

Encevra Research Round 31

Did the Root Actually Move? A Signed Sandbox Dry Run of the RISE Gate

Encevra Research

22 October 2026

Executable sandbox conformance round
Production readiness remains BLOCKED

Contents

Abstract	2
1. Archive continuity and numbering	3
2. Research question, scope, and exclusions	4
3. Literature map and competing positions	5
4. Claim and evidence map	7
5. The implemented vertical slice	7
6. Preregistered fault suite	9
7. Baselines and decision rules	10
8. Executed results	10
9. Worked example: stale readback	11
10. The bounded brain-inspired hypothesis	12
11. Alternative explanations	12
12. Falsifiable hypotheses	13
13. Proposed next experiment: E31-C boundary hardening and blinded mutation holdout	14
14. Readiness re-audit	15
15. Expected failure modes	16
16. Limitations	17
17. Practical conclusions	17
18. Next-round handoff	18
19. Reproducibility record	18
20. Full source list and inspection notes	19
21. Final answer to the research question	20

Research program: Reliable multi-agent reasoning, evidence exchange, and answer revision

Date: 22 October 2026

Archive position: Round 31, following the completed Round 30

Status: Executable sandbox conformance round; production readiness remains **BLOCKED**

Project: Encevera Research, Lovable project 8483bc59-da36-492e-b8b8-6bf684070241

Result boundary: This round ran a deterministic, file-backed sandbox and its tests. It did not run a new Encevera language-model benchmark, Noetis answer-quality benchmark, live connector intervention, provider experiment, or production mutation. The seven injected cases are authored conformance fixtures. Their detection rates do not estimate field failure rates, model accuracy, or causal effect size.

Abstract

Round 30 concluded that Encevera could not yet make a credible active evidence-lineage claim. The inspected system could identify ingested documents and record useful model traces, but it could not prove that an intended upstream root was the object actually changed, that experimental arms remained isolated, or that a signed success statement was true. Round 30 therefore defined the RISE Gate: **Root identity, Intervention fidelity, Separation, and Equivalence-aware inference**. It handed Round 31 an implementation task rather than another conceptual survey.

This round asks one narrow, falsifiable question:

Can a self-contained, receipt-producing sandbox distinguish an intended root action from its actual effect, and fail closed on off-target action, partial action, false success, stale readback, cache spillover, and structural non-identifiability before any model call is allowed?

We implemented one non-destructive replace operation over a file-backed policy-document connector. The slice records distinct root, authority, provider, local-source, packet-span, and claim identities; freezes pre-action state; records intent and predicted post-state; executes the action in a separate process; has a second process rescan every root and sign an observation with Ed25519; receipts cache and memory exposure; retains explicit lineage candidates; and returns one of resolved, unresolved_equivalence, insufficient_fidelity, or contaminated. Signed receipts are added to

a local hash chain. The harness permits zero model calls and compares four gates over the same artifacts, so intervention work is not multiplied across baselines.

Seven preregistered cases were executed. Unsigned self-report and signed self-report each made 1 of 7 correct gate decisions and falsely accepted all 6 non-clean cases. A separate witness made 5 of 7 correct decisions but still accepted cache spillover and structural equivalence because it lacked exposure and candidate-set checks. Full RISE made 7 of 7 correct gate decisions on this authored suite, with zero false accepts and zero false rejects. It also rejected a modified signed payload and a modified local chain entry. Six unit tests passed.

These results establish only that the implementation detects the faults it was explicitly designed to detect. They do not establish generalization, independent security, or downstream reasoning benefit. The same host, operating-system user, codebase, and orchestrator still contain the executor and witness. The append-only structure is only a locally writable hash chain. No ephemeral VM or container boundary exists. No paired answer benchmark was run. On the 34-control gate, the sandbox scores **29 present, 4 partial, and 1 absent**. The five critical failures are A3, I1, R4, R5, and C6. The verdict therefore remains **BLOCKED** for live E30-B.

The practical conclusion is sharper than either “sign everything” or “multi-agent review is enough.” Integrity, truth, and causal identification are different properties. A signature can protect a false statement. A second process can still share the same failure domain. A faithful action can remain non-identifying. The next round should harden the trust and isolation boundaries, add external receipt registration, and implement the paired matched-compute runner before any live pilot is authorized.

1. Archive continuity and numbering

The original automation request named Round 11. The live archive no longer ends at Round 10. Before selecting a topic, this run inspected the current Research overview, rounds index, publications list, research registry, and the full texts of Rounds 30 and 29, plus the full route texts of Rounds 10 and 9. Round 30 is the latest completed research round in the archive. The correct next sequential number is therefore **Round 31**.

This correction matters substantively, not only administratively. Round 11 would now collide with an existing archive item, and the program has advanced far beyond the question framed after Round 10. Round 30 gives a precise implementation handoff. Repeating uncertainty calibration, selective revision, evidence exchange, or reusable procedural learning would be redundant.

1.1 Carry-forward from Round 9

Round 9 argued that reusable procedures should be learned from verified experience rather than from fluent but unverified traces. Round 31 applies that constraint to intervention records. An executor's claim that an action succeeded is not verified experience merely because it is detailed, signed, or repeated. A reusable procedure should be learned only after intended action, observed target, resulting state, exposure, and residual ambiguity are separately recorded.

1.2 Carry-forward from Round 10

Round 10 developed claim-level evidence exchange and selective answer revision. It emphasized recoverable evidence, dependency-sensitive correction, and metrics that detect harmful revision. Round 31 does not re-study those mechanisms. It supplies a missing upstream precondition: if a later collaboration trial says that a source intervention caused an answer to change, the system must first show which root was addressed, what actually changed, and whether another arm was exposed.

1.3 Carry-forward from Round 29

Round 29 asked which source actually caused an answer to move. It showed that passive disagreement or output change cannot, by itself, identify root structure. Its handoff required root-addressed action,

an independent receipt, negative controls, preserved candidate sets, an unresolved-equivalence state, and matched compute.

