

Round 23

Can the Gate Catch Tomorrow's Change?

A Prospective Shadow Trial of Claim Revalidation

A dual-view dependency gate, one real post-freeze commit interval, and a fixed-budget prospective protocol

14 October 2026

Research paper / CTAG-0.1-shadow / D23-A deterministic diagnostic

One bounded diagnostic; no production gate, provider experiment, 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	5
3. Carry-forward from prior rounds	5
4. Scope and exclusions	6
4.1 In scope	6
4.2 Out of scope	6
5. Literature map and competing positions	7
Position A: authored exact dependencies are the clearest control	7
Position B: derived program dependencies can recover omitted relations	7
Position C: fail closed on unknown production paths	7
Position D: retest everything is the simplest high-recall baseline	7
Position E: assurance should evolve continuously	7
Position F: mutation suites are valuable but cannot stand in for real incidents	8
6. Proposed mechanism: the CTAG Shadow Gate	8
6.1 Claim cards	8
6.2 Change-event cards	8
6.3 Two dependency views	9
6.4 Fail-closed unknown rule	9
6.5 State transition and revalidation receipt	9
7. Diagnostic D23-A: design	9
7.1 Temporal split	9
7.2 Selectors	10
7.3 Gold labels	10
7.4 Metrics	10
8. D23-A results	11
8.1 Real post-freeze change interval	11
8.2 Claim selection and availability	11
8.3 Revalidation findings	12
8.4 Process observations	12
9. Controlled mutation suite	12

10. Worked example: the provider gateway catch

11. Claim/evidence table

12. Alternative explanations

12.1 The unique catch may reflect an incomplete manifest, not the superiority of import analysis

12.2 The same researcher may have favored the frozen hypothesis

12.3 Five claims make saturation likely

12.4 The commit was unusually favorable to imports

12.5 Negative claims may be easier to revalidate

12.6 Retest-all may be competitive at this scale

13. Falsifiable hypotheses

14. Proposed experiment E23-B

14.1 Objective

14.2 Prospective freeze

14.3 Units and sample

14.4 Independent adjudication

14.5 Baselines and ablations

14.6 Controlled total verification budget

14.7 Primary outcomes

14.8 Harm and regression metrics

14.9 Analysis

15. Brain-inspired hypothesis, kept testable

16. Expected failure modes and safeguards

17. Limitations

18. Architecture and product decision

19. Next-round handoff

20. Source inspection ledger

21. Full sources

22. Reproducibility manifest

23. Final claim boundary

13

14

14

14

14

14

14

14

15

15

15

15

15

16

16

16

16

16

16

16

16

17

17

17

18

18

18

19

19

20

20

21

22

Result boundary

This round evaluated a shadow claim-revalidation gate on one real Enceavra commit interval that occurred after the Round 22 dependency manifest was frozen. The diagnostic, D23-A, made a path-only prediction before inspecting diff content, then compared five selection strategies against post-hoc semantic labels. It also ran eight authored mutation fixtures as conformance checks. The hybrid selector found all three artifacts labeled relevant in the real interval, including one artifact uniquely added by an independently derived import view. It also selected four irrelevant artifacts and invalidated two claims that adjudication later found unaffected. These observations are useful design evidence, not a general validation result.

The post-hoc labels were produced by the same research process that froze the prediction. The temporal separation is real, but the adjudication was not independently blinded. Only one real commit interval and five current claims were studied. The mutation cases were authored, not sampled production incidents. No model-quality, answer-reliability, security, latency, cost, provider, or production-database outcome was measured. No production Claim-to-Artifact Graph (CTAG) or Attempt-Bound Measurement Gate (ABMG) implementation was added. No new Enceavra model or answer benchmark was run.

Abstract

Round 22 specified how Enceavra should distinguish frozen historical findings, current claims, and untested hypotheses. It also left a consequential question unanswered: can a prospective gate notice when a genuinely later software change should make a current claim stale, without unnecessarily withdrawing unrelated claims or rewriting history?

Round 23 turns that question into a shadow diagnostic. Before inspecting the content of the first post-freeze diff, D23-A froze a path-only prediction derived from Round 22's manifest. It then constructed an independent static-import view, applied a fail-closed rule to unknown production-looking paths, and compared five selectors: a narrow measurement watchlist, the frozen exact-path manifest, filename keywords, static import closure, and retest-all. A hybrid CTAG selector combined authored dependencies, import closure, and the unknown-path rule.

Across ten changed artifacts, post-hoc semantic inspection labeled three as relevant to at least one of five current claims. The narrow watchlist selected none. The exact-path manifest selected three artifacts and caught two relevant ones. Static import closure and the hybrid each selected seven artifacts and caught all three. Retest-all caught all three but selected all ten. The import view's unique catch was `src/lib/ai-gateway.server.ts`, whose change altered the serialized provider request but whose path was absent from the frozen exact list. At the claim level, the hybrid made all five current claims stale; adjudication associated the change with three, so claim recall was 1.0 and precision was 0.6 in this single interval. Revalidation found all five negative claims still true at the candidate commit. Two historical claims remained frozen, and two ABMG hypotheses remained untested.

Eight controlled mutation fixtures tested known failure modes such as a new provider helper, an ignored terminal marker, reordered finalization, a swallowed measurement-write failure, and an unknown runtime adapter. The hybrid matched all eight expected claim sets. That result shows internal conformance to the authored policy, not real-world discovery performance.

The main conclusion is deliberately narrow. A second dependency view added one unique real catch and eliminated stale-claim escape in this interval, but it increased review load and did not improve claim-level selectivity because the small five-claim scope saturated. The design is promising enough for a multi-commit, independently adjudicated shadow trial. It is not ready to block release, certify evidence, or support a claim that Enceavra answers are more reliable.

