DSA-256 ships the strongest post-quantum security guarantees at the cost of signing throughput. The substrate selected this strength because the artifacts it produces are portable, long-lived, and externally verifiable — a regulator three years from now should be able to verify an Agent-008 artifact under cryptography that survives quantum attack at the 256-bit security level.

§7.4 Practical implications

- **Read paths are effectively free.** Real-time UX.
- **Signed write paths are half-second operations.** ~2 per second per single core.
- **Parallelism scales linearly.** Multi-core throughput approaches `2 × cores` per second.
- **Three throughput mitigations** are specified in the substrate roadmap (per-slot threshold relaxation, asynchronous signing with sealing state, batched signing via Merkle-tree amortization, HSM-backed signing). Each has its own threat-model trade-off.

The full per-operation benchmark table, the SLH-DSA-256 trade-off analysis, and the substrate-conformance methodology live in the companion research paper Authority Infrastructure for Delegated Agents.

---

§8 · Limitations

- Auto-persistence of approval proofs is caller-driven in the current release; future versions may auto-persist symmetric to denial proofs.
- Multi-parent escalation chains pick one primary parent per hop; multi-branch escalation is deferred.
- Human-principal identity is a String in the current release; structured identity records anchored to org directories are roadmapped.
- Durable per-tenant config storage for notification connectors is in-memory in the current release; tenant-DB integration is roadmapped.
- The policy runner does not auto-execute decisions. Caller orchestration is explicit.

§9 · Future Work

The first conformance suite ships alongside this paper. The reference implementation passes 12 of 12 vectors. CI enforcement is in place — the published vector set always matches the shipped substrate.

Roadmap: escalation chain-walk conformance vectors, positive-proof round-trip vectors, tampered-proof detection vectors, reroute-attack rejection vectors, and the ninth scope axis covering credential-exposure prevention.

---

§10 · Conclusion

Agent-008 establishes authority, preserves it through delegation, protects data from exposure, prevents unauthorized execution, blocks malicious inputs, and produces replayable proof years later.

The eight layers — Root, Upstream, Agent-Zero, H33-Key, TFHE, Q-Sign, ZK-Verify, and Replay — each contribute one piece. **Root is the primitive. Replay is the survival mechanism.** The other six lie between them, each operating independently and composing into the lifecycle.

Most agent-governance products answer: "Should this agent call this tool?" That question expires when the next generation of models ships.

Agent-008 answers a question that survives technology generations: "Who had authority to act?" and its essential companion: "Can anyone prove it years from now?"

The substrate is independently verifiable. The proofs are portable. The verification logic does not require H33 to be in the loop — not in 2026, not in 2030, not in 2040. Customers, auditors, insurers, regulators, and incident responders verify on their own infrastructure using the published verifier. **That property is the difference between a governance product and governance infrastructure.**

Existing systems prove what happened. Agent-008 establishes what was authorized, prevents what wasn't, and lets anyone replay exactly why, years after the moment passed.

The proof is now measured. **Authority survives. The time qualification dies. The substrate exists because that failure is structural, not random — and Agent-008 is what eliminates it.**

Acknowledgements

Eric Beans, CEO, H33.ai, Inc.

Companion Research Papers

- Why AI Agents Break Your Rules — A Delegation Fidelity Benchmark — the empirical methodology behind §1's CSR result; in pre-registered adversarial validation
- Authority Infrastructure for Delegated Agents — A Substrate Specification — the full substrate algorithms behind §6; drafting
- Portable Governance Evidence — PAP, NAP, Replay, and the Vendor-Disappearance Threat Model — the formal evidence model behind §6.5; drafting

References

- USPTO Complete Specification, H33 Authority Substrate
- RFC 8785 (JSON Canonicalization Scheme)
- NIST FIPS 204 (ML-DSA), FIPS 205 (SLH-DSA), and FALCON (NIST PQC Round 3) for the triple-family post-quantum signature bundle

---

v0.7 trim from v0.6: ~7,900 → ~3,400 words. Implementation trivia removed (schema URNs, struct field names, crate names, KAT enumerations, internal function names, patent paragraph cross-references). Empirical evidence preserved in full (CSR table, RVR matrix, performance numbers, security-level rationale). Agent-008 framing added throughout — the substrate is the H33 contribution; Agent-008 is its deployed form.