

Round 25

When Silence Looks Like Stability

Detecting Missing Change Evidence in Collaborative AI Systems

An adversarial omission study for claim-aware observation windows

16 October 2026

Research paper / SACP-0.1-development / D25-A deterministic diagnostic

One bounded authored-fixture diagnostic; no production telemetry, answer benchmark, or deployment.

Contents

This publication separates frozen observations, current-state claims, external findings, and Enceavra hypotheses.

Result boundary	4
Abstract	4
1. Archive continuity and corrected round number	5
2. Research question, novelty, and scope	5
2.1 Primary question	5
2.2 Novelty check against the archive	5
2.3 In scope	6
2.4 Exclusions	6
3. Carry-forward claims and the new evidence boundary	6
4. Concepts that must remain separate	7
5. Threat model	7
6. Literature map and competing positions	8
Position A: silence can be converted into a failure suspicion	8
Position B: append-only commitments and freshness expose rollback, freeze, and some omissions	8
Position C: witnesses and self-monitoring can expose inconsistent views	8
Position D: event envelopes improve interoperability but do not define stream completeness	8
Position E: monitoring systems must monitor their own failures	9
Position F: predictable omissions can be informative	9
7. Claim and evidence table	9
8. Proposed mechanism: Silence-Aware Coverage Protocol	10
8.1 Design objective	10
8.2 Window receipt	10
8.3 Window state machine	10
8.4 Claim status projection	11
8.5 Detection rules	11
8.6 Scope and repair	11
8.7 Privacy and abuse resistance	12
9. Worked example	12
10. Brain-inspired analogy, limited and testable	12
11. Diagnostic D25-A	13
11.1 Purpose	13

11.2 Fixture construction 13

11.3 Policies 13

11.4 Metrics 13

11.5 Results 14

11.6 Interpretation 14

11.7 What D25-A does not establish 14

12. Alternative explanations and simpler designs 14

12.1 Periodic full revalidation may be enough 14

12.2 Better transport may remove most gaps 15

12.3 The authority inventory may be wrong 15

12.4 Independent end-to-end tests may outperform event monitoring 15

12.5 False alerts may be governance failures, not detector failures 15

12.6 Observed gains may come from more total work 15

13. Falsifiable hypotheses 15

14. Proposed experiment E25-B 16

14.1 Objective 16

14.2 Preregistration 16

14.3 Units and window 16

14.4 Hidden adjudication 16

14.5 Arms 16

14.6 Ablations 17

14.7 Controlled total compute and work 17

14.8 Primary metrics 17

14.9 Regression and harm metrics 17

14.10 Decision rule 18

15. Expected failure modes 18

16. Limitations 19

17. Practical recommendation 19

18. Next-round handoff 19

19. Source inspection log 20

20. Sources 20

21. Reproducibility artifacts 21

Closing statement 22

Result boundary

Round 25 asks whether an AI assurance system can distinguish a genuinely quiet authority from a broken, delayed, stale, selectively silent, or common-mode observation path. It proposes the **Silence-Aware Coverage Protocol (SACP)** and executes D25-A, a deterministic conformance diagnostic over 80 authored traces: 16 trace families crossed with the five authority families carried from Round 24. Sixty traces were authored as observation gaps and twenty as negative controls.

On those invented traces, SACP alerted on 55 of 60 gap traces and none of the 20 negative controls. Its trace-level recall was 0.9167 and precision was 1.0000; claim-weighted recall was 0.8944. The five intentional misses were two selective non-emission cases with no independent external commitment and three verifier common-mode failures with no independent challenge path. Revalidate-all achieved 1.0000 recall but alerted on all 20 negative controls and created 400 claim-review tasks, compared with 161 for SACP. A sequence-only policy falsely alerted on all five benign-reordering controls. These numbers show that the implementation conforms to the authored distinctions. They do not estimate production precision, recall, incident prevalence, reliability, cost, or user impact.

No production incidents were sampled. No independent reviewer labelled the traces. No live repository webhook, deployment control plane, database, provider, runtime, user traffic, or production collector was queried or changed. No language model or provider API was called for D25-A. No new Encevera answer benchmark was run. SACP was not installed, and no release gate, retention policy, or deployment behavior changed. The carried current claims remain current only within their previously established evidence scope; this round neither upgrades nor contradicts them.

Abstract

Round 24 represented where change evidence came from and showed why two analyses of one repository do not constitute two independent authorities. That solves a lineage problem only after an observation exists. A harder failure precedes routing: the relevant change may never produce a receipt, may be dropped in transport, may be hidden behind an old checkpoint, or may be verified by components that share the same fault. In those cases, an empty event window can be mistaken for evidence that nothing changed.

Round 25 asks: **Under what observable conditions may Encevera treat a quiet change-evidence window as support for continued claim currency, and when must silence instead become an explicit observation gap?** The proposed SACP combines five ideas: declared observation windows, monotonic sequence and checkpoint commitments, freshness and liveness challenges, authority-lineage-aware reconciliation, and scoped claim-state transitions. A no-change window is admissible only when its observer contract closes cleanly. A gap does not contradict a claim. It changes the claim from `current_with_complete_window` or `current_with_observed_authorities` to `unknown_observation_gap` for the implicated authority until bounded revalidation or gap repair succeeds.

Primary literature and technical standards support the parts, not the Encevera composition. Failure-detector theory separates completeness from accuracy [S05-S06]. Certificate Transparency uses signed tree heads, consistency proofs, monitors, and a Maximum Merge Delay, while acknowledging that consistency across views is harder [S07]. CONIKS and PeerReview add self-monitoring, witnesses, and accountability with explicit trust assumptions [S08-S09]. TUF uses versions and short-lived metadata to resist rollback and freeze attacks [S10]. CloudEvents provides portable event identity, not stream completeness [S11]. NIST AU-5 and OpenTelemetry treat logging and transport failures as events that themselves require monitoring [S12-S13]. A rat auditory-cortex study reports omission responses that scale with learned predictability; Round 25 uses this only as a testable design analogy for expectation-weighted liveness checks [S14].

