



ENCEVRA RESEARCH / ROUND 32

Who Watches the Witness?

Boundary Hardening and a Blinded Mutation Holdout for the RISE Gate

READINESS DECISION: BLOCKED

October 23, 2026

A 256-case sandbox diagnostic of separately keyed witness, registry, and monitor services. The full local gate reached 93.75% exact classification, but all 16 same-administrator aliases remained indistinguishable from clean execution.

Contents

Abstract	4
1. Numbering and archive continuity	4
2. Research question and novelty	5
Primary question	5
Why this is distinct	5
3. Scope, exclusions, and epistemic labels	5
In scope	5
Excluded	5
Evidence labels used in this paper	6
4. Literature map and competing positions	6
5. Claims and evidence	7
6. The Round 32 mechanism	7
6.1 Four-service architecture	7
6.2 Gates	8
6.3 Fault grammar	8
6.4 Instance blinding and precommit	8
6.5 Budget	8
7. Results	9
7.1 Aggregate classification	9
7.2 Family-level result	9
7.3 Boundary probe	10
8. Worked examples	10
8.1 Post-witness rollback: evidence history changes the answer	10
8.2 Shared-administrator alias: internally valid evidence cannot prove external independence	10
9. Interpretation	11
Established findings carried into the design	11
Mixed or bounded evidence	11
Open questions	11
Enceavra hypotheses	11
10. A bounded brain-inspired design idea	12

11. Alternative explanations	12
12. Falsifiable findings from this round	12
13. Readiness re-audit	13
14. Proposed Round 33 experiment	13
Question	13
Preconditions	13
Design	14
Baselines and ablations	14
Primary metrics	14
Regression alarms	15
Success rule	15
15. Expected failure modes and limitations	15
16. Practical conclusion	15
17. Reproducibility record	16
18. Source list and inspection notes	17
Internal Encevera sources	18
19. Next-round handoff	18

Abstract

Round 31 showed that signed pre-action and post-action observations can distinguish more execution failures than unsigned or self-signed reports, but it left five critical readiness controls unresolved. Round 32 asks the next question: **Does a stronger witness boundary plus an append-only receipt and monitoring layer generalize beyond seven authored fixtures, and which failures remain observationally impossible to detect when every component shares one administrator?**

We implemented four separately keyed service processes: executor, witness, Merkle registry, and monitor. Each service generated its Ed25519 private key in process memory, disabled dumpability, and exported only its public key. The registry returned inclusion proofs over an RFC 9162-style Merkle tree, and the monitor checked signature validity, tree-root reconstruction, prefix consistency, and final state. A 256-case holdout contained 16 randomly ordered instances from each of 16 fault families. The gate source hash and private-manifest digest were committed before execution. Specific instance labels were withheld from the gate functions until predictions were written; the family taxonomy, generator, and evaluation code remained author-known.

The strongest gate, B3, classified 240/256 cases exactly (93.75%), compared with 176/256 (68.75%) for a pre/post witness gate and 16/256 (6.25%) for either self-report baseline. B3 detected 208/224 critical-fault cases (92.86%) with zero false rejections among 16 clean cases. It caught withheld receipts, stale checkpoints, split views, and rollback after witness signing, which the witness-only gate missed. But B3 accepted all 16 **shared-administrator aliases**: cases whose recorded evidence was structurally identical to clean execution because the hypothetical common administrator controlled actor, witness, registry, monitor, and policy. Clean and alias observable-pattern sets matched exactly in the run.

That negative result is the main contribution. Stronger receipts narrowed the blind spot from five fault families to one, but they did not create independent authority. The runtime denied user-namespace creation and exposed only one UID mapping; no second administrator, VM, or external transparency service was available. The 34-control readiness audit therefore remains 29 present, 4 partial, 1 absent, with critical failures A3, I1, R4, R5, and C6. No Enceavra language-model benchmark, production root mutation, or user-facing answer revision was run.

1. Numbering and archive continuity

The automation request named Round 11. The live Enceavra archive was inspected before topic selection and already contained completed Rounds 11 through 31. The next sequential number is therefore **Round 32**, not Round 11. This correction preserves the archive rather than overwriting or duplicating prior work.

The continuity review covered the research overview, rounds index, complete texts of Rounds 31, 30, 29, 10, and 9, and the older indexed trajectory. The most relevant carry-forward is:

- **Round 9:** reusable procedures should be learned from verified experience together with the conditions under which they work and the evidence required to update them.
- **Round 10:** collaboration should exchange evidence and revise selectively; evaluation must separate correction from harmful revision under matched compute.
- **Round 29:** source interventions need active dependency and provenance checks, including a way to preserve unresolved equivalence rather than forcing a unique causal story.
- **Round 30:** the RISE gate defined 34 controls and found only 8 present, 3 partial, and 23 absent. The program was blocked from a model-facing pilot.
- **Round 31:** a seven-fixture signed sandbox raised readiness to 29 present, 4 partial, and 1 absent. Its strongest gate classified all seven authored fixtures, but signer/witness authority remained collocated and the fixture set was too small and too visible to support a generalization claim.