1. Archive continuity and corrected round number

The requested brief referred to Round 11. The live Enceavra archive was inspected before topic selection. It already contained completed Rounds 11 through 22, with Round 22 as the latest completed page. The correct sequential number is therefore **Round 23**.

The continuity review included the Research overview, rounds index, publications list, full texts of Rounds 22, 21, 10, and 9, and earlier rounds linked by the candidate question. Round 10 established a staged research program for evidence exchange and selective revision. Round 9 studied learning reusable procedures from verified experience. Those topics are not repeated here. Rounds 19 through 22 progressively established an evidence lifecycle: evidence delivery, minimal-disclosure receipts, fail-closed attempt-bound measurement, and dependency-triggered claim revalidation. Round 22 explicitly handed off a prospective shadow test of its CTAG proposal [S01, S02].

Round 23 is therefore neither a general discussion of evidence exchange nor a new proposal for procedural learning. Its distinct question is whether the prior round's current-claim expiry mechanism can catch a future change under a bounded verification budget.

2. Research question

Primary question:

Can a frozen claim-to-artifact graph, augmented by an independently derived static-import view and a fail-closed unknown-path rule, identify which Round 21 current claims require revalidation after genuinely later Enceavra changes, while preserving historical findings and limiting unnecessary invalidation?

The question is consequential because a current claim can fail in two directions:

- 1 Stale-claim escape:** a relevant change occurs, but the public claim remains marked current without revalidation.
- 2 Over-invalidatio**n: an unrelated change withdraws useful claims, creates avoidable review work, or trains maintainers to ignore a noisy gate.

The question is falsifiable. A change-to-claim selector fails its primary safety objective if an independently adjudicated relevant change leaves an affected current claim unstaled. It fails its usefulness objective if, at a fixed verification budget, it causes materially more false invalidation or lower useful-current availability than simpler baselines without improving stale-claim escape.

3. Carry-forward from prior rounds

Round 22 froze three claim classes:

Class	Meaning	Required handling in Round 23
Historical finding	A statement about a committed past diagnostic or audit	Preserve the statement, commit, artifact digest, and interpretation boundary. Never rewrite it as a current-system claim.
Current claim	A statement about the candidate system whose truth may change	Mark stale when a dependency event could violate its validity predicate. Restore current status only after the named procedure succeeds.
Untested hypothesis	A proposed causal or cost claim without a completed experiment	Keep explicitly untested. A software diff cannot upgrade it.

The carried claim set contained two historical findings, five current negative claims, and two untested ABMG hypotheses:

- **C21-HIST-34:** D21-A passed 34 of 34 deterministic checks at its frozen Round 21 commit.
- **C21-HIST-10-9-12:** at that commit, 10 of 31 required fields were present, 9 partial, and 12 absent.
- **C21-CUR-ENVELOPE:** the current measurement path does not commit the exact provider request and response envelope.
- **C21-CUR-TERMINAL:** it does not persist provider terminal markers or finish/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 ABMG implementation exists.
- **H21-IMPACT:** the proposed gate would improve an answer-level or reliability outcome.
- **H21-COST:** the proposed gate would do so at an acceptable cost.

Round 23 treats claim status as a lifecycle, not as a confidence adjective:

`current -> stale_needs_revalidation -> current | contradicted | retired`

Historical findings do not enter that transition. Hypotheses remain proposed until their own pre-registered experiment is completed.

4. Scope and exclusions

4.1 In scope

- The first available Encevera project commit interval after the Round 22 manifest was frozen.
- Ten changed repository artifacts in that interval.
- Five current claims inherited from Round 21 and formalized in Round 22.
- An authored dependency view, an independently reconstructed static-import view, and a fail-closed rule for unknown production-looking paths.
- Claim staleness, claim revalidation, affected-artifact recall, false invalidation, useful-current availability, and bounded process measurements.
- Eight controlled mutations representing the explicit Round 22 handoff cases.

4.2 Out of scope

- Whether ABMG improves answer correctness, calibration, factuality, safety, or user trust.
- Whether model collaboration improves reasoning.
- Production enforcement, schema migration, provider traffic, user traffic, or database writes.
- Full compiler-resolved TypeScript dependency analysis, runtime tracing, service-to-service dependencies, environment variables, database triggers, or hosted platform configuration.
- Customer prompts, provider payloads, credentials, proprietary routing instructions, and sensitive operational thresholds.
- A general claim that static imports or CTAG are safe regression-selection methods.

The privacy boundary was paths, hashes, typed dependency tags, test identifiers, and redacted findings. This is consistent with the earlier minimal-disclosure direction: a revalidation receipt can prove what was checked without publishing sensitive content.

5. Literature map and competing positions

The literature does not provide a ready-made answer for current claims about collaborative AI systems. It provides several partially overlapping positions that can be tested in Encevera.

Position A: authored exact dependencies are the clearest control

An authored manifest is reviewable, stable, and easy to bind to a claim. Structured assurance models such as OMG SACM represent claims, arguments, evidence, and relationships explicitly [S13]. W3C PROV separates entities, activities, agents, derivations, and bundles [S11]. SLSA similarly emphasizes verifiable facts about where, when, and how an artifact was produced [S12]. These standards support typed, addressable evidence instead of prose-only provenance.

The weakness is coverage drift. A new helper, adapter, provider path, or service may enter the execution graph without being added to the manifest. Authorship improves meaning but does not guarantee completeness.

Position B: derived program dependencies can recover omitted relations

Program dependence work makes control and data dependencies explicit [S06]. Chianti decomposes code changes and relates them to affected tests [S04]. Ekstazi uses observed file dependencies to select regression tests and reports meaningful time savings in its studied Java projects [S05]. These lines of work support an independent view that is not merely a second copy of the same authored list.