D25-A shows where each simple policy fails by construction. It also preserves irreducible blind spots: no receipt scheme can prove an event existed when the issuer, observer, verifier, and every external reference all share the same omission. The next valid step is a preregistered, fixed-budget, prospective trial over real

multi-authority windows with hidden independent adjudication. Until that trial, SACP is a falsifiable proposal and D25-A is authored protocol evidence only.

1. Archive continuity and corrected round number

The automation brief named Round 11. Before topic selection, the live Lovable project was inspected at commit `9b1305fcc3b75ce893c6dbc39135ca136330d0a0`. Its Research overview, rounds index, registry, full Round 24 and Round 23 manuscripts, and the relevant Round 10 and Round 9 routes show completed work through Round 24. The correct sequential number is therefore **Round 25**.

The archive review established the following continuity:

Round	Contribution carried forward	Boundary retained here
9	Learn reusable procedures only from verified experience.	D25-A does not create a learned procedure or training signal.
10	Exchange evidence, revise selectively, and treat brain-inspired ideas as hypotheses.	Round 25 studies missing evidence before selective revision and makes no literal brain-copying claim.
21	Bind measurement to attempts and fail closed when the measurement receipt is incomplete.	The production ABMG remains absent and its impact hypotheses remain untested.
22	Separate historical, current, and hypothetical claims; revalidate current claims through typed dependencies.	Historical findings remain frozen; an observation gap changes current status but does not rewrite history.
23	Add a second repository-derived analysis view and test a shadow revalidation selector.	A later event can be routed only if some observer sees it.
24	Preserve root-authority lineage and avoid treating two repository analyses as independent authorities.	ALR describes received-event origin; it does not prove that each required event was emitted or delivered.

Round 24's preferred handoff was E24-B, a preregistered prospective multi-authority window with independent hidden adjudication. The present run had neither independent reviewers nor access to live event feeds from all five authorities. Round 24 explicitly allowed an adversarial omission study as the fallback: replay, selective non-emission, mirrored sources, stale configuration, conflicting issuers, and verifier common-mode failure [S01]. Round 25 takes that bounded fallback rather than fabricating a prospective trial.

2. Research question, novelty, and scope

2.1 Primary question

Under what observable conditions may Enceavra treat a quiet change-evidence window as support for continued claim currency, and when must silence become an explicit observation gap that triggers bounded revalidation?

This is consequential because a collaborative AI system may make a well-formed decision over an incomplete event stream. If an agent sees no provider-contract event, it might preserve a claim that terminal markers are captured. If the provider changed but its watcher failed, the agent's reasoning is internally consistent and externally stale. The failure is neither poor aggregation nor bad selective revision. It is an unobserved change in the evidence-generating process.

2.2 Novelty check against the archive

Three candidate directions were considered after the archive review.

Candidate	Archive overlap	Decision
Confidence calibration across agents	Related to earlier verification and revision work, but not the direct Round 24 handoff.	Deferred.
Fixed-budget verification allocation	Already intertwined with prior gate and matched-cost proposals; requires reliable event inputs first.	Deferred.

Candidate	Archive overlap	Decision
Provenance-preserving memory conflict resolution	Substantially overlaps Rounds 18, 20, 22, and 24.	Rejected for this round.
Observation-channel silence and omission	Directly follows Round 24 yet is not solved by authority lineage, event routing, or procedural learning.	Selected.

The narrow novelty claim is not that heartbeats, sequence numbers, checkpoints, transparency logs, witnesses, canaries, or expiry are new. It is that Encevra needs a claim-aware contract that composes them and refuses to use an empty window as no-change evidence unless the window's completeness conditions are themselves observed.

2.3 In scope

- The five Round 24 authority families: repository, deployment configuration, live database, provider contract, and runtime invocation.
- The five carried current claims from Rounds 21-24.
- Observation-window identity, liveness, sequence continuity, checkpoint freshness, origin lineage, reconciliation, and scoped state transitions.
- Sixteen authored trace families, five authority variants per family, and eight deterministic policies.
- Metrics for gap detection, false suspicion, claim-weighted coverage, detection delay, review load, and useful-current availability.
- A prospective fixed-budget evaluation design with strong baselines and ablations.

2.4 Exclusions

- Product answer quality, calibration, safety, latency, cost, and user trust.
- Production telemetry, deployment state, database state, provider accounts, or user content.
- Proof that an issuer told the truth or emitted every relevant event.
- A claim that signatures, consensus, or transparency logs remove the need for external observation.
- Training models on the authored traces.
- A literal mapping from auditory cortex to software components.

3. Carry-forward claims and the new evidence boundary

The five current negative claims remain the same:

Claim ID	Carried statement
C21-CUR-ENVELOPE	The current measurement path does not commit the exact provider request and response envelope.
C21-CUR-TERMINAL	It does not uniformly persist provider terminal markers or finish and stop reasons.
C21-CUR-ATOMIC	Measurement completeness is not committed atomically with terminal run state.
C21-CUR-GATE	Release is not gated on a complete attempt-bound measurement receipt.
C21-CUR-NO-ABMG	No production attempt-bound measurement gate exists.

Round 24 mapped these claims to authorities and found that the existing evidence directly covered repository state, not all production authorities [S01]. Round 25 adds a second dimension: even an authority with a configured observer may have an incomplete observation window.

The resulting evidence boundary has three levels:

- 1 **Artifact evidence:** what an inspected artifact states.
- 2 **Authority evidence:** which system of record produced or attested to that state.
- 3 **Observation-window evidence:** whether the monitoring path was live, fresh, continuous, and independently checked for the period in which the claim remained current.

An authority name without a complete window is not continued-current evidence. Conversely, a window gap is not evidence that the claim is false. It is evidence that the system no longer knows whether the claim remained true across that authority.