1.4 Carry-forward from Round 30

Round 30 audited the actual repository path and found 8 present, 3 partial, and 23 absent controls, including 26 critical failures. It defined the 34-control RISE Gate and explicitly prohibited E30-B until a sandbox could produce and independently verify intervention receipts. Its Round 31 handoff required:

1. root and authority identity in the evidence schema and frozen snapshot;
2. one non-destructive root-addressing operation;
3. separate intent, witness, separation, and budget records;
4. a signed statement from outside the executor path;
5. four explicit terminal outcomes;
6. a rerun of the 34-control checker;
7. off-target, stale-readback, spillover, and false-success injections; and
8. a fail-closed decision on whether E30-B may begin.

Round 31 follows that handoff directly.

2. Research question, scope, and exclusions

2.1 Research question

Can a signed, separately witnessed sandbox prove enough about one file-backed root action to distinguish clean execution from specified fidelity, freshness, isolation, and identifiability failures before any model inference is allowed?

The question is consequential because later causal claims depend on it. If the intended root was not the actual target, or if the witness read stale state, or if another arm was exposed, a downstream output difference cannot safely be attributed to the intended action.

The question is falsifiable. The prototype fails this round if it accepts any preregistered non-clean case, rejects the clean case, cannot verify its own signatures and chain, exceeds its declared budget, or silently collapses equivalent candidates into one answer.

2.2 In scope

- One file-backed evidence type: a JSON policy document.
- One typed root operation: sandbox replace.
- Stable root, authority, provider, provider-source, version, URI, local source, span, and claim identifiers.
- Canonicalized pre-state and predicted post-state digests.
- Separate executor and witness subprocesses.
- Ed25519-signed executor and witness statements.
- Direct filesystem observation of every preregistered root and canary.
- Freshness nonce and action sequence comparison.
- Cache and memory namespace, read, write, and bypass receipts.
- Explicit candidate predictions and a first-class unresolved result.
- A local receipt hash chain and tamper checks.
- Four baselines applied post hoc to identical artifacts.
- Rerunning the Round 30 34-control gate against this slice.

2.3 Out of scope

- Production Noetis or Encevera modification.
- Live connector, SaaS provider, database, or network action.
- Model inference, agent debate, answer revision, or task utility.

- A claim that process separation on one host is independent security.
- Hardware attestation, confidential computing, HSM-backed keys, or separate administrative domains.
- A public or third-party transparency service.
- Statistical estimation of real fault prevalence.
- The 720-case E30-B experiment.
- Any claim of literal brain copying.

2.4 Why this topic is novel within the archive

Three adjacent directions were considered: confidence calibration, fixed-budget verification allocation, and provenance-preserving memory. All remain useful, but the archive already specifies a more immediate blocker. A calibrated confidence value cannot repair a misaddressed action. An optimal verification allocator cannot spend intelligently if the verifier has no trustworthy action receipt. A provenance-aware memory can preserve the wrong root identity perfectly. The distinct next question is therefore operational: can the intervention record itself be made testable?

This is not a repetition of Round 10's evidence exchange or selective revision. It does not ask whether agents should revise. It asks whether the experimental system can tell what it actually did. It is also not a repetition of Round 9's procedural learning. It produces the verification substrate that such learning would need.

3. Literature map and competing positions

No inspected source directly validates RISE for collaborative AI reasoning. The literature supplies components and warnings that can be combined into a testable design. The synthesis below distinguishes those sources from Enceavra's hypotheses.

3.1 Canonical data before cryptographic claims

RFC 8785 explains why hashing and signing require invariant serialization and specifies deterministic JSON property ordering, primitive serialization, and UTF-8 output [S05]. RFC 8032 specifies Ed25519 signatures and test vectors [S06]. Together they support an integrity layer: a verifier can detect whether a signed payload changed.

Competing interpretation: integrity is sometimes treated as evidence of truth. That is too strong. A valid signature shows that a key holder signed specific bytes. It does not show that the bytes accurately describe the world. RFC 9943 makes this limitation explicit for transparent statements: registration can show that an issuer made a statement without proving the statement true [S09]. Round 31 therefore signs both executor and witness records but never treats executor authentication as independent observation.

3.2 Self-report versus separate appraisal

The RATS architecture separates an Attester that produces Evidence, a Verifier that appraises it under policy, and a Relying Party that acts on an Attestation Result [S07]. EAT adds structured signed claims and emphasizes freshness mechanisms such as nonces [S08]. This supports the design decision to keep intent, execution report, observed effect, and final gate decision distinct.

Competing interpretation: a second process is sometimes called independent merely because it has another PID. That conclusion is not warranted here. The witness and executor share a host, user, source tree, and orchestrator. The slice calls this process separation and scores the relevant control partial. Independence must be defined by failure and authority domains, not by agent count.

3.3 Local hash chain versus external transparency

Certificate Transparency uses append-only Merkle trees, signed timestamps, inclusion proofs, consistency proofs, and monitoring to make log behavior auditable [S10]. SCITT generalizes transparent signed statements and receipts for digital supply chains [S09]. These designs support append-only registration and a distinct auditor role.

Competing interpretation: a local hash chain may be described as append-only because modifying an old entry breaks later hashes. That only makes undetected editing harder if a trusted party already holds a later checkpoint. The Round 31 chain has no external checkpoint, Merkle proof, timestamp authority, or monitor. It detects the included tamper test, but its operator could rewrite the entire chain. R5 therefore remains partial.

3.4 Build provenance as a transferable pattern

SLSA provenance records builder identity, parameters, dependencies, and generated subjects. Its higher-assurance requirements place provenance generation in a control plane, restrict key access, and require isolation that prevents one build from influencing another [S11]. in-toto link attestations bind step identity, materials, products, command, environment, and byproducts [S12]. W3C PROV distinguishes entities, activities, agents, derivations, and responsibility [S13].

These sources support explicit lineage and step records. They do not establish that supply-chain provenance improves AI answer revision. Round 31 adapts their structure, not their certification claims.