The weakness is model mismatch. The D23-A import closure is much lighter than a program dependence graph or dynamic dependency collector. Imports do not reveal reflection, runtime configuration, external services, database behavior, prompt construction, generated code, or semantic changes inside an apparently unrelated module. Ekstazi itself notes assumptions and unsupported dependency types [S05]. A derived view can be complementary without being complete.

Position C: fail closed on unknown production paths

Safety-oriented regression selection defines safety in terms of not omitting tests that can expose relevant change-induced faults under the method's assumptions [S03]. Assurance-case maintenance research also warns that a change can invalidate evidence or assumptions in ways that are not obvious [S07]. Under that framing, an unknown production adapter should not silently pass simply because no edge has been authored yet.

The weakness is noise. If every unfamiliar file invalidates every claim, the policy converges on retest-all. It may protect recall while destroying the practical value of current claims.

Position D: retest everything is the simplest high-recall baseline

Retest-all avoids selector omission in a bounded claim set. It also supplies an important baseline: a selective mechanism is not useful if it reproduces all review at greater implementation and governance cost.

The weakness is scale. Chianti and Ekstazi exist because broad regression work can be costly [S04, S05]. Practitioner evidence also suggests that change-impact analysis requires meaningful, variable labor and is difficult to evaluate in isolation [S08]. Encevera must measure review labor rather than assume selection is cheaper.

Position E: assurance should evolve continuously

Dynamic and perpetual assurance research treats assurance as an enduring process rather than a static document [S09, S15, S16]. ACCESS demonstrates model-based traceability to heterogeneous artifacts and automatic assurance evaluation in an autonomous-underwater-vehicle case study [S09]. This supports Encevera's claim lifecycle and change-event framing.

The weakness is transfer. A safety or self-adaptive-system case study does not validate an AI collaboration protocol. The relevant lesson is structural: claims, evidence, dependencies, and change events can be modeled and evaluated. The effectiveness of Enceavra's particular graph remains empirical.

Position F: mutation suites are valuable but cannot stand in for real incidents

Just et al. found that mutant detection and real-fault detection were correlated in their studied programs, even when controlling for coverage, while also documenting limits [S10]. Mutation fixtures are therefore useful for checking that a policy reacts to known classes of event.

They are not independent evidence that the policy will discover unknown changes. Round 23 reports authored fixtures separately from the real post-freeze interval.

6. Proposed mechanism: the CTAG Shadow Gate

The proposed mechanism is a **dual-view CTAG shadow gate**. It does not block release. It observes a change, predicts which claims should become stale, executes or queues named revalidation procedures, and emits a minimal receipt.

6.1 Claim cards

Each claim card contains:

- stable claim identifier;
- class: historical, current, or hypothesis;
- exact statement and scope;
- validity predicate;
- typed dependency tags;
- authored artifact dependencies;
- named revalidation procedure;
- status and status timestamp;
- evidence or result commitment;
- privacy classification;
- supersession relation, if any.

The validity predicate is essential. A file path alone cannot explain why a change matters. For example, the envelope claim's predicate is that every production provider path still lacks a binding commitment to the full dispatched request and received response. A changed provider helper can therefore be relevant even if the measurement table is untouched.

6.2 Change-event cards

Each event card contains:

- base and candidate commit identifiers;
- changed path and content digest;
- event type, such as provider boundary, stream parser, schema, measurement storage, release path, or orchestration;
- production, test, documentation, or generated classification;

- authored-view matches;
- derived-view matches;
- unknown-path status;
- selected claims and reason codes.

No raw prompt or provider payload is needed in the public receipt.

6.3 Two dependency views

View 1: authored CTAG. This view carries the semantic edges written with each claim. It is strongest when the owner knows that an artifact implements a validity condition.

View 2: derived static imports. This view starts from claim-linked modules and expands through the module import graph. It is independent in construction, even though it is not statistically independent. Its purpose is to catch renamed or newly factored helpers that the exact list omitted.

6.4 Fail-closed unknown rule

A changed production-looking artifact that enters a claim-linked execution region but appears in neither view is classified as unknown. In shadow mode, it makes the broad current claim set stale and opens an edge-review task. Documentation, tests, generated outputs, and explicitly scoped non-production assets are excluded by policy.

This is deliberately conservative. The experiment must measure how often it degenerates into broad invalidation.

6.5 State transition and revalidation receipt

Selection does not contradict a claim. It changes status from current to stale. Only the claim's named procedure can restore current status. A minimal receipt includes:

- claim ID and prior status;
- triggering event ID and commit pair;
- selector views and reason codes;
- revalidation procedure version;
- inspected artifact commitments;
- outcome: restored, contradicted, retired, or unresolved;
- reviewer or process identity;
- timestamp and result digest.

Historical claims are never re-run as if they described the candidate system. Their artifact commitments may be rechecked for integrity, but their historical counts remain frozen.

7. Diagnostic D23-A: design

7.1 Temporal split

The Round 22 manifest at base commit `d6a394145d62f184f7dc40b99bca009d313df559` was verified with SHA-256 digest:

`f716fd9fbfea129503d76880b0f2286bafa9342fbee3eff4054c01f7243ccf99`

The candidate commit was `fe1ab926fc7d8a521fdff09d93792e9285e7aa2c`. The changed-path list contained ten artifacts. Before inspecting diff hunks or current source content, D23-A wrote and hashed a path-only prediction:

```
7942c9b1fb67bb59ceca7d36c03229b45016f4321a8bb2a5d6ebb4dc5bf4bf62
```

The changed-path list digest was:

```
0f535e7d3459f5077c81876d7a4de33b9b497d7068db0d98d99ff51183e37b57
```

This prevents the prediction artifact from being silently rewritten after content inspection. It does not create independent blinding.