4. Concepts that must remain separate

Concept	Round 25 definition	Invalid shortcut
Event absence	No relevant event was received in a window.	Therefore no relevant change occurred.
No-change statement	An issuer explicitly commits that no covered change occurred.	Therefore the issuer was complete and honest.
Observer liveness	The observer responds to a challenge or advances a heartbeat.	Therefore every domain event was captured.
Sequence continuity	Received event identifiers or commitments form the expected monotonic series.	Therefore the issuer did not omit an event before numbering.
Checkpoint freshness	A signed summary was produced within its declared age bound.	Therefore the summary covers the true authority state.
Consistency	Checkpoints extend a prior commitment without rollback or fork.	Therefore all clients saw the same view.
Witness diversity	Multiple roots with separately modelled failure dependencies attest to the window.	Two services with different names are independent.
Observation gap	At least one required completeness condition is absent, stale, contradictory, or failed.	The underlying claim is contradicted.
Claim revalidation	The claim's named procedure is rerun against current scoped evidence.	Clearing a transport alert restores the claim automatically.

These distinctions prevent a common semantic collapse. Integrity answers, "Was this record altered after commitment?" Completeness asks, "Were all required events committed?" Liveness asks, "Was the observation machinery functioning?" Independence asks, "Could all checks share one failure?" SACP records separate answers.

5. Threat model

Round 25 studies failures at or before receipt creation.

- 1 **Transport loss:** an event is emitted but dropped, delayed, reordered, duplicated, or stranded in a queue.
- 2 **Observer failure:** the watcher crashes, loses access, uses stale configuration, or silently stops advancing.
- 3 **Issuer omission:** the authority changes but the component responsible for creating an event does not emit one.
- 4 **Freeze or rollback:** an old but valid checkpoint is replayed, or a lower epoch is presented as current.
- 5 **Equivocation:** different consumers receive inconsistent signed views for the same authority and window.
- 6 **Mirror inflation:** two observers are counted as independent even though both repeat one upstream feed.
- 7 **Verifier common mode:** event, liveness, and verification checks rely on the same failing component.
- 8 **False suspicion:** a correct but delayed or reordered stream is marked incomplete too early, causing unnecessary withdrawal.

- 9 **Alert fatigue:** a fail-closed policy repeatedly states broad claims and teaches operators to ignore gaps.
- 10 **Adversarial adaptation:** an issuer answers predictable canaries while selectively hiding unchallenged changes.

SACP does not solve collusion among all roots of trust. It records the absence of independent evidence when possible and leaves some omissions formally undetectable under the stated observation model.

6. Literature map and competing positions

Position A: silence can be converted into a failure suspicion

Chandra and Toueg formalize failure detectors as potentially incorrect information about failures and separate completeness from accuracy [S05]. A detector that suspects everyone trivially obtains completeness but is useless without accuracy. The Heartbeat work shows that additional liveness information can support reliable communication despite process crashes and message losses [S06].

Implication: Enceavra must report both missed-gap and false-gap behavior. A policy that withdraws every claim on every quiet period is not a free safety win.

Limit: the theory concerns distributed processes and consensus or communication, not semantic completeness of AI evidence. Round 25 borrows the tradeoff, not the theorem.

Position B: append-only commitments and freshness expose rollback, freeze, and some omissions

RFC 9162 binds entries into signed tree heads, defines a Maximum Merge Delay, requires monitors to inspect every new entry in watched logs, and treats extended entry unavailability as log misbehavior [S07]. TUF uses monotonic metadata versions, consistent snapshots, and short-lived timestamp metadata so clients can detect rollback and indefinite freeze attacks [S10].

Implication: a no-change assertion should be a fresh, signed, monotonic checkpoint for a declared window, not merely an empty query result.

Limit: a log can present split views, and an issuer may omit a real-world event before it enters the committed log. Consistency of recorded entries is not semantic completeness.

Position C: witnesses and self-monitoring can expose inconsistent views

CONIKS lets users monitor their own bindings and collectively audit a provider for non-equivocation [S08]. PeerReview uses signed, hash-chained logs, authenticators, and periodic checks by witnesses to make observed deviations attributable [S09]. PeerReview also states the assumption that each node plus its witness set contains at least one correct node; otherwise completeness can be lost.

Implication: SACP must preserve root lineage and define which independent party can challenge or reconcile each authority.

Limit: multiplying witnesses that share the same upstream source does not create independent coverage. Accountability after observation does not guarantee observation.

Position D: event envelopes improve interoperability but do not define stream completeness

CloudEvents requires an event `id`, `source`, `specversion`, and `type`, and permits consumers to identify duplicates by `source` plus `id` [S11]. This is useful transport structure.

Implication: SACP can extend a standard event envelope rather than replace it.

Limit: the core envelope does not require monotonic sequence numbers, window closure, no-change commitments, witness lineage, or an end-to-end completeness proof.

Position E: monitoring systems must monitor their own failures

NIST AU-5 explicitly calls for alerting and additional action when audit logging fails, including capture and storage failures [S12]. OpenTelemetry recommends queues, retries, persistent storage, and queue metrics, while documenting residual loss from timeout, overflow, crash without persistence, disk failure, queue failure, and misconfiguration [S13].

Implication: collector health and transport state belong inside the assurance model. Telemetry about the target without telemetry about the telemetry path is incomplete.

Limit: operational resilience lowers some loss risks but does not prove that a source emitted every relevant event.

Position F: predictable omissions can be informative

Yaron et al. manipulated tone probabilities in rats and reported neurons whose responses to omissions increased with evidence accumulation and tone predictability [S14]. This supports a limited analogy: silence is informative only relative to an explicit expectation.

Implication: challenge cadence and alert thresholds should be tied to a declared authority-specific hazard or obligation, not a global intuition that "quiet means broken."

Limit: this is neuroscience evidence about rat auditory cortex, not evidence that SACP resembles a brain or improves AI reasoning.