3.5 Attempted action versus actual treatment

Choo, Shiragur, and Uhler study causal discovery under off-target interventions, where an attempted intervention may affect a random subset of vertices [S14]. The exact setting differs, but the central separation transfers: attempted target and actual target are different variables.

Competing interpretation: an API success response may be taken as treatment assignment. In the prototype, it is only the executor's claim. The witness rescans every root to observe what changed. This is the mechanism tested by the off-target, partial-write, and false-success cases.

3.6 Isolation labels versus measured exposure

Aronow and Samii show that experiments with interference require a treatment design, an exposure mapping, and an estimand [S15]. Savje, Aronow, and Hudgens show why unknown interference can invalidate conventional uncertainty claims even when some estimands remain interpretable [S16].

Competing interpretation: assigning two calls different `arm_id` values is enough to make them independent. It is not. Cache, memory, provider session, retrieval, and filesystem state can create cross-arm exposure. Round 31 records cache and memory accesses and inserts a foreign cache canary. This is only a small start because the sandbox does not isolate network or provider state.

3.7 Fault injection: useful but easy to overclaim

Fault injection is an established way to evaluate system dependability under controlled failures [S17]. It is appropriate for checking whether a detector responds to a known condition.

Competing interpretation: perfect performance on authored faults shows real-world reliability. It does not. The authors selected both the faults and the detection rules. The suite is a conformance test. It provides no prevalence estimate, no unseen holdout, and no measure of adversarial generalization.

3.8 Brain-inspired analogy, bounded and testable

Corollary discharge research describes a copy of a motor command routed to sensory systems so predicted consequences can be compared with sensory input [S18]. The useful analogy is functional: copy the intended root action to an observer, predict what a successful action should change, and compare that prediction with separately observed state.

The analogy yields a testable design idea:

A protocol that compares an intent copy with independently observed consequences should reject more false-success and off-target cases than a protocol that trusts the executor's report.

That is H31.2 below. Nothing in this round claims that the software architecture reproduces a brain, that corollary discharge is the only biological mechanism involved, or that biological similarity is evidence of correctness.

4. Claim and evidence map

Claim	Epistemic status
Stable serialization is needed for repeatable JSON hashes and signatures.	Established technical result
Ed25519 can authenticate an exact receipt payload and detect payload modification.	Established technical result;
A valid signature does not prove that the signed statement is true.	Established conceptual limita
Attempted and actual intervention targets must be represented separately.	Established in adjacent caus
Interference requires explicit exposure modeling.	Established statistical princip
A separate witness process detects the authored action-fidelity faults that self-report misses.	Observed in this sandbox onl
Cache receipts and candidate sets add detection beyond direct action witnessing.	Observed in this sandbox onl
Process separation on one host is independent verification.	Not established; treated as fa
The local hash chain is an append-only transparency service.	Not established; explicitly rej
Full RISE improves answer reliability at fixed compute.	Encevra hypothesis, untested
The intent-observation comparison is brain-like in a literal sense.	Excluded claim

5. The implemented vertical slice

5.1 Connector and evidence object

The connector is deliberately simple: two JSON files represent an intended policy root and a preregistered decoy root. A third file is a negative-control canary. The action is a sandbox replace that changes the intended policy from version 1 to version 2.

Each root document records:

- root_id;
- authority_id;
- provider;
- provider source ID;
- provider version;
- source URI;
- root version;
- observation time;
- validity interval; and
- content.

The run snapshot keeps the root distinct from source:policy-primary-v1, which is the local ingested source. A typed derivation graph connects root, channel, ingested source, packet span, and claim. This directly addresses the local-ID-versus-root-ID confusion found in Round 30.

5.2 Intent record

Before execution, the control harness records:

- action ID and sequence;

- action type;
- intended root and authority;
- pre-action digest;
- predicted post-action digest;
- predicted changed-root set;
- a freshness nonce;
- issue time; and
- replacement payload.

The intent is immutable for the case once the executor starts. The expected post-state is therefore a preregistered prediction, not a description written after observation.

5.3 Executor record

The executor subprocess receives the action and its own signing key. It performs the action or injected fault, writes a readback record, records cache and memory accesses, and emits a signed self-report.

Every fixture intentionally makes the self-report optimistic: status is success, claimed target is the intended root, and claimed digest is the expected digest. This is not a realistic prevalence model. It isolates the question of whether signing or repetition can substitute for separate observation.

5.4 Witness record

After the executor exits, a second subprocess:

1. reads the frozen intent and pre-state;
2. directly rehashes the intended root, decoy root, and canary;
3. computes the actual changed-root set;
4. compares the executor readback nonce, sequence, and digest with direct state;
5. inventories cache and memory exposure;
6. records the trust-boundary limitations; and
7. signs the resulting payload with a different Ed25519 key.

The witness private key is stored in an ephemeral mode-0600 directory and is not passed to the executor subprocess. This is meaningful process-level separation, but it is not independent security because the same host user and orchestrator can access both processes.

5.5 Canonicalization and signatures

Receipt payloads are serialized as compact, sorted-key UTF-8 JSON before SHA-256 hashing and Ed25519 signing. The fixture value domain contains strings, integers, booleans, nulls, lists, and dictionaries. The implementation is compatible with the JCS subset needed for this corpus, but it is not advertised as a full RFC 8785 implementation because it does not implement every number and Unicode edge case.

The verifier checks the algorithm, payload digest, public key, and signature. Changing the signed payload without recomputing the signature is rejected.

5.6 Separation record

Each arm records:

- its workspace scope;
- cache namespace, reads, writes, and bypass state;
- memory namespace, reads, and writes;
- known external influences;
- the root canary digest; and
- foreign cache or memory reads found by the witness.

The cache-spillover fixture inserts `cache:foreign-arm:ROUND31-SPILOVER-CANARY`. The B3 verifier treats any such exposure as contaminated even when the intended root action is otherwise correct.