7.2 Selectors

Five baselines and one hybrid were evaluated:

- 1 **Direct measurement watchlist:** only the core measurement paths named in the narrowest operational reading.
- 2 **Frozen exact-path manifest:** exact artifact matches from Round 22.
- 3 **Filename keyword:** a broad heuristic based on measurement, provider, release, run, and related names.
- 4 **Static import closure:** changed modules within the reconstructed import closure of claim-linked execution modules.
- 5 **Retest-all:** every changed artifact.
- 6 **CTAG hybrid:** union of authored exact edges, static import closure, and fail-closed unknown production paths, with documentation and test exclusions.

Each selector produced an artifact set and then a claim set. Artifact relevance was defined as: the observed semantic change could have changed the truth of at least one named current claim and therefore required claim-specific revalidation.

7.3 Gold labels

After the prediction was frozen, the same research process inspected every diff record and relevant current source path. It labeled three artifacts semantically relevant:

- `src/lib/ai-gateway.server.ts`
- `src/lib/ai-providers.server.ts`
- `src/lib/panel-core.server.ts`

Seven artifacts were labeled irrelevant to the five exact claims. These included documentation-only, test-only, type-only, cost-reservation-only, registry-copy, and diagnostic-description changes. The labels are reasoned adjudications, not ground truth guaranteed by an independent panel.

7.4 Metrics

For artifact selection:

- $\text{recall} = \text{relevant selected} / \text{all labeled relevant};$
- $\text{precision} = \text{relevant selected} / \text{all selected};$
- $\text{false positives} = \text{selected artifacts labeled irrelevant};$
- $\text{false negatives} = \text{relevant artifacts omitted}.$

For claims:

- $\text{stale-claim escape} = \text{affected current claims left current};$

- false invalidation = unaffected current claims made stale;
- useful-current availability = revalidated true current claims / all true current claims;
- historical rewrite count;
- hypothesis-upgrade count.

Process observations included direct diff records inspected, current source files inspected, base-current identity checks, and explicit automation wall time. Wall time is a lower-bound process observation, not a human-labor estimate.

8. D23-A results

8.1 Real post-freeze change interval

Selector	Selected	TP	FP	FN	Precision	Recall
Direct measurement watchlist	0	0	0	3	1.0000*	0.0000
Frozen exact-path manifest	3	2	1	1	0.6667	0.6667
Filename keyword	8	3	5	0	0.3750	1.0000
Static import closure	7	3	4	0	0.4286	1.0000
CTAG hybrid, fail closed	7	3	4	0	0.4286	1.0000
Retest-all	10	3	7	0	0.3000	1.0000

*Precision for the empty selection is reported as 1.0 by the diagnostic convention. It must not be read as useful performance; recall was zero.

The main positive observation is not that the hybrid achieved perfect recall. With three positive artifacts, one interval is too small for that conclusion. The useful observation is a traceable unique catch: static import closure added `src/lib/ai-gateway.server.ts`, which the frozen exact-path manifest omitted.

The main negative observation is cost. The hybrid selected seven of ten artifacts, four of which were labeled irrelevant. It was more selective than retest-all but still broad. Filename keywords matched recall while adding one more false positive, showing that crude lexical breadth can look competitive in a small interval.

8.2 Claim selection and availability

The hybrid made all five current claims stale. Post-hoc adjudication associated the observed semantic changes with three claims:

- C21-CUR-ENVELOPE
- C21-CUR-GATE
- C21-CUR-NO-ABMG

Claim recall was therefore 1.0 and claim precision 0.6 in this interval. C21-CUR-TERMINAL and C21-CUR-ATOMIC were false invalidations. Useful-current availability fell from 1.0 to 0.0 when the five claims were marked stale, then returned to 1.0 after all five named procedures completed.

This exposes a distinction that artifact recall alone hides. The independent import view found a unique artifact, but the union of existing edges already saturated the five-claim set. The new view added artifact-level protection without adding claim-level selectivity.

8.3 Revalidation findings

Claim	Trigger	Revalidation finding	Status
Envelope commitment	Provider request and routing changes	Output-cap serialization changed, but RunTrace and AttemptRow still retain hashes and operational metadata rather than commitments to complete serialized request and response envelopes.	Revalidated current negative claim
Terminal evidence	Conservative broad invalidation	One gateway path observes message_stop and refusal internally; direct paths ignore terminal sentinels as recordable evidence. No uniform terminal or finish/stop reason is persisted.	Revalidated current negative claim
Atomic completeness	Conservative broad invalidation	Four core completion and measurement files were byte-identical across the interval. Terminal state still precedes three best-effort measurement upserts whose failures are logged rather than propagated.	Revalidated current negative claim
Receipt gate	Final-check retry and release-diagnostic changes	The new bounded retry and diagnostics do not depend on measurement-receipt completeness. Measurement remains post-terminal and non-blocking.	Revalidated current negative claim
No production ABMG	Changes across provider and orchestration paths	The changes improve availability and diagnostic clarity but add no attempt receipt, atomic completeness transaction, or receipt-qualified release label.	Revalidated current negative claim

The two historical claims remained frozen. Their past counts were not replaced by candidate-state observations. **H21-IMPACT** and **H21-COST** remained untested.

8.4 Process observations

D23-A inspected ten direct diff records, thirteen current source files, and four base-current identity checks. The explicit automated interval between recorded revalidation markers was 120 seconds. This excludes preparation, literature review, manuscript work, human attention, and any future operational integration. It is not evidence that CTAG is cheap.