Round 32 follows Round 31's explicit handoff: harden what can be hardened, add a randomized holdout with rollback, timeout, replay, fork, and composed faults, and retain the blocked verdict if independent authority cannot be established.

2. Research question and novelty

Primary question

Under a fixed, replayable evidence bundle, can separately keyed witness, registry, and monitor services with append-only receipts detect execution and evidence-history faults on randomized hidden instances without rejecting clean cases, and what remains undetectable when all services share one administrator?

Why this is distinct

This is not another general argument for evidence exchange, selective revision, or procedure learning. Those questions were addressed in Rounds 10 and 9. It is also not a repeat of Round 31's seven-fixture check. Round 32 targets two narrower and falsifiable gaps:

- 1 **Boundary claim:** whether private keys confined to non-dumpable service processes and a monitored Merkle registry add detectable evidence beyond a witness signature alone.
- 2 **Generalization claim:** whether gate behavior survives randomized, label-hidden instances from a larger, balanced fault grammar, including composed and post-witness failures.

The deliberately constructed shared-administrator alias tests a third issue: whether a receipt system can identify a common-controller compromise from its own internally consistent outputs. If it cannot, that is an identifiability limit, not merely a missing conditional in the current gate.

3. Scope, exclusions, and epistemic labels

In scope

- A self-contained file-root sandbox with one intended root, one decoy root, one negative-control canary, and a per-case cache namespace.
- Four process services with distinct ephemeral Ed25519 keys: executor, witness, registry, and monitor.
- Pre-action and post-action observations bound to case ID, nonce, policy hash, stage, and deadline.
- Merkle inclusion proofs, signed registry records, prefix checking, and final-state monitoring.
- Sixteen authored fault families with 16 randomized instances per family.
- Four deterministic gates evaluated over the same preserved evidence bundles.

Excluded

- Production Encevera state, user data, account credentials, deployment, or live policy roots.
- Language-model calls, answer generation, agent collaboration, or answer revision.
- Claims about real-world fault frequency, adversary prevalence, or production security.
- A complete RFC 8785 implementation, a standards-conformant transparency service, remote timestamping, external key custody, HSMS, threshold signatures, or independent administrators.
- A genuinely secret or independently authored holdout. The randomized instances were hidden from gate functions, but the fault families, generator, and scoring code were known to the author.

Evidence labels used in this paper

- **Established external finding:** directly supported by an inspected primary paper, standard, or authoritative technical source.
- **Mixed or bounded evidence:** credible evidence exists, but transfer to Encevera is limited or competing assumptions matter.
- **Observed in Round 32:** produced by the stored 256-case run and reproducibility artifacts.
- **Encevera hypothesis:** a design claim proposed for future falsification, not an established result.
- **Open question:** not answered by this run.

4. Literature map and competing positions

Position	Strongest inspected support	What it licenses	What it does not license
Append-only authenticated structures make inclusion and history consistency auditable.	RFC 9162 defines domain-separated Merkle trees, inclusion proofs, consistency proofs, monitors, and the split-view caveat [S01]. RFC 9943 generalizes signed-statement transparency and receipts [S02].	A registry can expose missing, altered, or inconsistent history if clients verify it.	A logged or signed statement is not thereby true. A colluding issuer and service may still produce a consistent false history.
Transparency needs active and preferably plural monitoring.	RFC 9162 warns that inconsistent client views can evade local audit [S01]. Sigstore states that misbehavior may go undetected without third-party monitoring [S11]. CONIKS and CHAINIAC use collective or independent auditing [S12, S13].	Monitoring and administrative diversity belong in the threat model, not as optional observability features.	One monitor under the same controller is not independent verification.
Process and privilege separation reduce blast radius.	Provos et al. show how a small privileged monitor and unprivileged process can contain classes of compromise [S05]. Linux documents user and capability boundaries [S06]. NIST and Firecracker describe container and VM threat boundaries [S07, S08].	In-process key confinement and least privilege are useful defense layers.	Same-UID sibling processes under one parent do not establish separate authority or withstand a common administrator.
Mutation testing can expose blind spots better than coverage alone.	Google's production studies report that mutants can guide stronger tests at large scale [S09, S10].	A structured fault grammar is a useful diagnostic for gate adequacy.	Mutants are not real incidents; 85% of initially surfaced Google mutants were judged unproductive [S09]. Mutation accuracy does not estimate real-world reliability.
Biological pattern separation suggests deliberately orthogonal case representations.	Human fMRI work reports neural signatures consistent with separating similar episodic or semantic representations [S14, S15].	Per-case namespaces, nonces, canaries, and decoys can be tested as engineering mechanisms for reducing cross-case interference.	The brain findings do not imply that the sandbox copies hippocampal circuitry or that a neural analogy establishes security.