5.7 Candidate set and terminal outcomes

Each case preserves candidate lineage structures and their predicted observations under replace. The verifier does not force a winner when multiple candidates make the same prediction.

The four terminal outcomes are:

Outcome	Meaning
resolved	The signed observation matches intent, isolation receipts are clean, and the permitted action d
unresolved_equivalence	Action fidelity is adequate, but at least two candidates remain observationally equivalent.
insufficient_fidelity	Target, digest, readback, freshness, or action semantics do not establish that the intended trans
contaminated	Off-target change, negative-control change, or cross-arm exposure invalidates clean interpretat

These states separate two uncertainties that were previously easy to merge: “we do not know what action happened” and “we know what action happened, but it cannot distinguish the structures.”

5.8 Receipt registration

Every signed witness receipt is appended to a JSON Lines file. Each entry contains the previous entry hash plus hashes of the receipt payload and signature. A deterministic verifier recomputes the chain, and a mutation test confirms that changing an included entry breaks verification.

This mechanism is useful for local tamper evidence. It is not a transparency service. An operator with write access can regenerate the entire chain. R5 stays partial until an external service returns independently monitorable receipts or checkpoints.

5.9 Budget record

Each case permits:

- zero model calls;
- zero input and output tokens;
- two subprocesses, one executor and one witness; and
- a 5,000 ms wall-time cap.

All seven cases used two subprocesses, zero model calls, and remained inside the wall-time cap. The four baseline decisions are computed from the same completed artifacts. Therefore B3 did not receive additional action attempts or model compute relative to B0 through B2 in this dry run.

6. Preregistered fault suite

Case	Injected condition	Expected full outco
clean_on_target	Exact predicted replacement of intended root	resolved
off_target	Decoy changes; self-report claims intended root	contaminated
partial_write	Intended root changes to an unpredicted partial payload	insufficient_fid
false_success	No root changes; self-report claims success	insufficient_fid
stale_readback	Correct action occurs, but readback uses pre-action digest and stale nonce	insufficient_fid
cache_spillover	Correct action plus foreign-arm cache canary exposure	contaminated
equivalent_candidates	Correct action; two candidates make the same prediction	unresolved_equiv

The fixture registry is fixed in code before execution. Expected outcomes are retained by the parent harness and are not written into the executor-visible snapshot. The executor necessarily receives the selected fault injection because it must enact it, so this is not a security-adversarial blind test. It is only an oracle-label separation for conformance evaluation.

7. Baselines and decision rules

The comparison uses the same receipts and state for every baseline.

Baseline	Information used	Acceptance rule
B0: self-report	Executor payload only	Accept status=succes
B1: signed self-report	Executor payload plus signature	Accept valid signature
B2: dual-path witness	Signed witness, direct post-state, changed roots, nonce, sequence, canary	Accept only exact inter
B3: full RISE	B2 plus cache/memory receipts and candidate set	Accept only resolved

The B0 and B1 rules are intentionally simple because the question is whether authentication alone changes truth discrimination when the reporter is the same. B2 and B3 differ only in the information used by the gate, not in action opportunities.

8. Executed results

8.1 Case-level results

Case	B0	B1	B2	B3 outcome	Expected match
Clean on-target	Accept	Accept	Accept	resolved	Yes
Off-target	Accept	Accept	Reject	contaminated	Yes
Partial write	Accept	Accept	Reject	insufficient_fidelity	Yes
False success	Accept	Accept	Reject	insufficient_fidelity	Yes
Stale readback	Accept	Accept	Reject	insufficient_fidelity	Yes
Cache spillover	Accept	Accept	Accept	contaminated	Yes
Equivalent candidates	Accept	Accept	Accept	unresolved_equivalence	Yes

8.2 Aggregate gate metrics

Baseline	Correct gate decisions	False accepts	False rejects	Total
B0: self-report	1	6	0	7
B1: signed self-report	1	6	0	7
B2: dual-path witness	5	2	0	7
B3: full RISE	7	0	0	7

The full outcomes were two contaminated, three insufficient_fidelity, one resolved, and one unresolved_equivalence, exactly matching the preregistered fixture labels.

8.3 Integrity checks

- All witness receipts verified with Ed25519.
- The seven-entry local receipt chain verified.
- Mutating a signed payload caused signature verification to fail.
- Mutating a local chain entry caused chain verification to fail.
- No private key file was exported into the result package.
- Six unit tests passed.
- Every case remained inside its declared budget.

8.4 What these results do and do not show

They show that the implemented comparator can detect the exact discrepancies encoded in the suite. They also provide a concrete counterexample to “a signed success report is enough”: B1 is no better than B0 because both authenticate the same optimistic claim.

They do not show that B3 will generalize to new fault families, adversarial implementations, opaque providers, production connectors, or model-answer tasks. There are only seven authored cases. Confidence intervals would be misleading because the cases are not a sample from a target population. The correct interpretation is pass or fail against a small conformance specification.

9. Worked example: stale readback

The stale-readback case is useful because the root action itself succeeds. A gate that checks only final target state may accept it, while a gate that requires a coherent observation chain should refuse to interpret the run.

9.1 Frozen intent

The intent says:

- replace root:policy-primary;
- expected pre-state digest = (d_0);
- expected post-state digest = (d_1);
- predicted changed-root set = {root:policy-primary};
- action sequence = 1; and
- nonce = round31-nonce-05-stale_readback.

9.2 Actual execution

The executor writes the exact replacement, so a direct filesystem scan observes the target at (d_1). The decoy and canary remain unchanged.

9.3 Faulty readback

The executor's readback reports:

- digest (d_0), the pre-action state;
- sequence 0;
- nonce stale-pre-action-nonce; and
- source pre-action-cache.

The signed executor self-report still claims success and (d_1).

9.4 Gate behavior

- **B0 accepts** because the self-report says success.
- **B1 accepts** because the false success statement is validly signed by the executor.
- **B2 rejects** because direct observation (d_1) conflicts with readback (d_0), the nonce is wrong, and the observation sequence precedes the action.
- **B3 returns insufficient_fidelity** for the same reason. It does not call the action failed, because the target did move. It says the receipt path is incoherent and therefore cannot support the planned causal inference.