The diagnostic result, excluding its own hash field, has SHA-256 digest:

2231bfba62ca1c66b2e6e339471460e1f372eb06778e4655c043aef2d479d481

9. Controlled mutation suite

The eight fixtures correspond to the Round 22 handoff:

- 1 new imported provider helper changes a serialized request;
- 2 known parser ignores a terminal marker;
- 3 new completion helper writes terminal state before measurement;
- 4 measurement writer swallows a required write failure;
- 5 unknown production runtime adapter enters the graph;
- 6 documentation-only change;
- 7 test-only diagnostic change;
- 8 pricing-only change outside the five claim scopes.

Selector	Exact cases	Event recall	Claim-edge precision	Claim-edge recall	False-alarm events	Mean claims/event
Direct watchlist	3/8	0.2000	0.6000	0.2000	0	0.625
Frozen exact paths	3/8	0.2000	0.6667	0.1333	0	0.375

Selector	Exact cases	Event recall	Claim-edge precision	Claim-edge recall	False-alarm events	Mean claims/event
Filename keyword	2/8	0.8000	0.4000	0.8000	2	3.750
Static import closure	7/8	0.8000	1.0000	0.6667	0	1.250
CTAG hybrid	8/8	1.0000	1.0000	1.0000	0	1.875
Retest-all	1/8	1.0000	0.3750	1.0000	3	5.000

These figures test whether the implementation matches the authored policy on authored cases. They do not estimate production incident rates, future precision, or future recall. The perfect hybrid score is expected evidence of conformance, not evidence of generalization.

10. Worked example: the provider gateway catch

The real interval changed `src/lib/ai-gateway.server.ts` so that an Anthropic gateway request used dynamic output-token headroom. That is a real change to the serialized request envelope.

The frozen exact-path manifest did not name this artifact. A measurement-only watchlist also missed it. The static-import view reached it through provider and panel modules already connected to the claims. The hybrid therefore made `C21-CUR-ENVELOPE` and `C21-CUR-NO-ABMG` stale.

The named procedure then asked two separate questions:

- 1 Did the provider request change? Yes. The output-cap field and related dispatch path changed.
- 2 Did the measurement contract begin committing the complete request and response, with attempt lineage and receipt-gated release? No. The trace projection still exposed selected hashes and operational fields rather than binding commitments to complete envelopes.

The current negative claims were therefore restored as current. This is exactly the intended distinction between change detection and claim contradiction. A gate should say "inspect this" before it says "the claim is false."

The example also shows why import closure is insufficient by itself. A runtime-loaded adapter, an environment change, a database trigger, or a provider contract change may alter behavior without an import edge. D23-A's unique catch is a reason to retain the second view, not a reason to trust it alone.

11. Claim/evidence table

Claim	Evidence status	Direct support	What remains open
The archive's next number is 23	Established for this run	Live overview, index, publications list, and Round 22 page ended at 22	Future archive changes after this run
The post-freeze interval changed ten artifacts	Established for D23-A	Commit comparison and committed changed-path digest	Whether all external dependencies were captured
Three artifacts were semantically relevant to the five claims	Bounded adjudication	Full diff and source inspection with written reasons	Independent reviewer agreement and latent semantic effects
The exact-path manifest missed one labeled relevant artifact	Established for D23-A	It omitted <code>ai-gateway.server.ts</code> ; label tied the change to envelope claims	Frequency across more commits
The import view added that artifact	Established for D23-A	Reconstructed import closure and selector trace	Completeness of the reconstructed graph
The hybrid had 1.0 artifact recall and 0.4286 precision	Descriptive result only	Three TP, four FP, zero FN on ten artifacts	Confidence interval, external validity, independent labels
All five current negative claims remain true at candidate commit	Current source-audit finding	Named revalidation procedures and source checks	Runtime behavior not exercised; future changes
The hybrid is safer than exact paths	Open hypothesis	One unique catch is consistent with the hypothesis	Multi-commit independent trial under fixed budget
The hybrid is cheaper than retest-all	Open hypothesis	Seven versus ten selected artifacts in one interval	Selector overhead, human time, repeated-change load
CTAG improves Encevera answer reliability	Untested	No answer benchmark was run	Causal experiment with answer-level outcomes

12. Alternative explanations

12.1 The unique catch may reflect an incomplete manifest, not the superiority of import analysis

Round 22 authored its dependency table manually. Adding the missing gateway path to that table would have let exact matching catch the change. The result may therefore show that a review step was incomplete, not that import closure is generally better. E23-B must compare the maintenance burden and drift of both views over time.

12.2 The same researcher may have favored the frozen hypothesis

Although the prediction artifact preceded content inspection, the same process later assigned relevance labels. Knowledge of the mechanism could influence borderline labels. Independent reviewers and hidden selector outputs are required.

12.3 Five claims make saturation likely

C21-CUR-NO-ABMG is deliberately broad. Provider or orchestration changes frequently touch it, and union routing can quickly invalidate all five claims. In a larger registry, the same artifact-level improvement might or might not preserve claim-level precision.

12.4 The commit was unusually favorable to imports

The key relevant omission was an imported TypeScript module. Static imports are well suited to that case. A configuration-only, database-only, hosted-service, or provider-contract change could reverse the result.

12.5 Negative claims may be easier to revalidate

The five current claims assert missing capabilities. Source inspection can often confirm absence without running a behavior benchmark. Positive performance or reliability claims may require substantially more costly evidence and may have different dependency patterns.

12.6 Retest-all may be competitive at this scale

Ten changed artifacts and five claims are small. Saving three artifact reviews may not repay graph maintenance, selector execution, receipt review, and governance. The relevant comparison is end-to-end cost at a fixed assurance target, not selector count alone.

13. Falsifiable hypotheses

These hypotheses remain proposed unless explicitly tied to the bounded D23-A observation.