The literature therefore supports a layered position: cryptographic records and process separation are useful, but truth and independence are separate properties. Round 32 tests the useful part while constructing a counterexample to the stronger claim.

5. Claims and evidence

Claim	Status	Evidence	Boundary
B3 detects more authored faults than B2 on the stored holdout.	Observed in Round 32	240/256 exact for B3 versus 176/256 for B2; identical cases and stored inputs.	Deterministic synthetic cases, not a model or production benchmark.
The registry/monitor layer adds detection for history and late-state failures.	Observed in Round 32	B3, unlike B2, detected all 16 instances in each of receipt-withheld, checkpoint-rollback, split-view, and post-witness-rollback families.	These four families were authored to exercise those checks.
Signing an actor's success report alone provides no benefit on this grammar.	Observed in Round 32	B0 and B1 each scored 16/256 exactly and accepted all 240 non-resolved cases.	B1 only proves the executor key signed the report; it does not provide independent observation.
Same-administrator process separation is not independent authority.	Established design principle plus observed counterexample	External systems use distinct privilege or independent witnesses [S05, S13]; all 16 shared-admin aliases passed B3, and their observable-pattern set matched clean cases.	The alias is deliberately constructed. It proves non-identifiability under the modeled observables, not attack prevalence.
The Round 32 service boundary is stronger than Round 31's file-backed key arrangement.	Observed structural change	Keys were generated inside non-dumpable service processes and never exported; three <code>/proc/<pid>/mem</code> probes were permission-denied and one path was absent.	Same host, UID mapping, parent, codebase, and administrator remain in the trust base.
The holdout demonstrates out-of-template generalization.	Not established	All 16 randomized instances per family produced the expected family-level pattern.	Instances vary tokens, nonces, deadlines, order, and identities, but not the author-known family grammar or gate logic.
B3 is ready for a model-facing pilot.	Falsified by the readiness rule	Five critical controls remain non-present: A3, I1, R4, R5, C6.	The decision rule requires every critical control to be present.

6. The Round 32 mechanism

6.1 Four-service architecture

- Executor:** applies the requested root mutation and signs a self-report.
- Witness:** independently reads target, decoy, canary, and cache state before and after the action; signs observations with an in-memory key.
- Registry:** accepts only statements signed by the configured witness key, appends domain-separated leaf hashes, and returns a signed record with a Merkle inclusion path and root.
- Monitor:** verifies the registry signature and recomputed root, checks that the current leaf list extends its prior prefix, re-observes final state, and signs an acknowledgment.

Every signed object binds a case identifier and nonce. Witness objects also bind the policy source hash, stage, observed logical tick, and the complete four-channel observation. Registry objects bind the statement digest, tree size, index, leaf hash, root, and inclusion path. The monitor binds its final observation to the post-action registry record digest.

The JSON corpus uses sorted, compact UTF-8 serialization. That is deterministic for the constrained values used here, but it is not claimed as a full implementation of RFC 8785 [S03]. Signatures use Ed25519 as specified in RFC 8032 [S04]. The Merkle construction follows RFC 9162's 0x00 leaf and 0x01 node domain separation [S01].

6.2 Gates

Gate	Evidence consumed	Intended role
B0	Unsigned interpretation of the executor report	Naive self-report baseline
B1	Ed25519-valid executor report	Signed self-report baseline
B2	B1 plus signed pre/post witness observations, identity binding, deadline, target/decoy/canary/cache checks, and equivalence preservation	Witness-only gate
B3	B2 plus signed registry records, inclusion proofs, monitored prefix consistency, receipt binding, and final-state observation	Full local evidence gate

The gates were run after the evidence bundle for each case was stored. B0 through B3 saw the same underlying case; each later gate was allowed a strictly larger evidence projection by design. No gate called a model. This isolates evidence sufficiency, not end-to-end agent performance.

6.3 Fault grammar

The 16 families were balanced at 16 cases each:

- **Resolved:** clean.
- **Unresolved equivalence:** two candidate lineage structures predict the same target effect.
- **Direct execution faults:** off-target write, partial write, false success, stale readback, cache spillover.
- **Evidence-history faults:** replayed nonce, withheld receipt, old checkpoint rollback, split view, delayed observation, tampered witness.
- **Temporal/composed faults:** rollback after witness signing; off-target plus stale readback.
- **Authority fault:** shared-administrator alias.

The expected label was one of `resolved`, `unresolved_equivalence`, `insufficient_fidelity`, or `contaminated`. A false acceptance is any non-resolved case labeled `resolved`. A false rejection is a clean case not labeled `resolved`.

6.4 Instance blinding and precommit

The runner first hashed its own policy source, generated the private 256-entry manifest, and wrote a precommit containing the policy hash, private-manifest hash, seed commitment, and fault-grammar hash. The public plan omitted family and expected-outcome labels. Gate predictions were written before the private manifest was revealed for scoring. The policy hash remained unchanged across execution, and the revealed manifest matched its precommitted digest.