This is the key distinction between outcome inspection and receipt validity. A favorable final state does not retroactively make a stale observation trustworthy.

10. The bounded brain-inspired hypothesis

The corollary-discharge analogy contributes one mechanism, not a biological claim. In the analogy, an action command is copied to a predictive observer. The observer anticipates sensory consequences, then compares prediction with input. In RISE:

- the intent record is the command copy;
- the expected digest and changed-root set are predicted consequences;
- the witness scan is observed input; and
- the verifier computes a structured prediction error.

This yields two testable ablations:

1. remove the intent copy while keeping post-state observation;
2. keep the intent but replace direct observation with executor self-report.

If either ablation performs as well as full comparison on unseen fidelity faults, the analogy-derived mechanism has not earned its complexity. The analogy does not imply that software should copy neural anatomy, nor that biological origin confers correctness.

11. Alternative explanations

11.1 The rules were written for the faults

Yes. The suite is white-box conformance testing. Perfect fixture performance can reflect specification alignment rather than robustness. The next step needs generated mutations and a frozen holdout maintained outside the implementation path.

11.2 B1 failed because its baseline was deliberately weak

B1 is deliberately narrow, but it tests a real misconception: that signing a self-report makes it an independent receipt. It does not compare against a production-grade remote attestation service. A stronger B1 would still need to answer who measured the state and which authority controlled the key.

11.3 B3 wins by rejecting more

In this seven-case suite, the only clean case was accepted and all non-clean cases were rejected, so there was no observed over-rejection. That says little about broader coverage. E30-B must report selective risk, structural-decision coverage, and unnecessary-unresolved rate together. B3 should be rejected if it reduces error only by becoming inert.

11.4 The witness could simply be wrong

Correct. A witness signature protects its statement from undetected modification but cannot guarantee accurate measurement. The same-host boundary makes correlated error plausible. Independent measurement, privilege separation, challenge-response freshness, and compromise testing remain necessary.

11.5 The local chain is enough for a small system

It may be operationally useful, but it does not meet the gate's meaning of external verifiability. Without an independently held checkpoint, the operator can rewrite history. The readiness score correctly remains partial.

11.6 No agent was involved, so the result is irrelevant

The slice intentionally stops before model calls. It tests infrastructure claims that must hold regardless of model behavior. This is relevant as a prerequisite, not as evidence of answer improvement.

11.7 A simpler deterministic verifier is sufficient

For this stage, yes. The verifier is intentionally deterministic. If future full RISE logic can be replaced by smaller rules with equal detection and lower cost, Encevera should use the smaller rules. RISE is a contract, not an entitlement to architectural complexity.

12. Falsifiable hypotheses

H31.1 - Signature is integrity, not truth

Hypothesis: When the same executor authors both the action and success claim, adding a valid signature will not materially improve detection of off-target, partial, false-success, or stale-readback faults relative to unsigned self-report.

Round 31 observation: B0 and B1 each made 1 of 7 correct gate decisions and 6 false accepts.

Future rejection rule: Reject this hypothesis if a frozen holdout shows that executor signing alone improves fault detection for reasons not attributable to leaked labels or different information.

H31.2 - Intent plus separate observation detects fidelity mismatch

Hypothesis: Comparing frozen intent with a separately signed direct observation will detect at least 95% of off-target, partial, false-success, and stale-readback holdout faults at no more than 2% false rejection on clean cases.

Round 31 observation: B2 detected all four represented fidelity faults and accepted the clean case. The sample is too small and authored to estimate the proposed rates.

Reject if: either sensitivity or clean-case specificity misses its threshold on the frozen holdout.

H31.3 - Exposure receipts add information beyond action witnessing

Hypothesis: Cache and memory receipts plus hidden canaries reduce undetected spillover by at least 75% relative to a direct-action witness without exposure telemetry, with false-positive rate below 5%.

Round 31 observation: B2 accepted the cache-spillover case; B3 returned contaminated.

Reject if: the relative reduction or false-positive threshold is missed on varied hidden canaries and spillover routes.

H31.4 - Explicit equivalence prevents false structural certainty

Hypothesis: A candidate-set representation with unresolved equivalence reduces false root-independence by at least 30% relative to forced binary classification while retaining at least 70% structural-decision coverage.

Round 31 observation: B3 preserved the equivalent candidate set; B0 through B2 accepted without expressing residual ambiguity. No false-root-independence benchmark was run.

Reject if: the full experiment misses either error reduction or coverage floor.

H31.5 - Stronger boundaries are necessary for production readiness

Hypothesis: Moving the witness signer and receipt registry into separately administered, policy-enforced services and running each arm in an ephemeral isolated environment will close A3, I1, R4, and R5 without introducing more than 2% accounting overhead or unacceptable operational failure.

Reject if: compromise tests show shared control, key exposure, history rewriting, cross-arm contamination, or budget overrun.

H31.6 - End-to-end value remains unproven

Hypothesis: Once the infrastructure gate passes, full RISE will reduce false-root independence and harmful revision relative to the strongest active baseline at matched total compute, without more than a 1.5 percentage-point answer-accuracy loss.

Round 31 observation: Not tested. No answer benchmark was run.

Reject if: the primary endpoints do not improve, regression margins are crossed, or benefits disappear under identifier blinding.

13. Proposed next experiment: E31-C boundary hardening and blinded mutation holdout

E30-B remains too large and too semantically loaded for the current boundary. The next experiment should close the remaining readiness failures before any answer-level pilot.