H23-1, stale-claim escape: Across a preregistered multi-commit window, the hybrid will leave fewer independently adjudicated affected claims current than the frozen exact-path manifest at the same claim-revalidation budget.

H23-2, unique-view value: Removing the static-import view will increase affected-artifact false negatives on at least one held-out commit without an equal reduction in false-invalidated claims.

H23-3, unknown-path value: Removing the fail-closed unknown rule will increase stale-claim escape on held-out adapter, configuration, or execution-path changes.

H23-4, selectivity: At equal stale-claim escape, the hybrid will select fewer artifact-review tasks and fewer unaffected claims than retest-all.

H23-5, availability: The hybrid plus cached claim-specific procedures will maintain greater useful-current availability over time than a policy that expires all current claims on every production change.

H23-6, lifecycle integrity: Neither selector execution nor revalidation will rewrite historical findings or upgrade untested hypotheses in any evaluated commit.

H23-7, cost: End-to-end review labor, including graph maintenance and selector triage, will be lower for the hybrid than retest-all at matched stale-claim escape.

H23-8, mismatch trigger: Revalidation triggered by a typed dependency mismatch will achieve lower unnecessary revalidation than periodic expiry at matched stale-claim escape.

D23-A is consistent with H23-2 and H23-4 in one interval, but it does not establish them. It did not test H23-1, H23-5, H23-7, or H23-8 over time.

14. Proposed experiment E23-B

14.1 Objective

Measure the hybrid's stale-claim escape, over-invalidation, availability, and total review cost across genuinely unseen Encevera changes under controlled verification budgets.

14.2 Prospective freeze

Before the evaluation window begins:

- 1 Freeze the claim registry, validity predicates, dependency tags, authored path view, import-graph extractor, unknown-path classifier, selector versions, and revalidation procedures.
- 2 Hash the complete protocol and store its commit.

- 3 Select a future time window or next N qualifying commits without seeing their content.
- 4 Define exclusions for generated artifacts, formatting-only commits, test-only commits, and emergency security work.
- 5 Pre-register handling of merges, renames, vendored code, monorepo moves, provider-contract changes, and database migrations.

14.3 Units and sample

The primary unit is a commit-event pair. A qualifying event contains one candidate commit relative to its first parent and all changed artifacts visible to the project repository. The initial target should be at least 30 qualifying events spanning provider, orchestration, measurement, release, schema, configuration, and documentation changes. This is a design target, not a power calculation. A statistician or experiment owner should perform a power analysis after estimating event prevalence from a non-evaluation pilot.

14.4 Independent adjudication

Two reviewers who cannot see selector outputs independently label:

- whether each changed artifact could alter each claim's validity predicate;
- which claims require revalidation;
- whether the named procedure is sufficient;
- whether revalidation restores, contradicts, retires, or leaves the claim unresolved.

Disagreements are recorded before adjudication. Report raw agreement, claim-edge Cohen's kappa where defined, and adjudicated labels. If reviewers cannot assess a dependency without protected operational details, the item is marked unresolved rather than negative.

14.5 Baselines and ablations

Run all methods on the same frozen events:

- direct measurement watchlist;
- frozen exact paths;
- filename keywords;
- static import closure only;
- hybrid without import closure;
- hybrid without unknown-path fail-closed routing;
- hybrid without exclusions;
- retest-all;
- periodic full expiry at a fixed cadence;
- random task ordering, repeated over fixed seeds, as a budget-control sanity baseline.

14.6 Controlled total verification budget

Define two preregistered budgets per event:

- **B_artifact**: maximum changed-artifact inspections;
- **B_claim**: maximum claim-specific revalidation procedures.

Every method emits an ordered queue before labels are revealed. Only the first **B_artifact** and **B_claim** tasks are executed for the primary comparison. Use identical source snapshots, tooling, reviewers, time limits, and

permitted evidence. Selector computation and graph-maintenance time are recorded separately and included in the cost analysis.

Evaluate a budget curve rather than one arbitrary point. Recommended points are 25%, 50%, 75%, and 100% of the retest-all task count for each event, capped at the event's total available tasks. A method cannot claim savings by silently doing more verification.

14.7 Primary outcomes

- stale-claim escape rate;
- affected-claim recall;
- affected-artifact recall;
- false-invalidated claims per event;
- false-positive artifact reviews per event;
- useful-current availability over wall-clock time;
- unresolved-claim duration;
- total reviewer minutes;
- selector and graph-maintenance minutes;
- budget-normalized correct revalidations.

14.8 Harm and regression metrics

- publicly current claims later found stale;
- historical findings overwritten or detached from their commitments;
- hypotheses incorrectly upgraded to findings;
- unknown production paths allowed without review;
- privacy leakage in event cards or receipts;
- release delay attributable to false invalidation;
- repeated noisy invalidations ignored by reviewers;
- claim status restored without executing the named procedure;
- selector disagreement hidden rather than logged;
- increase in answer or run failure rate if any future enforcement pilot is separately authorized.

14.9 Analysis

Report paired event-level differences because every selector sees the same commits. Use exact confidence intervals for rare stale-escape events, bootstrap intervals for labor and availability, and sensitivity analyses that count unresolved labels first as positive and then as negative. Preserve per-event traces so a favorable aggregate cannot hide one catastrophic miss.

The release decision should be asymmetric. Any independently confirmed affected claim left current by the hybrid is a stop-and-revise event. High false-invalidation rates are not safety escapes, but they can still reject the mechanism as operationally unusable.

15. Brain-inspired hypothesis, kept testable

No claim of literal brain copying is needed. The useful prior analogy is mismatch-triggered memory updating: a stored representation is revisited when new evidence conflicts with what its validity conditions predict. Round 23 translates that into an engineering comparison between two policies:

- **mismatch-triggered revalidation:** typed dependency events make only implicated claims stale;
- **periodic expiry:** all current claims expire on a fixed schedule, whether or not a relevant change occurred.