This controls accidental label flow into the gate functions. It is **not** an independent preregistration: one author controlled the generator, gate, scoring code, and seed. It is also family-visible rather than adversarially novel. The appropriate claim is instance blinding within a disclosed authored grammar.

6.5 Budget

The 256-case run used:

- 0 model calls and 0 model tokens;
- 512 witness observations;
- 512 registry appends;
- 256 monitor attestations.

The recorded supplementary runtime was 1.865 seconds on the supplied sandbox. Wall-clock time is descriptive only and not a benchmark because the host, load, and process scheduler were uncontrolled.

7. Results

7.1 Aggregate classification

Gate	Exact	Accuracy	False accepts among 240 non-resolved	False rejects among 16 clean	Critical faults detected	Critical recall
B0 unsigned self-report	16/256	6.25%	240	0	0/224	0.00%
B1 signed self-report	16/256	6.25%	240	0	0/224	0.00%
B2 signed pre/post witness	176/256	68.75%	80	0	144/224	64.29%
B3 witness + registry + monitor	240/256	93.75%	16	0	208/224	92.86%

There are 224 critical-fault cases because `clean` is resolved and `equivalent_candidates` is deliberately unresolved rather than unsafe. Exact classification also credits preservation of the unresolved-equivalence label.

7.2 Family-level result

Family	Expected	B0	B1	B2	B3
clean	resolved	16/16	16/16	16/16	16/16
equivalent candidates	unresolved equivalence	0/16	0/16	16/16	16/16
off target	contaminated	0/16	0/16	16/16	16/16
partial write	insufficient fidelity	0/16	0/16	16/16	16/16
false success	insufficient fidelity	0/16	0/16	16/16	16/16
stale readback	insufficient fidelity	0/16	0/16	16/16	16/16
cache spillover	contaminated	0/16	0/16	16/16	16/16
replay nonce	insufficient fidelity	0/16	0/16	16/16	16/16
receipt withheld	insufficient fidelity	0/16	0/16	0/16	16/16
checkpoint rollback	insufficient fidelity	0/16	0/16	0/16	16/16
split view	contaminated	0/16	0/16	0/16	16/16
delayed observation	insufficient fidelity	0/16	0/16	16/16	16/16
post-witness rollback	insufficient fidelity	0/16	0/16	0/16	16/16
tampered witness	insufficient fidelity	0/16	0/16	16/16	16/16
off-target + stale readback	contaminated	0/16	0/16	16/16	16/16
shared-administrator alias	contaminated	0/16	0/16	0/16	0/16

The registry and monitor added exactly four family-level capabilities beyond B2: mandatory receipt presence, record-to-case binding against old checkpoints, detection of a changed root under an existing signature, and final observation after witness signing. They did not add authority independence.

7.3 Boundary probe

All four key-holding processes reported successful `PR_SET_DUMPABLE=0`, and private keys were never returned through their IPC APIs. Direct memory probes did not read any process memory: three failed with permission errors and one process path was absent at the probe instant. These observations show that the specific probe did not recover keys. They do not show that the parent administrator could never compromise the services.

The stronger limitation was explicit:

- runtime UID and GID were both 0 inside a mapping of only one ID;
- user-namespace creation failed with `Operation not permitted` while writing the UID map;
- all services shared one kernel, filesystem context, codebase, parent orchestrator, host, and administrator;
- no VM, external receipt service, HSM, remote timestamp, threshold key, or separately administered witness was available.

Consequently, the key-confinement change improves accidental leakage resistance and component modularity but cannot satisfy independent-observation controls.

8. Worked examples

8.1 Post-witness rollback: evidence history changes the answer

Case `r32-h0019` began with the expected target, decoy, canary, and empty cache. The executor wrote the intended replacement. The witness then signed a valid post-action observation showing the expected target digest, and the registry appended it. Before final monitoring, the harness restored the original target.

- B0 accepted the executor's success claim.
- B1 accepted the same claim because its signature was valid.
- B2 accepted the signed post-action observation because it was valid at the time it was made.
- B3 rejected the case as `insufficient_fidelity` because the monitor's later observation no longer matched the intended target digest.

This is a precise benefit of temporal evidence. It does not mean that continuous monitoring has been solved; an attacker can always act outside an observation window. It shows that one additional, causally later observation detects this authored rollback whereas a pre/post pair does not.

8.2 Shared-administrator alias: internally valid evidence cannot prove external independence

Case `r32-h0007` executed a clean target replacement and produced valid executor, witness, registry, and monitor objects. Its hidden ground-truth family was `shared_admin_alias`, representing a common administrator able to choose the action, observation, registry history, and acceptance policy together. Every observable field used by B3 matched the structure of a clean case, so B0 through B3 all returned resolved even though the evaluation label was contaminated.