13.1 System variants

Variant	Witness boundary	Arm isolation
C0	Executor self-report	Logical directory only
C1	Same-host witness process	Fresh directory
C2	Separate service account and policy boundary	Ephemeral container with denied cross-arm mounts and network
C3	Separate administrative service or host	Ephemeral container or VM plus audited cache/memory namespaces

13.2 Fault generator and holdout

Freeze a mutation grammar before scoring. It should vary:

- path alias and symlink target;
- off-target root and multiple-target write;
- partial JSON field replacement;
- success response without write;
- write followed by rollback;
- stale cache age and nonce replay;
- delayed read after timeout;
- cache and memory cross-arm reads;
- canary encoding and location;
- signer-key exposure attempt;
- receipt omission, duplication, reordering, and chain fork;
- candidate sets with exact, partial, and action-dependent equivalence.

Keep at least 40% of generated cases in a blinded holdout stored outside the implementation repository. The implementation team may see the grammar but not the exact holdout instances or labels.

13.3 Controlled budget

Per case and variant:

- one root action slot;
- one witness observation slot for variants that have a witness;
- identical maximum filesystem operations;
- identical maximum wall time;
- zero model calls;
- zero network calls except the preregistered registry endpoint;
- all signing, registry, verification, retry, and cleanup costs charged to the variant; and

- failure to stay inside the cap scored as a budget failure.

Only after C3 passes should a small E30-B pilot allow model calls. That pilot must restore Round 30's matched caps for tokens, calls, tools, cost, latency, and provider mix.

13.4 Primary metrics

1. False acceptance of a non-clean action.
2. False rejection of a clean action.
3. Off-target detection sensitivity and precision.
4. Stale-readback detection sensitivity and precision.
5. Cross-arm exposure detection sensitivity and precision.
6. Unresolved-equivalence recall and unnecessary-unresolved rate.
7. Receipt-verification failure under payload, ordering, fork, and replay mutation.
8. Compromise containment: whether executor access can sign a witness statement or rewrite externally checkpointed history.

13.5 Regression and harm metrics

Risk	Metric	Gate
Detector becomes inert	Clean-case acceptance and unnecessary-unresolved rate	Clean false rejection below
Fault remains undetected	False acceptance	Below 1% on critical holdout
Isolation is cosmetic	Unauthorized canary exposure	Zero tolerance
Signer is agent-writable	Executor-to-witness key-access test	Zero successful access
Registry is rewritable	Fork, deletion, and history-rewrite tests	All detected by independent
Freshness can be replayed	Nonce and sequence replay acceptance	Zero acceptance
Compute is hidden	Fully reconciled operations, wall time, and service cost	Within 2% of cap
Cleanup causes carryover	Post-case rollback verification	Zero unreported failures
Privacy or authorization is violated	Unauthorized root read/write	Zero tolerance

13.6 Ablations

- Remove separate authority identity.
- Remove direct root scan and use executor readback.
- Remove nonce and action sequence.
- Remove the decoy root.
- Remove the negative-control canary.
- Remove cache receipt.
- Remove memory receipt.
- Remove external checkpointing.
- Remove independent monitor.
- Collapse the candidate set to a forced label.
- Reuse an environment across arms.

Each ablation must keep the same total action and verification budget. An ablation that saves compute may spend the remainder on preregistered neutral work, or results must be analyzed on an explicit cost frontier.

14. Readiness re-audit

The Round 30 manifest contained 34 controls. The same atomic controls were applied to the sandbox implementation and emitted artifacts.

14.1 Change summary

Status	Round 30	Round 31 sandbox	Change
Present	8	29	+21
Partial	3	4	+1
Absent	23	1	-22
Critical failures	26	5	-21
Verdict	BLOCKED	BLOCKED	No release

This comparison is scoped carefully. Round 30 audited the production repository path at a pinned commit. Round 31 scores a self-contained prototype. The improved count does not mean the production application inherited these mechanisms.

14.2 Remaining critical failures

Control	Status	Why it still fails
A3: independently observe actual target	Partial	Witness is a separate process but shares host, user, source tree, and
I1: ephemeral arm environment	Partial	Fresh directories are not containers, VMs, network namespaces, or s
R4: sign outside experiment-controlled path	Partial	Witness key is withheld from executor invocation, but same-host au
R5: append-only verifiable registration	Partial	Local chain lacks external checkpoints, proofs, and independent mo
C6: paired matched-compute answer runner	Absent	No model or answer benchmark was run.

14.3 Decision

The gate is fail-closed. Any partial or absent critical control means **BLOCKED**. Therefore:

- do not start the 720-case E30-B trial;
- do not make a production intervention claim;
- do not interpret the 7-case result as an answer-quality result; and
- proceed only to boundary hardening and a blinded mutation holdout.

15. Expected failure modes

1. **Canonicalization mismatch.** Another implementation serializes a number or Unicode string differently and invalidates a legitimate signature.
2. **Same-host compromise.** The executor reads or replaces the witness key despite process separation.
3. **Witness code reuse.** Shared helper logic gives executor and witness the same bug.
4. **Root alias escape.** A symlink, provider alias, or hidden administrative mapping changes a different object than the visible path suggests.
5. **Partial success semantics.** A provider applies some fields while reporting a generic success state.
6. **Stale but fresh-looking readback.** A cache copies the new nonce onto old content.
7. **Time-of-check to time-of-use race.** State changes after observation but before the receipt is relied upon.
8. **Canary cueing.** A visible marker changes system behavior instead of merely detecting exposure.
9. **Unobserved remote state.** A local connector receipt omits provider-side queues, replicas, or policy propagation.
10. **Local chain rewrite.** An operator regenerates the entire chain and publishes only the rewritten checkpoint.
11. **Candidate language misspecification.** The true lineage structure is not represented among candidates.

12. **Outcome laundering.** unresolved_equivalence is counted as success without reporting coverage.
13. **Budget laundering.** Signing, verification, retries, registry calls, or cleanup are excluded from compute accounting.
14. **Rollback persistence.** A sandbox change leaks into a later case.
15. **False independence from clean local state.** Opaque provider memory or shared infrastructure creates exposure outside the observed receipt.
16. **Overfitting to named faults.** The verifier catches exact fixtures but fails on composition or semantic variants.