The analogy earns its place only through H23-8. E23-B can compare unnecessary revalidation and stale-claim escape at matched budgets. If periodic expiry performs as well or better, the mismatch-triggered design loses support. The result would say nothing about whether Enceavra is brain-like.

16. Expected failure modes and safeguards

Failure mode	Consequence	Detection	Safeguard
Missing authored edge	Affected claim stays current	Independent view or later adjudication catches omitted artifact	Dual views, edge-drift audit, fail-closed unknowns
Incomplete import graph	Runtime or external dependency missed	Held-out config/service mutations; incident review	Add typed non-code events and optional dynamic traces
Broad unknown classifier	Excess invalidation	False-invalidated claims and reviewer minutes rise	Explicit production/test/doc taxonomy; review unknown edges once
Broad aggregate claim	Claim set saturates	Mean claims per event approaches registry size	Split claims by provider, execution mode, and invariant
Circular evidence	Selector output influences gold label	Reviewer sees reason codes	Blind independent adjudication
Historical rewrite	Past result appears to change	Commitment and status audit	Immutable historical cards and append-only supersession
Procedure drift	Old procedure no longer tests the predicate	Procedure-version mismatch	Bind each claim to procedure version and validity schema
Privacy leak	Sensitive payload enters receipt	Receipt schema audit and canary tests	Paths, hashes, typed findings, and redaction only
Gate fatigue	Reviewers ignore repeated stale flags	Acknowledgment delay and override rate	Measure noise; cap scope; do not enforce before shadow evidence
False confidence from fixtures	Authored cases mistaken for incidents	Reporting audit	Keep conformance and real-event sections separate
Graph maintenance cost	Selective system costs more than retest-all	End-to-end labor accounting	Include maintenance and triage in H23-7
Common-origin failure	Both views share the same blind spot	Correlated misses across ablations	Add runtime/config/provider-contract view in later work

17. Limitations

- 1 Only one post-freeze real commit interval was evaluated.
- 2 The same research process froze the path-only prediction and later adjudicated semantic impact. Temporal separation is not independent blinding.
- 3 The static import closure was reconstructed from inspected imports, not a complete compiler-resolved graph.
- 4 The eight mutation cases are authored conformance fixtures, not sampled production incidents.
- 5 A five-claim scope saturates quickly. Artifact-level gains did not yield claim-level selectivity.

- 6 The candidate interval may not represent changes to configuration, database behavior, external provider contracts, infrastructure, or runtime-loaded modules.
- 7 Revalidation used source inspection and identity checks; it did not exercise production traffic or provider behavior.
- 8 The 120-second automated interval excludes substantial work and cannot support a cost claim.
- 9 Relevance labels depend on exact claim wording. A broader security, quality, or availability claim could label the same artifacts differently.
- 10 No answer-level outcome was measured, so this round does not show more reliable reasoning or collaboration.
- 11 No production CTAG or ABMG implementation was added.
- 12 Literature from regression testing and safety assurance supplies design patterns, not direct validation for collaborative AI systems.

18. Architecture and product decision

Decision: no architecture change. Keep CTAG in shadow research status.

D23-A clears only a weak continuation threshold: the independent view produced one traceable unique catch, the hybrid did not miss a labeled relevant artifact in the one real interval, and lifecycle invariants were preserved. It does not clear a release-gating threshold.

The next implementation step, if separately authorized, should be a non-blocking recorder that emits redacted event and claim-status receipts. It should not modify provider requests, execution routing, release decisions, public claim status, or production databases. The recorder should make it possible to run E23-B without reconstructing evidence after the fact.

19. Next-round handoff

Round 24 should not repeat D23-A on another convenient commit and call the mechanism validated. It should do one of two things:

- 1 **Preferred:** preregister and begin E23-B, with independent hidden adjudication and fixed budget curves across future commits.
- 2 **If a prospective window cannot yet be supported:** study common-origin and source-authority failure. Ask when two apparently independent dependency views inherit the same blind spot, and define a third event source for configuration, schema, provider-contract, or runtime evidence.

The handoff must preserve these stops:

- Stop if an unknown production path can enter without a stale flag.
- Stop if current status can be restored without the named procedure.
- Stop if historical findings are rewritten.
- Stop if untested hypotheses are upgraded.
- Stop if sensitive payloads enter receipts.
- Stop if evaluation labels cannot be blinded from selector output.

The next round should report independent reviewer agreement, at least one held-out miss opportunity, selector overhead, and useful-current availability. It should not claim answer reliability until an answer-level causal benchmark is separately run.

20. Source inspection ledger

ID	Source and date	What was inspected	What it supports here
S01	Encevera Round 22, 2026-10-13	Full manuscript, manifest, statuses, handoff	Frozen CTAG and prospective-test requirement
S02	Encevera Round 21, 2026-10-12	Claim cards and diagnostic commitments	Five current claims, two historical findings, two hypotheses
S03	Rothermel and Harrold, 1997	ACM record and abstract	Safe-selection framing under assumptions
S04	Ren et al., 2004	IBM page, abstract, case-study summary	Atomic changes and affected-test analysis
S05	Gligoric et al., 2015	Full paper	Dynamic dependencies, savings, assumptions, limits
S06	Ferrante et al., 1987	ACM record and paper description	Explicit derived dependency representation
S07	Jaradat et al., 2014	Primary record and abstract	Change can invisibly invalidate assurance evidence
S08	Borg et al., 2016	Primary record and abstract	CIA labor and evaluation difficulty
S09	Wei et al., 2024	Version 2 record and abstract	Evolving model-based assurance and traceability
S10	Just et al., 2014	Author-hosted abstract and study design	Mutants as limited proxy, not real incidents
S11	W3C PROV-DM, 2013	Recommendation abstract and core model	Typed provenance entities, activities, agents, derivations
S12	SLSA v1.2, inspected 2026-10-14	Approved provenance page	Verifiable where/when/how artifact facts
S13	OMG SACM 2.3, 2023	Official spec page and artifacts	Structured assurance claims and evidence
S14	NIST SSDF 1.1, 2022	Official publication page	Lifecycle-integrated verification vocabulary
S15	Weyns et al., 2017/2019	Primary record and abstract	Assurance as an enduring lifecycle process
S16	Calinescu et al., 2018	Version 2 record and abstract	Runtime/design-time evidence in bounded cases