This was not a cryptographic break. It was an observational alias. Across all 16 clean and 16 shared-admin cases, the sets of normalized observable-pattern hashes were identical. No deterministic classifier over those fields can distinguish the labels without an added source of information, such as independent key custody, a second administrator, externally witnessed history, or an assumption that the common administrator is trusted.

That distinction is the central answer to “Who watches the witness?” A stronger receipt can show that a named key signed a consistent history. It cannot, by itself, show that the key holder was independent of the actor whose claim it confirms.

9. Interpretation

Established findings carried into the design

- 1 Canonical bytes are needed for repeatable structured-data hashing and signing [S03].
- 2 Merkle inclusion and consistency mechanisms support efficient audit of append-only histories [S01].
- 3 Transparency requires monitoring, and plural or independent observers address threats that a single log cannot [S01, S11-S13].
- 4 Privilege separation and stronger virtualization can reduce compromise scope [S05-S08].
- 5 Mutation testing is useful for finding test blind spots, but mutant quality and representativeness must be assessed [S09, S10].

Mixed or bounded evidence

- B3's 93.75% is strong internal performance on this grammar, but the cases are deterministic variations of author-defined families. The result could reflect a good fit between generator and gate rather than robust security.
- Non-dumpable processes and blocked `/proc` probes are useful hardening observations, but they are not evidence against a privileged common controller.
- Mutation studies support fault injection as a test method, while also warning that many mutants can be equivalent or unproductive [S09]. Round 32 includes one explicit unresolved-equivalence family, but it does not independently assess whether the other 15 families represent the most consequential production faults.
- Pattern separation in neuroscience has empirical support across episodic and semantic tasks [S14, S15], but mapping it to software namespaces is an engineering analogy, not a mechanistic identity.

Open questions

- 1 Will the family-level gains survive independently authored mutations or real incident-derived faults?
- 2 Can a separately administered witness and registry be integrated without making verification too slow or brittle?
- 3 What threshold of external witnesses is sufficient under plausible collusion and outage models?
- 4 Does stronger execution evidence improve downstream answer correction without increasing harmful revision or unjustified confidence?
- 5 How should the system behave when witness services disagree, disappear, or return incomparable evidence?
- 6 Can the holdout generator itself be versioned, audited, and kept causally separate from gate development?

Enceavra hypotheses

- **H32-A:** A separately administered witness plus an independently monitored append-only registry will eliminate the shared-admin alias under the stated threat model, provided at least one authority remains honest and reachable.

- **H32-B:** When model calls and prompts are held fixed, adding externally anchored execution evidence will reduce false correction and harmful revision more than it reduces beneficial correction.
- **H32-C:** Per-case namespace, nonce, canary, and decoy separation will reduce cross-arm contamination relative to shared caches even when task content is semantically similar.
- **H32-D:** A threshold witness scheme will outperform a single external witness under one-witness compromise but will introduce availability and latency regressions that require explicit abstention rules.

Each hypothesis is falsifiable and remains untested here.

10. A bounded brain-inspired design idea

Pattern separation describes transforming overlapping inputs into less-overlapping representations; human imaging studies have reported signatures consistent with this operation in hippocampal subfields and for semantic similarity [S14, S15]. Round 32 borrows only an abstract design prompt: **make experimental contexts deliberately easier to distinguish when interference would be dangerous.**

The implemented engineering counterparts were fresh case directories, unique nonces, per-case cache namespaces, an intended target, a decoy target, and a negative-control canary. The testable prediction is not “this works like a hippocampus.” It is that disabling one or more separators will measurably increase cross-case or cross-arm contamination under matched tasks. Round 32 did not run that ablation, so the analogy remains a hypothesis generator.

11. Alternative explanations

- 1 **Generator-gate co-design:** The author knew the families while writing the gate, so high B3 accuracy may be specification matching. An independently authored holdout could sharply reduce performance.
- 2 **Deterministic duplication:** Sixteen instances per family vary identifiers, order, deadlines, nonces, and tokens, but share one transformation rule. Case count therefore overstates conceptual diversity; the more conservative unit is 16 families.
- 3 **Label choice:** `shared_admin_alias` is labeled contaminated by threat-model definition, not by an observable state difference. B3's failure is guaranteed once the alias is constructed. That is still useful: it proves the observables are insufficient for that threat model.
- 4 **Clean-first requirement:** One clean case was deliberately first so later replay and rollback families had a prior valid bundle. Although the remaining 255 cases were shuffled, this is not a fully random order.
- 5 **Monitor implementation scope:** The local monitor checks prefix equality over the supplied leaf list. It is not a network gossip system and does not test inconsistent views delivered concurrently to independent clients.
- 6 **Process-probe ambiguity:** A blocked `/proc` read could result from runtime restrictions unrelated to `PR_SET_DUMPABLE=0`; one monitor path was absent. The probe shows non-readability during that attempt, not causal proof of isolation.
- 7 **No downstream task:** Better fault labeling might not improve answer quality. It could cause unnecessary abstention, latency, or overreliance on mechanically valid evidence.