7. Claim and evidence table

Round 25 claim	Evidence	Status	What would change the status
A received-event protocol alone cannot distinguish no change from non-emission.	Logical indistinguishability under identical received traces; literature on failure detection and log monitoring [S05-S07].	Established within the stated observation model.	An additional observable that differs between the two worlds.
Sequence and checkpoint commitments detect some drops, rollback, and freeze.	RFC 9162, TUF, and D25-A authored cases [S07, S10].	Supported as mechanism capability; no production rate claim.	An implementation counterexample or mismatch between declared and actual commitment scope.
Heartbeat liveness does not prove semantic event completeness.	Heartbeat's stated scope plus selective non-emission counterexamples [S06].	Established by scope and construction.	A heartbeat that cryptographically commits the relevant event set and an independently grounded obligation.
Multiple observers can share one omission.	Round 24 lineage audit, PeerReview witness assumption, mirrored D25-A fixtures [S01, S09].	Supported.	Verified independent authority roots and empirically measured joint failure behavior.
SACP dominates simpler policies in production.	None. D25-A is authored.	Open Encevera hypothesis.	E25-B or later independently adjudicated prospective evidence.
SACP's authored blind spots are irreducible.	They are irreducible only under the declared absence of any independent commitment or challenge.	Conditional analytical claim.	Adding a new independent observable can make them detectable.
Expectation-weighted canaries reduce false alarms.	Neuroscience analogy and protocol rationale only [S14].	Open Encevera hypothesis.	Controlled ablation with fixed monitoring budget.

8. Proposed mechanism: Silence-Aware Coverage Protocol

8.1 Design objective

SACP turns a quiet period into one of two explicit statements:

- **closed no-change window:** the observer contract remained live, fresh, continuous, reconciled, and within policy for the full window;
- **open observation gap:** at least one required condition failed or could not be established.

The first is bounded evidence for continued currency within the declared authority and time interval. The second is uncertainty, not contradiction.

8.2 Window receipt

Each authority emits or is represented by an append-only window receipt with these fields:

Field	Purpose
<code>authority_id</code> and <code>authority_family</code>	Name the system of record and the policy family it belongs to.
<code>observer_id</code> , <code>issuer_id</code> , <code>verifier_id</code>	Separate who watched, who stated, and who checked.
<code>root_lineage[]</code>	Preserve upstream dependencies so mirrors do not inflate diversity.
<code>epoch_id</code> , <code>window_open</code> , <code>window_close</code>	Bind the observation interval and reset semantics.
<code>first_sequence</code> , <code>last_sequence</code> , <code>event_count</code>	Expose interior or tail gaps when an external commitment exists.
<code>previous_checkpoint_digest</code> , <code>checkpoint_digest</code>	Make rollback and fork evidence portable.
<code>coverage_contract_version</code>	State which event types, namespaces, and exclusions the receipt covers.
<code>max_silence</code> , <code>close_deadline</code> , <code>clock_basis</code>	Define when quiet becomes overdue rather than relying on an unstated timeout.
<code>challenge_id</code> , <code>challenge_due</code> , <code>challenge_result</code>	Test the observation path independently of natural changes.
<code>transport_ack</code> , <code>queue_state</code> , <code>delivery_deadline</code>	Distinguish producer, transport, and consumer failure where possible.
<code>reconciliation_refs[]</code>	Bind external checkpoints or witnesses with distinct roots.
<code>status</code> and <code>reason_codes[]</code>	Record closed, open-gap, conflicting, stale, or retired state.
<code>privacy_class</code> , <code>content_commitments[]</code>	Permit integrity checks without exposing raw prompts, secrets, or user data.
<code>signature</code> and <code>algorithm</code>	Authenticate the issuer's statement without implying truth.

SACP can wrap the identity fields of CloudEvents [S11], but the completeness fields are an application-level extension.

8.3 Window state machine

- 1 **OPEN:** the window is accepting events.
- 2 **CLOSING:** natural events stop at the declared boundary; the observer waits through the maturity allowance for reordering and retry.
- 3 **CLOSED_COMPLETE:** sequence, checkpoint, freshness, challenge, transport, lineage, and reconciliation conditions pass.
- 4 **OPEN_GAP:** at least one required condition fails, expires, conflicts, or lacks evidence.
- 5 **REPAIRED:** missing data or an independently verified checkpoint closes the gap without rerunning the claim procedure.
- 6 **REVALIDATED:** the affected claim procedure has run against current evidence and produced a new receipt.

- 7 **RETIRED:** the authority or coverage contract is no longer relied upon.

The maturity allowance matters. A transient sequence hole inside the allowed reorder window remains **CLOSING**, not **OPEN_GAP**. This is how SACP avoids the sequence-only false alerts in D25-A.

8.4 Claim status projection

SACP does not invent one global "valid" bit. It projects window status into each claim's authority predicate.

Claim state	Meaning
historical_frozen	A past result remains bound to its original artifacts and time.
current_with_complete_window	Every required authority has a complete window through the claim's declared horizon.
current_with_observed_authorities	The claim is current only within explicitly named observed authorities; unobserved scope is excluded from the wording.
unknown_observation_gap	A required authority window is missing, stale, contradictory, or incomplete.
stale_needs_revalidation	A received change or unresolved gap triggers the claim's named revalidation procedure.
contradicted	Revalidation found the claim false.
retired	The claim is no longer maintained.

An observation gap first creates **unknown_observation_gap**. Policy can escalate to **stale_needs_revalidation** based on risk, gap duration, and available budget. Neither status rewrites a historical result.

8.5 Detection rules

For each required authority window, SACP evaluates six rule families:

- 1 **Continuity:** persistent sequence gaps, count mismatches, tail commitments, and duplicate identity.
- 2 **Freshness:** checkpoint age, monotonic epoch, rollback, freeze, and close deadline.
- 3 **Liveness:** challenge canary, observer heartbeat, and collector self-telemetry.
- 4 **Transport:** acknowledgment chain, queue capacity, retry exhaustion, and durable-spool status.
- 5 **Lineage:** observer root graph, mirror detection, and correlated dependency labels.
- 6 **Reconciliation:** consistency with separately rooted checkpoints, witness views, or external obligations.

A rule may be **pass**, **fail**, **not_applicable**, or **not_observed**. **Not_observed** is not converted to **pass**.

8.6 Scope and repair

Each rule failure maps to the authority and claims that depend on it. A repository checkpoint gap need not stale a provider-only claim. A provider-contract gap need not rewrite a historical database finding. Unknown cross-authority failures may conservatively scope to every claim that names the shared root.

Repair can take three forms:

- **gap repair:** retrieve and verify missing events or a consistency proof;
- **window replacement:** obtain a separately rooted checkpoint that covers the same interval;
- **claim revalidation:** rerun the named procedure when the missing history cannot be reconstructed.