Each failure should map to a field, mutation test, stop condition, or explicit unresolved state. An unreceipted failure cannot be treated as harmless missing data.

16. Limitations

- The suite has seven authored cases and no probabilistic sampling frame.
 - The authors wrote the fault injector, verifier, and expected outcomes.
 - The executor and witness share a host, user, source, runtime, and parent process.
 - The private-key directory is permission-restricted but not hardware-backed or separately administered.
 - The serializer implements only the constrained data subset used by the fixture and is not a general JCS library.
 - The local hash chain is operator-rewritable and has no independent checkpoint.
 - Fresh directories do not prevent shared kernel, filesystem, process, or network influence.
 - Cache and memory receipts cover only the explicit fixture files, not hidden runtime or provider state.
 - Only a file-backed connector was implemented. Database, API, object-store, and event-stream semantics differ.
 - The candidate language is tiny and manually specified.
 - Only one action type was implemented.
 - No timeout, rejection, rollback, or unknown-target case was executed, although the outcome vocabulary represents them.
 - No privacy, authorization, or multi-tenant threat model was validated.
 - No new Encevera benchmark, model call, agent collaboration, answer revision, or task utility measurement was performed.
 - The improved 34-control score applies to the prototype, not automatically to the production repository.
 - The literature supports components and cautions, not the end-to-end effectiveness of RISE for Encevera.
 - The proposed numerical thresholds are design commitments to test, not literature-derived facts.
-

17. Practical conclusions

1. **The root can be made observable in a small vertical slice.** Root, authority, lineage, intent, predicted state, actual state, exposure, and candidate ambiguity can coexist in one machine-verifiable record.
2. **Signing self-report is insufficient.** It protects bytes, not truth. In this suite, signed and unsigned self-report were behaviorally identical.
3. **Separate observation is the largest first gain.** It caught off-target, partial, false-success, and stale-readback fixtures that self-report missed.
4. **Action fidelity is not the whole gate.** A clean action can coexist with cache contamination or structural non-identifiability.
5. **Process count is not trust independence.** The witness boundary remains partial until authority and failure domains are separated.

6. **A local chain is not external transparency.** It is useful tamper evidence but cannot make its own history credible.
 7. **Unresolved equivalence is a result.** Faithful intervention does not guarantee identification.
 8. **The sandbox must not leak into production claims.** The current production gate remains blocked.
 9. **The next investment should be boundary hardening, not more conceptual prose.** Separate the signer and witness, isolate arms, register receipts externally, and build the paired runner.
 10. **RISE should still be rejected if it does not improve downstream outcomes at fixed compute.** Passing infrastructure controls is necessary, not sufficient.
-

18. Next-round handoff

Round 32 should implement **E31-C boundary hardening and blinded mutation holdout**.

Required deliverables:

1. move the witness into a separately authorized service account or host;
2. ensure the executor cannot read, replace, or invoke the witness signing key outside policy;
3. run every arm in an ephemeral container or VM with audited mounts, cache, memory, and network policy;
4. register signed receipts in an external append-only service with inclusion and consistency evidence plus an independent checkpoint;
5. freeze a mutation grammar and keep exact holdout fixtures outside the implementation repository;
6. exercise rollback, timeout, rejection, unknown-target, replay, chain-fork, and composed faults;
7. implement the paired matched-compute runner and accounting reconciliation, but keep model calls disabled until A3, I1, R4, and R5 pass;
8. rerun the 34-control gate; and
9. authorize a small E30-B model pilot only if every critical control is present.

If separate authority or external registration cannot be built, Round 32 should preserve the blocked verdict and document the constraint. It must not rename same-host process separation as independent attestation.

19. Reproducibility record

The accompanying package contains:

- `round31_rise_sandbox.py`, the executable harness;
- `test_round31_rise_sandbox.py`, six unit tests;
- `results_final/`, with all seven case snapshots, intents, signed executor reports, signed witness receipts, access receipts, budgets, verification results, public keys, chain, and summary;
- `readiness_manifest.json`, the 34-control re-audit;
- `readiness_summary.json`, the deterministic gate output;
- `audit_readiness.py`, the manifest validator;
- `source_inspection_ledger.csv`, with dates, links, inspected material, supported claims, and limits;
- this paper in Markdown and PDF; and
- a checksum manifest.

19.1 Reproduction commands

From the repro directory:

```
python -m unittest -v test_round31_rise_sandbox.py
python round31_rise_sandbox.py run --output results_rebuilt
python audit_readiness.py --output readiness_summary_rebuilt.json
```

The exact Ed25519 keys and signatures change on each run because keys are generated ephemerally. Semantic outcomes, baseline counts, and readiness counts should remain the same. This is intentional: reproducibility applies to the decision logic, not to reusing a fixture private key.

20. Full source list and inspection notes

Every public source below was opened directly on 22 October 2026. The machine-readable ledger records the same information in tabular form.

Enceavra archive

[S01] Enceavra Research. Round 30: Can We Intervene for Real? A Fail-Closed Readiness Gate for Active Evidence-Lineage Trials. 21 October 2026. Inspected full paper, 34-control manifest, conclusions, and Round 31 handoff in Lovable project 8483bc59-da36-492e-b8b8-6bf684070241. Supports the present implementation assignment and fail-closed rule. Internal program evidence, not independent validation.

[S02] Enceavra Research. Round 29: Which Source Actually Caused the Answer to Move? 20 October 2026. Inspected full paper, diagnostic outputs, intervention ladder, failure modes, and handoff. Supports root-addressed action, equivalence preservation, off-target controls, and matched compute. Internal program evidence.

[S03] Enceavra Research. Round 10: Evidence exchange, selective revision, and a testable path toward brain-inspired intelligence. 2026. Inspected full route text and conclusions. Supports claim-level evidence exchange, selective revision, correction, and harm metrics. This round does not repeat those mechanisms.

[S04] Enceavra Research. Round 9. 2026. Inspected full route text and conclusions. Supports learning reusable procedures only from verified experience.