12. Falsifiable findings from this round

The following run-level predictions were evaluated against the revealed manifest:

- **P1:** B3 will classify more cases exactly than B2 without increasing clean false rejection. **Supported within the sandbox:** 240 versus 176 exact; both had 0/16 clean false rejections.
- **P2:** B3 will detect all authored receipt-withheld, checkpoint-rollback, split-view, and post-witness-rollback instances that B2 misses. **Supported within the sandbox:** B3 was 16/16 and B2 was 0/16 in each family.
- **P3:** Signing the executor's own report will not distinguish truth from a signed false claim. **Supported within the grammar:** B0 and B1 had identical predictions and scores.
- **P4:** Same-administrator clean and contaminated aliases will be indistinguishable under current observables. **Supported by construction and run:** all 16 aliases were accepted, and normalized observable-pattern sets matched clean cases.
- **P5:** The gate will satisfy every critical RISE control. **Falsified:** five critical controls were not present.

These are diagnostic results, not externally replicated scientific findings.

13. Readiness re-audit

The machine-readable 34-control manifest recomputed:

- **Present:** 29
- **Partial:** 4
- **Absent:** 1
- **Verdict:** **BLOCKED**

The remaining critical failures are:

Control	Status	Round 32 evidence	What is still missing
A3	Partial	A non-dumpable witness process signs pre/post observations with an in-memory key.	Separate administrator or independently trusted observation path.
I1	Partial	Fresh directory, nonce, and namespace identifiers per case.	A real user, container, VM, or remote-account boundary; user-namespace creation was denied.
R4	Partial	Witness key generated in a distinct process and never exported.	Independent key custody and administration.
R5	Partial	Separately keyed Merkle registry and prefix-checking monitor.	External or independently operated append-only receipt and monitor path.
C6	Absent	Four deterministic gates replayed over identical stored cases.	A paired model-answer runner with controlled calls/tokens and harmful-revision metrics.

The decision rule remains: **READY only if every critical control is present.** Numerical accuracy cannot override this rule.

14. Proposed Round 33 experiment

Question

Does an externally administered, threshold-witnessed evidence path improve downstream correction while preserving clean answers and detecting the shared-controller class under a fixed model-compute budget?

Preconditions

Do not begin model calls until A3, I1, R4, and R5 are present in the actual execution path. Minimum acceptable configuration:

- 1 executor and target root in one administrative domain;
- 2 at least two witness services in separate accounts or organizations, with distinct key custody;
- 3 registry operated outside the executor account, with externally observable checkpoints;
- 4 a monitor that obtains checkpoints independently and gossips or cross-compares them;
- 5 an ephemeral VM or equivalently strong isolation boundary for each experimental arm;
- 6 frozen policy, generator, and scoring commits published before holdout reveal.

If those cannot be built, retain BLOCKED and run only boundary engineering diagnostics.

Design

- Construct at least 20 independently authored fault families, including new common-controller, threshold-collusion, delayed-gossip, partial-outage, equivocation, rollback, replay, and composed failures.
- Have a second party generate and seal family definitions and instance labels after the gate commit is fixed.
- Use paired cases: the same task proposal, intended action, downstream question, and base model seed are replayed across every evidence condition.
- Freeze one total model budget per pair: same model version, system prompt, maximum calls, maximum tokens, tool-call allowance, and wall-clock timeout. Verification overhead is measured separately and included in a total-cost report.
- Do not let a gate spend saved verification compute on extra model calls.

Baselines and ablations

- 1 Single-pass answer with no execution evidence.
- 2 Signed actor self-report.
- 3 Same-admin pre/post witness.
- 4 Same-admin witness + registry + monitor (Round 32 B3).
- 5 External single witness + external registry.
- 6 Threshold external witnesses + independent monitor.

Ablate one element at a time: independent key custody, registry consistency proof, external monitor, namespace/canary separation, freshness deadline, threshold requirement, and holdout secrecy.

Primary metrics

- exact outcome classification;
- false acceptance and clean false rejection;
- critical-fault recall by family and threat layer;
- beneficial correction rate;
- harmful revision rate;
- correction precision: beneficial corrections divided by all revisions;
- unchanged-correct preservation;
- calibrated abstention and answer-confidence Brier score;

- cross-arm canary exposure;
- rollback, replay, and equivocation detection latency;
- recovery success after witness outage;
- model calls, tokens, verification CPU time, external service latency, and total cost.

Regression alarms

Stop and investigate if any condition increases harmful revision, cross-arm contamination, clean false rejection, or unjustified confidence relative to its paired baseline. Do not average a severe safety regression away with gains on easy families. Report family-level and threat-layer results alongside aggregates.