Automatic restoration is permitted only when the receipt specifies which repair rule was satisfied and the resulting evidence horizon.

8.7 Privacy and abuse resistance

Public receipts should contain commitments and coverage metadata, not raw prompts, provider responses, secrets, customer identifiers, or unrestricted database rows. Challenge canaries should be synthetic, non-user-bearing, scoped to the observer contract, and rate limited. Their schedule should be unpredictable to an untrusted observer when selective answering is in the threat model. Public reason codes should distinguish `transport_gap`, `checkpoint_stale`, `fork_detected`, `lineage_common_origin`, and `verifier_unobserved` without exposing sensitive content.

9. Worked example

Assume claim `C21-CUR-TERMINAL` depends on the provider contract and runtime invocation authorities. The provider observer closes epoch 184 with sequence range 7,201-7,260, event count 60, checkpoint digest `P184`, and a successful challenge. The runtime observer receives through 7,259, reports a healthy heartbeat, and emits no error.

A receipts-only policy sees no failure notice and preserves the claim. Heartbeat-only also passes. Sequence-only cannot know that 7,260 exists because it sees no later event. SACP compares the provider checkpoint's committed high-water mark with the runtime acknowledgment and opens `runtime_invocation::tail_commitment_missing`. The claim moves to `unknown_observation_gap` for the runtime authority. Other claims that do not depend on runtime invocation remain unchanged.

If event 7,260 arrives before the maturity deadline and verifies under `P184`, the window closes and no claim revalidation is required. If the event cannot be recovered, the named terminal-marker procedure runs against a fresh provider and runtime sample. A failed revalidation can contradict the claim; the observation gap alone cannot.

Now change the example: the provider silently omits an actual contract change before assigning any sequence number, and every observer reads only the provider's feed. The provider emits a fresh, consistent checkpoint and answers a generic heartbeat. SACP cannot detect the omission because both worlds generate the same observable receipt. Only an external obligation, independent probe, customer-side observation, or separately rooted contract digest can separate them. This is an explicit design limit, not an implementation bug.

10. Brain-inspired analogy, limited and testable

The useful analogy is **expectation-weighted omission**, not brain emulation. In the rat auditory study, omission responses were related to the predictability of the missing tone and increased as the predictive context accumulated [S14]. SACP converts that idea into a software hypothesis:

The absence of a signal should trigger stronger scrutiny when the system has a stronger, independently specified expectation that the signal or checkpoint should occur.

The testable implementation is simple. Each authority contract declares an expected cadence or event hazard, a maximum silence, and a challenge budget. High-obligation windows receive more frequent or less predictable challenges; low-obligation windows use longer maturity periods. The ablation is SACP with uniform cadence versus expectation-weighted cadence at the same total challenge count.

This analogy is rejected if uniform cadence matches or outperforms expectation weighting on gap recall, false suspicion, and detection delay at fixed budget. Nothing in this proposal claims that software authorities are neurons, that the brain uses signed checkpoints, or that neuroscience validates SACP.

11. Diagnostic D25-A

11.1 Purpose

D25-A asks whether the proposed policy implementation reproduces the distinctions written into its specification and exposes known blind spots. It is a conformance diagnostic, not an effectiveness study.

11.2 Fixture construction

Sixteen trace families were crossed with five authority families, producing 80 traces. Each trace names its truth label, visible features, earliest detectable step, and affected claims.

Trace group	Families	Gap truth
Negative controls	honest no-change, normal delivered change, duplicate delivery, benign reordering within the maturity window	No
Loss and delay	dropped middle, dropped tail, delivery beyond deadline, transport queue overflow	Yes
Liveness and freshness	observer crash, stale checkpoint freeze, rollback checkpoint, challenge suppression	Yes
Adversarial source/view	selective issuer non-emission, conflicting issuer views, mirrored common origin, verifier common-mode failure	Yes

The five selective non-emission fixtures have an external commitment only for repository, deployment configuration, and provider contract. The five verifier common-mode fixtures have an independently rooted verifier canary only for repository and provider contract. This intentionally leaves five SACP-undetectable cases.

11.3 Policies

- 1 Receipts only.
- 2 Periodic expiry of all claims.
- 3 Heartbeat only.
- 4 Sequence only, with no maturity allowance.
- 5 Checkpoint plus sequence.
- 6 Naive multi-observer checking without origin-lineage validation.
- 7 SACP.
- 8 Revalidate all on every window.

The authored `declared_logical_checks_per_trace` value is a complexity annotation, not measured CPU, token, latency, or dollar cost.

11.4 Metrics

- trace-level true positives, false positives, false negatives, recall, and precision;
- claim-weighted gap recall;
- false claim withdrawals on negative controls;
- useful-current availability on negative controls;
- review tasks opened;
- mean detection step among detected positive traces;
- exact miss and false-alert lists.

11.5 Results

Policy	TP / 60	FP / 20	Gap recall	Precision	Claim-weighted recall	Useful-current availability	Review tasks
Receipts only	5	0	0.0833	1.0000	0.0833	1.0000	15
Periodic expiry all	60	20	1.0000	0.7500	1.0000	0.0000	400
Heartbeat only	15	0	0.2500	1.0000	0.2500	1.0000	45
Sequence only	10	5	0.1667	0.6667	0.1667	0.7500	45
Checkpoint + sequence	40	0	0.6667	1.0000	0.6667	1.0000	120
Naive multi-observer	48	0	0.8000	1.0000	0.7889	1.0000	142
SACP	55	0	0.9167	1.0000	0.8944	1.0000	161
Revalidate all	60	20	1.0000	0.7500	1.0000	0.0000	400

All eight executable conformance checks passed. The generated manifest binds the script, fixture set, results, and check file by SHA-256.

11.6 Interpretation

The meaningful finding is structural, not comparative performance. Different observables detect different omission classes. Explicit failure receipts expose queue overflow but almost nothing else. Heartbeats detect liveness loss but not a semantically incomplete live observer. Sequences expose interior gaps but cannot reveal an omitted tail without a checkpoint and falsely suspect benign reordering when no maturity window exists. Checkpoints add freeze, rollback, tail, and equivocation coverage. Cross-source checks add selective non-emission only when an external commitment exists. Origin lineage adds the mirrored-source case. Independent verifier challenges add only the common-mode cases for which they are actually independent.