The machine-readable source ledger distributed with this paper records working links, publication dates, inspected material, and explicit non-support boundaries.

21. Full sources

- 1 Encevera Research. **Round 22: When Should a Current Claim Expire?** 2026-10-13. <https://id-preview--8483bc59-da36-492e-b8b8-6bf684070241.lovable.app/research-round-22>
- 2 Encevera Research. **Round 21: What Evidence Is Missing Before We Test a Hard Release Gate?** 2026-10-12. <https://id-preview--8483bc59-da36-492e-b8b8-6bf684070241.lovable.app/research-round-21>
- 3 Rothermel, G., and Harrold, M. J. **A Safe, Efficient Regression Test Selection Technique.** ACM Transactions on Software Engineering and Methodology 6(2), 173-210, April 1997. <https://doi.org/10.1145/248233.248262>
- 4 Ren, X., Shah, F., Tip, F., Ryder, B. G., and Chesley, O. **Chianti: A Tool for Change Impact Analysis of Java Programs.** OOPSLA 2004. Published 2004-12-01. <https://research.ibm.com/publications/chianti-a-tool-for-change-impact-analysis-of-java-programs>
- 5 Gligoric, M., Eloussi, L., and Marinov, D. **Ekstazi: Lightweight Test Selection.** ISSTA 2015. <https://users.ece.utexas.edu/~gligoric/papers/GligoricETAL15Ekstazi.pdf>
- 6 Ferrante, J., Ottenstein, K. J., and Warren, J. D. **The Program Dependence Graph and Its Use in Optimization.** ACM TOPLAS 9(3), 319-349, July 1987. <https://doi.org/10.1145/24039.24041>
- 7 Jaradat, O., Graydon, P., and Bate, I. **An Approach to Maintaining Safety Case Evidence After A System Change.** Submitted 2014-04-28. <https://arxiv.org/abs/1404.6846>

- 8 Borg, M., de la Vara, J.-L., and Wnuk, K. **Practitioners' Perspectives on Change Impact Analysis for Safety-Critical Software - A Preliminary Analysis**. Submitted 2016-05-23.
<https://arxiv.org/abs/1605.07105>
- 9 Wei, R., et al. **ACCESS: Assurance Case Centric Engineering of Safety-critical Systems**. Version 2, 2024-04-16. Journal of Systems and Software, DOI 10.1016/j.jss.2024.112034.
<https://arxiv.org/abs/2403.15236>
- 10 Just, R., Jalali, D., Inozemtseva, L., Ernst, M. D., Holmes, R., and Fraser, G. **Are Mutants a Valid Substitute for Real Faults in Software Testing?** FSE 2014.
<https://homes.cs.washington.edu/~mernst/pubs/mutation-effectiveness-fse2014-abstract.html>
- 11 W3C. **PROV-DM: The PROV Data Model**. W3C Recommendation, 2013-04-30.
<https://www.w3.org/TR/prov-dm/>
- 12 SLSA. **SLSA v1.2 Provenance**. Approved specification, inspected 2026-10-14.
<https://slsa.dev/spec/v1.2/provenance>
- 13 Object Management Group. **Structured Assurance Case Metamodel Version 2.3**. Adopted October 2023.
<https://www.omg.org/spec/SACM/2.3/About-SACM>
- 14 NIST. **Secure Software Development Framework (SSDF) Version 1.1**. SP 800-218, 2022-02-03.
<https://csrc.nist.gov/pubs/sp/800/218/final>
- 15 Weyns, D., et al. **Perpetual Assurances for Self-Adaptive Systems**. Journal reference 2017; arXiv record 2019-03-12. <https://arxiv.org/abs/1903.04771>
- 16 Calinescu, R., et al. **ENTRUST: Engineering Trustworthy Self-Adaptive Software with Dynamic Assurance Cases**. Version 2, 2018-11-22. <https://arxiv.org/abs/1703.06350>

22. Reproducibility manifest

The companion bundle contains:

- this manuscript in Markdown;
- `changed_paths.json`;
- `preinspection_prediction.json`;
- `shadow_claim_cards.json`;
- `actual_impact_labels.json`;
- `mutation_fixtures.json`;
- `shadow_gate.py`;
- `diagnostic_results.json`;
- `source_ledger.json`;
- `round22_claim_dependency_manifest.json`;
- an integrity manifest with SHA-256 digests.

The diagnostic is deterministic and makes no network, provider, model, production-database, or production-write calls.

23. Final claim boundary

Round 23 establishes one bounded fact: in the first studied post-freeze Encevera commit interval, a second, import-derived dependency view added one relevant artifact that the frozen exact-path manifest missed, while the fail-closed hybrid selected more review work and invalidated two unaffected claims. The hybrid conformed to eight authored mutation fixtures and preserved the historical/current/hypothesis distinction. This justifies a larger prospective shadow trial. It does not justify production enforcement, a safety claim, a cost claim, or a claim that Encevera's answers are more reliable.