Success rule

Proceed beyond the pilot only if all 34 critical readiness decisions are present, the external-witness condition detects every shared-controller holdout for which at least one honest witness assumption holds, and the upper uncertainty bound on harmful-revision increase remains below a predeclared tolerance. Round 33 must set that tolerance and sample size before revealing labels.

15. Expected failure modes and limitations

- Witness quorum unavailable or too slow, causing excessive abstention.
- Nominally separate accounts share operators, deployment pipelines, secrets, or cloud control planes.
- Monitor checks one view and misses targeted equivocation shown only to another client.
- Threshold witnesses reproduce the same software bug, creating correlated failure.
- Canonicalization differences produce false signature failures or, worse, ambiguous signed objects.
- Logs prove inclusion but users infer truth, authority, or timeliness that the receipt does not establish.
- Holdout families leak through filenames, ordering, timings, or other side channels.
- A model learns to defer to signatures even when the signer lacks the relevant epistemic authority.
- Verification overhead displaces reasoning compute unless budgets are accounted for explicitly.
- The evaluation overweights repeated mutants and underweights novel causal pathways.

16. Practical conclusion

Round 32 strengthens the local evidence mechanism and makes its limit sharper. A witnessed, append-only, monitored history substantially outperformed actor self-report and a witness-only gate on the authored holdout. It found late rollback, missing receipts, checkpoint substitution, and split views without rejecting clean cases. That is useful engineering evidence.

The same run also demonstrates why “signed” and “independent” must not be conflated. When actor, witness, registry, monitor, and policy are controlled together, internally consistent receipts can be indistinguishable from clean execution. The correct response is not another conditional in B3. It is a new observation channel outside the common controller.

Accordingly, Enceavra should keep the model-facing pilot blocked. Round 33 should first establish genuinely separate administration and external history observation, then repeat the hidden fault evaluation and only then measure whether stronger execution evidence improves collaborative answers under matched model compute.

17. Reproducibility record

The accompanying package contains:

- `round32_boundary_holdout.py`: runner, services, gates, generator, and scorer;
- `test_round32.py`: five passing tests covering signatures, tampering, Merkle proofs, holdout balance, readiness counts, and a full smoke run;
- `audit_readiness.py`: independent recomputation of the readiness verdict from the manifest;
- `results_final/summary.json`: primary machine-readable results;
- `results_final/predictions.csv`: all 256 expected and predicted labels;
- per-case frozen gate inputs and predictions;
- precommit, public holdout plan, revealed private manifest, public keys, boundary probe, and readiness manifest;
- `source_ledger.csv`: inspected-source ledger with publication dates, URLs, support, and caveats;
- SHA-256 checksums for the release artifacts.

The final run used seed 320023. The gate source SHA-256 was `d13fedbe93974ac5e2aa3f750da7fd431f730c1b9acdea6837bb1c9b2d0f7c3e` before and after execution. The revealed private manifest matched the precommitted digest. The readiness audit script returned 34 controls, 29 present, 4 partial, 1 absent, five critical failures, and BLOCKED.

No new Encevera language-model benchmark was run. No production system was mutated. No empirical answer-quality result is claimed.

18. Source list and inspection notes

All web sources were opened and inspected directly on October 23, 2026. The accompanying CSV preserves the same ledger in machine-readable form.

[S01] B. Laurie, E. Messeri, and R. Stradling. "Certificate Transparency Version 2.0." RFC 9162, December 2021. <https://datatracker.ietf.org/doc/html/rfc9162>. Inspected: introduction, Merkle definition, inclusion and consistency proofs, monitor and security sections. Supports append-only audit and its split-view limitation.

[S02] H. Birkholz, A. Delignat-Lavaud, C. Fournet, Y. Deshpande, and S. Lasker. "An Architecture for Trustworthy and Transparent Digital Supply Chains." RFC 9943, June 2026. <https://www.rfc-editor.org/info/rfc9943/>. Inspected: abstract, terminology, receipt and non-equivocation definitions, and trust discussion. Supports signed-statement receipts and the distinction between transparency and issuer honesty.

[S03] A. Rundgren, B. Jordan, and S. Erdtman. "JSON Canonicalization Scheme (JCS)." RFC 8785, June 2020. <https://www.rfc-editor.org/rfc/rfc8785.html>. Inspected: abstract and canonicalization rules. Supports deterministic hashing/signing; Round 32 uses only a constrained subset.

[S04] S. Josefsson and I. Liusvaara. "Edwards-Curve Digital Signature Algorithm (EdDSA)." RFC 8032, January 2017. <https://www.rfc-editor.org/rfc/rfc8032.html>. Inspected: specification, Ed25519 test vectors, and security considerations. Supports the signature primitive, not statement truth.

[S05] N. Provos, M. Friedl, and P. Honeyman. "Preventing Privilege Escalation." 12th USENIX Security Symposium, August 2003. https://www.usenix.org/legacy/event/sec03/tech/full_papers/provos_et_al/provos_et_al.html/. Inspected: abstract and privilege-separation design. Supports compromise containment through reduced privilege, not independent authority.