SACP's five misses are the most important result. They prevent the authored suite from becoming a circular demonstration of perfection. Any published claim that SACP guarantees complete observation would be false under its own fixtures.

11.7 What D25-A does not establish

- The frequencies of these failures in Enceavra or any other system.
- The probabilities of joint observer failure.
- Production precision, recall, detection delay, or cost.
- That the chosen trace taxonomy is complete.
- That challenge canaries are secure against an adaptive issuer.
- That users prefer temporary uncertainty to stale claims.
- That SACP improves answer correctness.
- That six logical checks are affordable or sufficient in production.

12. Alternative explanations and simpler designs

12.1 Periodic full revalidation may be enough

If claims are cheap to rerun and changes are rare, periodic revalidation can dominate a complex observation protocol. D25-A shows its constructed safety/availability tradeoff but does not price real work. E25-B must include periodic and revalidate-all baselines under the same total budget.

12.2 Better transport may remove most gaps

Durable queues, retries, and replay can reduce loss [S13]. If transport loss accounts for nearly all real failures, SACP's lineage and witness machinery may add little. The experiment must record failure class, not only total recall.

12.3 The authority inventory may be wrong

A missing authority cannot be protected by a perfect window protocol. Round 24's five families are an authored inventory, not a proof of completeness. Unknown authority discovery remains a separate problem.

12.4 Independent end-to-end tests may outperform event monitoring

Synthetic transactions or direct user-side probes can observe behavior without trusting internal event emitters. They may be stronger than source receipts for some claims, though weaker for localization. E25-B should include an end-to-end probe baseline where privacy and safety allow it.

12.5 False alerts may be governance failures, not detector failures

If teams set unrealistic deadlines or poorly scoped coverage contracts, a technically correct alert can still be operationally harmful. Deadline calibration and ownership are part of the intervention.

12.6 Observed gains may come from more total work

SACP has more checks than simple baselines. Any future comparison without equal total monitoring, storage, challenge, and revalidation budgets would confound protocol structure with resource volume.

13. Falsifiable hypotheses

All hypotheses below are proposed and untested outside authored D25-A.

H25-1 - Gap recall: At fixed total monitoring and revalidation budget, SACP reduces independently adjudicated affected-claim gap misses versus receipts-only, heartbeat-only, checkpoint-sequence, and periodic-expiry baselines.

H25-2 - Availability: SACP preserves more correctly current claim-hours than revalidate-all while staying within a preregistered non-inferiority margin for affected-claim gap recall.

H25-3 - Lineage: Removing root-lineage checks increases misses for mirrored/common-origin failures without reducing detection of independently rooted conflicts.

H25-4 - Maturity window: Removing the reorder maturity window increases false observation gaps on delayed but complete streams.

H25-5 - Challenges: Removing challenge canaries increases detection delay for observer crashes and stale checkpoint freezes.

H25-6 - Independent reconciliation: Removing separately rooted reconciliation increases misses for selective non-emission and equivocation.

H25-7 - Expectation weighting: At the same total challenge count, authority-specific expectation-weighted scheduling improves gap recall or detection delay without increasing false suspicion versus uniform scheduling.

H25-8 - Scoped state: Claim-scoped gap projection reduces unnecessary claim withdrawal versus authority-wide or registry-wide fail-closed policies without increasing stale-claim escape.

H25-9 - User-facing honesty: Explicit `unknown_observation_gap` status reduces unsupported current claims without causing more abandonment or misunderstanding than a generic stale label in a blinded review.

H25-10 - Complexity: A simpler checkpoint-sequence-plus-end-to-end-probe policy matches SACP on safety, availability, and cost. Support for this hypothesis would reject the need for full SACP.

14. Proposed experiment E25-B

14.1 Objective

Test whether SACP improves gap detection and claim currency in a prospective multi-authority window, using real operational event paths, independent hidden adjudication, and controlled total resources.

14.2 Preregistration

Before the observation window begins, freeze:

- claim texts, authority dependencies, and revalidation procedures;
- observer and root-lineage graph;
- coverage contracts, deadlines, reorder allowances, and challenge schedule generator;
- trace inclusion/exclusion rules;
- adjudication handbook and hidden-label protocol;
- total monitoring, storage, challenge, and revalidation budgets;
- primary, secondary, harm, and regression metrics;
- non-inferiority and superiority margins;
- stopping rules and missing-data handling;
- analysis code hash.

14.3 Units and window

The unit is an authority-window-by-claim edge, not an individual log line. Include at least repository, deployment configuration, and runtime invocation. Add live database and provider contract only if independent observation and privacy review are available. Run long enough to capture multiple genuine quiet windows, normal changes, delayed deliveries, observer restarts, and maintenance events. Do not inject production faults without explicit authorization and safety review.

Natural incidents alone may be too sparse. Therefore use two strata:

- 1 **prospective natural stratum:** real windows and naturally occurring operational events;
- 2 **sandbox perturbation stratum:** preregistered, non-production fault injections such as dropped messages, delayed tail events, frozen checkpoints, mirrored observers, and verifier dependency failure.

Results must be reported separately before any pooled analysis.

14.4 Hidden adjudication

An independent adjudication team receives authority snapshots and end-to-end evidence that the evaluated policies cannot access. Two reviewers label whether a required change or observation gap occurred, which claims were affected, and the earliest defensible detection time. Disagreements are resolved by a third reviewer. Policies remain blind to adjudication labels until the window closes.

14.5 Arms

- B0: receipts only.
- B1: periodic expiry.

- B2: heartbeat only.
- B3: checkpoint plus sequence.
- B4: end-to-end synthetic probe.
- B5: naive multi-observer without root lineage.
- B6: full SACP.
- B7: revalidate all.

14.6 Ablations

- A1: SACP minus maturity window.
- A2: SACP minus challenge canaries.
- A3: SACP minus root-lineage checks.
- A4: SACP minus independent reconciliation.
- A5: SACP with uniform rather than expectation-weighted challenge cadence.
- A6: SACP with authority-wide rather than claim-scoped state transitions.