Public technical and research sources

[S05] Rundgren, A., Jordan, B., and Erdtman, S. JSON Canonicalization Scheme (JCS). RFC 8785, June 2020. <https://www.rfc-editor.org/rfc/rfc8785.html>. Inspected abstract, status, canonical serialization, recursive property sorting, UTF-8 generation, and security considerations. Supports invariant serialization before hashing or signing. Informational RFC; the prototype implements only its constrained fixture subset.

[S06] Josefsson, S., and Liusvaara, I. Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032, January 2017. <https://www.rfc-editor.org/rfc/rfc8032.html>. Inspected Ed25519 signing and verification, key and signature sizes, test vectors, and security considerations. Supports receipt integrity, not statement truth.

[S07] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and Pan, W. Remote ATtestation procedureS (RATS) Architecture. RFC 9334, January 2023. <https://www.rfc-editor.org/rfc/rfc9334.html>. Inspected Attester, Verifier, Relying Party, Evidence, Attestation Result, appraisal, freshness, and trust-boundary roles. Supports separating claim production from appraisal.

[S08] Lundblade, L., Mandyam, G., and O'Donoghue, J. The Entity Attestation Token (EAT). RFC 9711, January 2025. <https://www.rfc-editor.org/rfc/rfc9711.html>. Inspected claims sets, profiles, nonce, freshness, and signed-token structure. Supports explicit freshness. The prototype does not claim EAT conformance.

[S09] Birkholz, H., Steele, O., Tschofenig, H., and Wittenzellner, B. An Architecture for Trustworthy and Transparent Digital Supply Chains. RFC 9943, June 2026. <https://www.rfc-editor.org/rfc/rfc9943.html>. Inspected Issuer, Transparency Service, Auditor, Statement, Receipt, append-only sequence, and security limits. Supports independent registration and the distinction between issued and true statements.

[S10] Laurie, B., Messeri, E., and Stradling, R. Certificate Transparency Version 2.0. RFC 9162, December 2021. <https://www.rfc-editor.org/rfc/rfc9162.html>. Inspected append-only Merkle trees,

signed timestamps, inclusion and consistency proofs, monitors, and split-view limits. Supports why a local chain is weaker than externally audited transparency.

[S11] SLSA project. SLSA v1.2 Build Provenance and Requirements for Producing Artifacts. Version 1.2, current specification inspected 22 October 2026. <https://slsa.dev/spec/v1.2/build-provenance> and <https://slsa.dev/spec/v1.2/build-requirements>. Inspected builder identity, parameters, dependencies, control-plane provenance, key access, isolation, ephemeral environments, and verification. Supports adapting control-plane signing and isolation patterns. This is not a SLSA certification.

[S12] in-toto project. Link Attestation v0.3. <https://in-toto.io/attestation/link/v0.3>. Inspected state-ment subject, materials, command, byproducts, and environment. Supports per-step action records. The prototype borrows the pattern but does not claim in-toto conformance.

[S13] W3C Provenance Working Group. PROV-DM: The PROV Data Model. W3C Recommendation, 30 April 2013. <https://www.w3.org/TR/prov-dm/>. Inspected entities, activities, agents, responsibility, derivation, bundles, temporal relations, and the mutable-entity caveat. Supports typed lineage and authority identity. Provenance alone does not establish causality.

[S14] Choo, D., Shiragur, K., and Uhler, C. Causal Discovery under Off-Target Interventions. arXiv:2402.08229, 13 February 2024. <https://arxiv.org/abs/2402.08229>. Inspected abstract and stochastic intervention model in which attempted action may produce an actual intervention on a random subset of vertices. Supports separating intended from actual target. Transfer to AI evidence roots is an Enceavra hypothesis.

[S15] Aronow, P. M., and Samii, C. Estimating Average Causal Effects Under General Interference, with Application to a Social Network Experiment. arXiv:1305.6156, submitted 27 May 2013, revised 20 June 2018. <https://arxiv.org/abs/1305.6156>. Inspected design, exposure mapping, estimands, and interference-aware inference. Supports explicit exposure records.

[S16] Savje, F., Aronow, P. M., and Hudgens, M. G. Average Treatment Effects in the Presence of Unknown Interference. arXiv:1711.06399, 17 November 2017. <https://arxiv.org/abs/1711.06399>. Inspected abstract and central unknown-interference result. Supports conservative treatment of unmeasured spillover. It does not prescribe software receipts.

[S17] Hsueh, M.-C., Tsai, T. K., and Iyer, R. K. Fault Injection Techniques and Tools. Computer 30(4), April 1997, pp. 75-82. <https://doi.org/10.1109/2.585157>. Inspected abstract and bibliographic record. Supports injected faults as a dependability evaluation method. Authored injections do not estimate deployed failure prevalence.

[S18] Crapse, T. B., and Sommer, M. A. Corollary discharge circuits in the primate brain. Current Opinion in Neurobiology 18(6), December 2008. <https://pmc.ncbi.nlm.nih.gov/articles/PMC2702467/>. Inspected summary and introduction on command copies, predicted consequences, and sensory comparison. Supplies a bounded functional analogy only.

[S19] Python Cryptographic Authority. Ed25519 signing - cryptography documentation. Version 3.1 documentation. <https://cryptography.io/en/3.1/hazmat/primitives/asymmetric/ed25519/>. Inspected key generation, signing, verification, raw key serialization, and invalid-signature behavior. Supports implementation details, not production key custody.

21. Final answer to the research question

Yes, but only at conformance scale and not with production-grade independence. The sand-box can distinguish the intended action from specified actual effects, detect the injected fidelity and spillover faults, preserve structural equivalence, and produce verifiable signed receipts. It therefore demonstrates that the RISE contract is implementable for one simple connector.

No, it is not ready to authorize the live trial. The witness, isolation, signer, and receipt registry remain inside too much shared authority, and the paired answer-level runner does not exist. The correct scientific result is not “RISE works.” It is:

The mechanism is executable, the small conformance suite passes, and the trust-boundary gate still fails closed.