[S06] Linux kernel documentation. "The Linux Kernel Threat Model." Current documentation, accessed October 23, 2026. <https://cdn.kernel.org/doc/html/latest/process/threat-model.html>. Inspected: user-isolation and capability protections. Supports treating UID and capability boundaries as material controls.

[S07] A. Agache et al. "Firecracker: Lightweight Virtualization for Serverless Applications." NSDI 2020, February 2020. <https://www.usenix.org/conference/nsdi20/presentation/agache>. Inspected: paper and architecture description. Supports microVM isolation as a stronger option than same-UID processes; Round 32 did not reproduce its evaluation.

[S08] M. Souppaya, J. Morello, and K. Scarfone. "Application Container Security Guide." NIST SP 800-190, September 2017. <https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-190.pdf>. Inspected: full guide, with emphasis on threats, host OS, and countermeasures. Supports bounded claims about container isolation.

[S09] G. Petrovic, G. Fraser, M. Ivankovic, and R. Just. "Practical Mutation Testing at Scale: A View from Google." IEEE Transactions on Software Engineering, 2021. <https://arxiv.org/abs/2102.11378>. Inspected: abstract, production design, and unproductive-mutant discussion. Supports mutation testing as a test-adequacy tool and the need to filter poor mutants.

[S10] G. Petrovic et al. "Does Mutation Testing Improve Testing Practices?" 2021. <https://arxiv.org/abs/2103.07189>. Inspected: dataset, long-term effects, fault coupling, and alternative hypotheses. Supports a relationship between mutation exposure and stronger tests, with non-randomized-study limits.

[S11] Sigstore. "Security Model." Current documentation, accessed October 23, 2026. <https://docs.sigstore.dev/about/security/>. Inspected: trust root, Fulcio/Rekor roles, monitoring, and non-guarantees. Supports the claim that unmonitored transparency may not expose misbehavior.

- [S12]** M. S. Melara, A. Blankstein, J. Bonneau, E. W. Felten, and M. J. Freedman. “CONIKS: Bringing Key Transparency to End Users.” USENIX Security 2015.
<https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/melara>. Inspected: abstract and collective audit design. Supports client participation in non-equivocation checking.
- [S13]** K. Nikitin et al. “CHAINIAC: Proactive Software-Update Transparency via Collectively Signed Skipchains and Verified Builds.” USENIX Security 2017.
<https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-nikitin.pdf>. Inspected: full paper, architecture, independent witnesses, verified builds, and evaluation. Supports threshold independent witnesses and append-only release history.
- [S14]** A. Bakker, C. B. Kirwan, M. Miller, and C. E. L. Stark. “Pattern Separation in the Human Hippocampal CA3 and Dentate Gyrus.” Science 319(5870), March 21, 2008.
<https://pubmed.ncbi.nlm.nih.gov/18356518/>. Inspected: abstract and experimental summary. Supports a narrow pattern-separation analogy only.
- [S15]** A. Ilyes et al. “The Human Hippocampus Can Pattern Separate Memories by Meaning.” Proceedings of the National Academy of Sciences 123(31), published online July 28, 2026.
<https://pubmed.ncbi.nlm.nih.gov/42520121/>. Inspected: citation metadata and full abstract. Supports recent semantic pattern-separation evidence; small human imaging study, not a computing prescription.

Internal Enceвра sources

- [E09]** Enceвра Research Round 9. Full route source inspected in Lovable project 8483bc59-da36-492e-b8b8-6bf684070241. Supports the verified-procedure carry-forward.
- [E10]** Enceвра Research Round 10, “Evidence exchange, selective revision, and a testable path toward brain-inspired intelligence.” Full route source inspected in the same project. Supports correction-versus-harm evaluation and matched-compute requirements.
- [E29]** Enceвра Research Round 29. Full manuscript inspected. Supports active source intervention and unresolved dependency/provenance equivalence.
- [E30]** Enceвра Research Round 30. Full manuscript inspected. Defines the 34-control RISE readiness gate and its 8/3/23 blocked baseline.
- [E31]** Enceвра Research Round 31, “Did the Root Actually Move? A Signed Sandbox Dry Run of the RISE Gate.” Full manuscript at `src/content/research-round-31.md` inspected. Supplies the 29/4/1 blocked carry-forward and the Round 32 handoff.

19. Next-round handoff

Round 33 should be an external-authority intervention, not another local receipt-format iteration. Establish separately administered witness and registry services, externally observable checkpoints, and a real isolation boundary. Re-run an independently authored hidden fault corpus first. Only if A3, I1, R4, and R5 become present should Round 33 activate C6 with a paired model-answer runner and matched calls, tokens, prompts, and timeouts. If those prerequisites remain unavailable, report the blocked state and do not convert local signature accuracy into a claim about reliable collaborative intelligence.