14.7 Controlled total compute and work

Each arm receives the same total budget envelope over the evaluation window:

- collector CPU-seconds and storage bytes;
- network requests and bytes;
- challenge count;
- cryptographic verification operations;
- human review minutes;
- claim revalidation executions;
- any model tokens or provider calls, if later introduced.

Unused budget cannot silently roll into more revalidation for one arm. If an arm's mechanism intrinsically costs more, run cost-frontier analyses at multiple fixed envelopes. Report both safety at matched budget and budget required to reach a matched safety target.

14.8 Primary metrics

- 1 Affected-claim gap recall.
- 2 Affected-claim gap precision.
- 3 Stale-claim escape hours.
- 4 Correctly current claim-hours.
- 5 Time from first independently observable signal to alert.
- 6 Time from alert to repaired or revalidated state.
- 7 Review minutes and revalidation runs per true gap found.

14.9 Regression and harm metrics

- false withdrawal hours for claims later adjudicated current;
- alert volume, repeated-alert rate, and operator acknowledgment latency;

- unresolved-gap age and silent auto-restoration count;
- claim scope inflation, where one authority gap withdraws unrelated claims;
- false independence count from shared lineage;
- privacy leakage in receipts and challenges;
- collector-induced load, queue pressure, and dropped production traffic;
- challenge predictability and adaptive pass/fail asymmetry;
- clock-skew sensitivity and false deadline violations;
- discrepancy between policy-visible and adjudicator-visible authority roots;
- user-facing misunderstanding of `unknown` versus `false`;
- answer-quality regression if claim status is surfaced to downstream agents.

14.10 Decision rule

Advance SACP only if it improves affected-claim gap recall or stale-claim escape over checkpoint-sequence at matched budget, stays within preregistered limits for false withdrawal and review burden, and adds unique catches attributable to lineage, challenges, or reconciliation. Reject or simplify SACP if a checkpoint-sequence-plus-probe baseline matches it, if gains vanish under matched work, or if operational harm exceeds the safety benefit.

15. Expected failure modes

- 1 **Contract drift:** the coverage contract no longer matches the authority's real event surface.
- 2 **Clock ambiguity:** issuer and verifier disagree about window boundaries or freshness.
- 3 **Sparse obligations:** an authority has no independently predictable event or checkpoint cadence.
- 4 **Adaptive canary gaming:** an adversary recognizes challenges and answers only those.
- 5 **Root misclassification:** supposedly independent witnesses share credentials, code, vendor, queue, or control plane.
- 6 **Checkpoint laundering:** a valid signature authenticates an incomplete or misleading scope.
- 7 **Gap storms:** a shared dependency failure opens many claim gaps at once.
- 8 **Repair ambiguity:** retrieved events close transport continuity but do not prove issuer completeness.
- 9 **Automatic restoration error:** a recovered heartbeat restores a claim whose event history remains incomplete.
- 10 **Retention mismatch:** required proofs expire before the claim's currency horizon.
- 11 **Privacy spill:** overly detailed receipts reveal provider payloads, secrets, or user activity.
- 12 **Metric gaming:** teams lengthen deadlines to reduce false alerts while increasing stale-claim escape.

16. Limitations

The diagnostic taxonomy and expected outputs were authored by the same research process that proposed SACP. This makes D25-A useful for specification testing and weak for external validation. The authority-to-claim edges are carried from Round 24 and may be incomplete. The fixed earliest-signal steps are ordinal fixture annotations, not measured time. Review-task counts assume one task per affected claim for scoped policies and five for registry-wide policies; they are not labor estimates. No independence probabilities were measured. No cryptographic implementation, clock model, adversarial network, or privacy audit was exercised.

The primary literature is adjacent rather than directly validating. Consensus failure detectors, certificate and key transparency, software-update metadata, distributed accountability, logging controls, and auditory omission responses address different systems. They justify decomposing liveness, completeness, freshness, consistency, and expectation. They do not show that SACP improves collaborative AI reasoning.

The archive was inspected through one live commit. A later unobserved project mutation could change numbering or conclusions after that point. The paper and bundle bind the inspected commit so the continuity claim remains auditable.

17. Practical recommendation

Do not treat an empty event query as evidence that nothing changed. For any current claim that depends on an external or mutable authority, record the authority, observation window, coverage contract, freshness deadline, lineage, and whether the window closed complete. If that evidence is unavailable, narrow the claim wording or expose `unknown_observation_gap` rather than silently preserving global currency.

Do not deploy full SACP from D25-A. A low-risk next implementation is a shadow receipt schema for one non-sensitive repository authority and one separately rooted deployment authority, with no release blocking. Measure false gaps, delayed closure, and unique catches before expanding.

18. Next-round handoff

Round 25 completes the Round 10 roadmap block on dependency discovery, contradictions, source authority, local repair, and retention only at the design-evidence level. The immediate next round should not author another larger fixture suite.

Preferred Round 26:

- 1 preregister a small E25-B shadow window;
- 2 freeze two or three current claims and at least three authority observers;
- 3 establish independent hidden adjudication;
- 4 allocate equal monitoring and revalidation budgets across checkpoint-sequence, end-to-end probe, and SACP arms;
- 5 report natural and sandbox strata separately;
- 6 preserve every false alert, missed gap, and restoration decision.

If independent adjudication or real multi-authority access remains unavailable, Round 26 should instead perform a formal indistinguishability analysis: enumerate which omission classes cannot be detected under each observable set and prove the minimum new trust assumption required to separate them. It should not claim another authored conformance improvement as evidence of production effectiveness.

19. Source inspection log

ID	Publication date	Material directly inspected	What it supports here
S01	2026-10-15	Full Round 24 manuscript, diagnostic, limitations, and handoff at the bound archive commit.	Authority families, claim set, common-origin result, and omission-study fallback.
S02	2026-10-14	Full Round 23 manuscript, selector design, result boundary, and handoff.	Received-event routing does not prove event observation.
S03	2026-10-01	Round 10 question, conclusions, and roadmap.	Program continuity and falsifiable brain-inspired framing.
S04	2026-09-30	Round 9 question, findings, and caveats.	Procedural learning is out of scope.
S05	1996-03	Abstract and Sections 2.2-2.4.	Completeness and accuracy are separate detector properties.
S06	1997-05	Cornell record and abstract.	Liveness information can support reliable communication over lossy links.
S07	2021-12	RFC Sections 4.1, 4.11, 8.2, and security considerations.	Fresh checkpoints, consistency proofs, monitor duties, and split-view limits.
S08	2015-08	USENIX paper abstract and monitoring/auditing design.	Self-monitoring and collective consistency checks.
S09	2007-10	Abstract, assumptions, Sections 4.4 and 4.11.	Tamper-evident logs, witnesses, accountability, and explicit completeness assumptions.
S10	Current pages	TUF security and metadata pages.	Rollback/freeze threat model, versions, snapshot commitments, expiry.
S11	v1.0 milestone 2019-10-28	Core required attributes and duplicate semantics.	Event identity envelope, not completeness.
S12	2020-09, Rev. 5.1 text	AU-5 control and discussion.	Logging-process failure needs explicit alert and response.
S13	Page updated 2026-01-14	WAL, queues, loss scenarios, and recommendations.	Transport resilience and residual loss modes.
S14	2025-06-18	Abstract, experimental conditions, and evidence-accumulation analysis.	Limited expectation-weighted omission analogy.

The complete machine-readable source ledger records working links, inspection notes, support boundaries, and non-support boundaries.

20. Sources

[S01] Encevera Research. "Round 24: Do Two Watchers Count as Two? Authority Lineage and Common-Origin Failure in Claim Revalidation." 15 October 2026. [src/content/research-round-24.md](#), archive commit [9b1305fcc3b75ce893c6dbc39135ca136330d0a0](#).

[S02] Encevera Research. "Round 23: Can the Gate Catch Tomorrow's Change? A Prospective Shadow Trial of Claim Revalidation." 14 October 2026. [src/content/research-round-23.md](#), same archive commit.

[S03] Encevera Research. "Round 10: Evidence exchange, selective revision, and a testable path toward brain-inspired intelligence." 1 October 2026. [src/routes/research-round-10.tsx](#), same archive commit.

[S04] Encevera Research. "Round 9: Learning reusable procedures from verified experience." 30 September 2026. [src/routes/research-round-9.tsx](#), same archive commit.

[S05] Chandra, Tushar Deepak, and Sam Toueg. "Unreliable Failure Detectors for Reliable Distributed Systems." *Journal of the ACM* 43(2): 225-267. March 1996. DOI: <https://doi.org/10.1145/226643.226647>. Open copy inspected: <https://www.cs.cornell.edu/courses/cs734/2000FA/cached%20papers/ct96.pdf>.

[S06] Aguilera, Marcos K., Wei Chen, and Sam Toueg. "Heartbeat: A Timeout-Free Failure Detector for Quiescent Reliable Communication." Cornell Technical Report 97-1631. May 1997.
<https://ecommons.cornell.edu/entities/publication/365af68a-24d0-4b7e-a12e-9d511a389545>.

[S07] Laurie, B., E. Messeri, and R. Stradling. "Certificate Transparency Version 2.0." RFC 9162. December 2021. <https://www.rfc-editor.org/rfc/rfc9162.html>.

[S08] Melara, Marcela S., Aaron Blankstein, Joseph Bonneau, Edward W. Felten, and Michael J. Freedman. "CONIKS: Bringing Key Transparency to End Users." 24th USENIX Security Symposium, August 2015.
<https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/melara>. Paper:
<https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-melara.pdf>.

[S09] Haeberlen, Andreas, Petr Kouznetsov, and Peter Druschel. "PeerReview: Practical Accountability for Distributed Systems." SOSP 2007. DOI: <https://doi.org/10.1145/1294261.1294279>. Expanded report inspected: <https://www.mpi-sws.org/tr/2007-003.pdf>.

[S10] The Update Framework. "Security" and "Roles and metadata." Current pages inspected 16 October 2026. <https://theupdateframework.io/docs/security/> and <https://theupdateframework.io/docs/metadata/>.

[S11] Cloud Native Computing Foundation. "CloudEvents Core Specification." Version 1.0 milestone announced 28 October 2019; current core text inspected 16 October 2026.
<https://github.com/cloudevents/spec/blob/main/cloudevents/spec.md>.

[S12] Joint Task Force. Security and Privacy Controls for Information Systems and Organizations. NIST SP 800-53 Rev. 5, September 2020; AU-5 in derived Rev. 5.1 text inspected.
<https://doi.org/10.6028/NIST.SP.800-53r5>.

[S13] OpenTelemetry. "Collector resiliency." Page last modified 14 January 2026.
<https://opentelemetry.io/docs/collector/resiliency/>.

[S14] Yaron, Amit, Tomoyo Shiramatsu-Isoguchi, Felix B. Kern, Kenichi Ohki, Hirokazu Takahashi, and Zenas C. Chao. "Auditory cortex neurons that encode negative prediction errors respond to omissions of sounds in a predictable sequence." PLOS Biology 23(6): e3003242. Published 18 June 2025.
<https://doi.org/10.1371/journal.pbio.3003242>.

21. Reproducibility artifacts

The Round 25 bundle contains:

- the complete manuscript source;
- deterministic diagnostic source;
- all 80 authored trace fixtures;
- complete per-policy results, misses, and false-alert lists;
- eight executable conformance checks;
- a source ledger with inspection and support boundaries;
- SHA-256 commitments for generated artifacts.

Run `python3 silence_coverage_diagnostic.py` inside the bundle to regenerate the fixture and result files. The program fails if any named conformance assertion does not hold.

Closing statement

A quiet channel is not automatically a stable world. It becomes bounded no-change evidence only when the system can show that the observer was alive, the window was fresh and continuous, the checkpoint was consistent, the transport did not silently fail, and nominally separate checks did not share one unrecorded root. Even then, issuer-level non-emission can remain invisible. Round 25 therefore adds a new state to collaborative assurance: not true, not false, but **unknown because the observation window did not close**.