PREPRINT · MEASUREMENT STUDY · 15 JUNE 2026

# Admit Only What Re-Derives: A Preregistered, Verifier-Arbitrated Study of Trust Surfaces in Persistent Agents

Max Griswold

Nexi Technologies

Companion to the open-source `veriker` verifier and the `relyable` gate. Preregistrations and reports release alongside the veriker drop; see §8.

Download PDF · Code & data · DOI: 10.5281/zenodo.20724553

ABSTRACT

Autonomous agents increasingly consume stored trust surfaces — a skill's claimed verdict, a recalled memory note, their own report that a task is done — as fact. We ask whether re-deriving such a claim, rather than trusting its label, changes agent behavior, and we pre-commit to what counts as success before running anything. Across two models (two sizes within the Claude 4.x family, `claude-sonnet-4-6` and `claude-haiku-4-5`), two structurally different task families, and a preregistered arm matrix arbitrated solely by a deterministic verifier, we report three findings, each labeled by evidence tier.

**(1)** On a cross-session memory-poisoning surface, a mandatory out-of-band re-derivation gate collapses corroborated-poison attack-success to the floor (ASR 0.000, flat across save strength, both tiers) precisely where the agent's own optional self-search over a corroborating channel **amplifies** it (ASR up to 0.950): the rise reflects the content of a channel built to corroborate the poison, not a measured verification skill, and its magnitude is corpus-construction-dependent. Stated alongside, and load-bearing: when the channel reliably carries the truth, the agent's own check ties the gate. The gate's value is channel-integrity-independence, not agent helplessness. **(2)** Every measured win is harm-neutralization, not uplift. An effect that looked in a pilot like the gate makes agents better (Δ+0.60) reversed to null under power (Δ+0.06, p=0.68); we

publish the reversal in full, because preregistration is what makes it legible. **(3)** An unattested surface degrades agents; a re-derivable one restores baseline.

The artifact at the end — `relyable`, an Apache-2.0 re-derivation gate built on the open `veriker` verifier — is the smallest of our claims. This is benchmark-grade measurement, not a deployed-fleet study; we report sign and direction across models, not a transfer claim.



## 1 Introduction

Consider an agent that finishes a task and reports: `✓ done — all tests pass.` Re-derive the run in a clean context and the tests resolve `NOT-RUN`; the agent wrote the summary, not the result. Under a gate that admits only what re-derives, the verdict is `ERROR`, fail-closed — the claim is refused because it does not reproduce, not flagged as low-confidence and passed through. We reproduce this failure mode under control in §5.3, where a negative-control run exhausted its retries and landed `ERROR` rather than a silent pass.

That gap — between what a persistent agent reports and what actually re-derives — is what we set out to measure before building anything to address it. The motivation is not hypothetical. More than 40% of agentic AI projects are forecast to be canceled by the end of 2027 [7]; AI models introduced an OWASP Top-10 vulnerability in 45% of code samples across more than 100 models and four languages [8]; 81% of enterprise technology leaders report increased production issues attributable to AI-generated code, and 70% say maintaining the test suite is now a larger burden than writing code [9]. The agents themselves carry open, linkable demand for exactly the surfaces we study: a persistent-agent harness whose agent authors its own skills with **no external validation point** [12], and a file-memory agent whose recalled notes are trusted with no freshness or provenance check, so a stale note silently overrides current truth [13][14].

This is a measurement paper. We treat three things a persistent agent trusts — its **skills**, its **memory**, and its own **verdicts** — as trust surfaces (a trust surface is a

stored artifact an agent consumes as fact without recomputing it), and we ask one question of each: if the consumer re-derives the claim instead of trusting the label, what changes? The three are a taxonomy, not one shared apparatus: skills and memory pass through the shared `relyable.gate` admission path, while verdicts re-derives through its own runner (§5.3). We pre-registered every experiment (design, arms, primary statistic, stopping rule) before any agent solved anything, and a deterministic verifier — not the solver's self-report — is the sole arbiter of every outcome. Solver self-reports were discarded; across every experiment they lied in both directions (cells narrated "verified, correct" while landing a rejected answer, and the reverse). That an agent's self-asserted confidence is not a usable trust signal is itself one of our findings.

Our headline result is counterintuitive and is the one a reviewer cannot wave away: on a poisoned memory surface, letting the agent verify itself by searching a corroborating channel makes it **more** wrong, while a mandatory out-of-band re-derivation gate drives the attack-success rate to zero. But the same data carry the honest bound that keeps that headline from overclaiming, and we state them together throughout. The tool we ship is deliberately the last and smallest thing in the paper.

## 2 Related work, and how we reconcile with it

Our results line up with the recent literature rather than fighting it; where they appear to conflict, the conflict resolves on closer reading.

**Self-correction is a labeling artifact, not a capability ceiling.** Huang et al. [4] established that LLMs do not reliably self-correct reasoning unaided. The sharper recent result is the Self-Correction Illusion [1]: relabeling a model's own erroneous content as an external source raises its explicit-correction rate by 23 to 93 percentage points (range across 13 model-by-domain cells) — a chat-template role artifact, not a reasoning limit. One of our memory experiments reproduced this from the other side: a loud "this is confirmed correct, do not re-derive" cue, read by the agent as an external, untrusted instruction, **provoked** verification rather than suppressing it (§5, Exp M). We do not claim "loud warnings always backfire"; we read both results as the same mechanism — externalize a claim and the agent will check it. Our gate is a way to externalize **every** claim an agent makes about its own

work, so the check no longer depends on the agent choosing to treat the claim as someone else's.

**Memory poisoning propagates, without an attacker.** The HEARTBEAT study [2] shows that routine background memory-saving promotes adversarial content to durable memory at rates up to 91% (their Table 5a, strongest save level), with cross-session attack-success up to 76%, and that enabling the agent's own web search dents but does not eliminate it (from 80% to 60% in the software domain at the strongest level). Prompt injection is not required; ordinary social misinformation suffices. HEARTBEAT names the head-to-head we run — mandatory out-of-band re-derivation versus the agent's optional discretion-verification — but did not test it. The "Taming OpenClaw" threat taxonomy [3] frames the lifecycle (memory poisoning, skill supply-chain contamination, indirect prompt injection, intent drift as compound cross-stage risks) and observes that re-derivation and out-of-band attestation are an open gap. We adopt HEARTBEAT's promotion model as a stated assumption (we do not re-measure promotion; §7) and measure the recall-edge contrast it left open.

## 3 The gate

The object under study is `relyable.gate`: a host-side, out-of-band re-derivation admission gate. It admits a stored artifact — a skill's claimed verdict, a recalled memory note — only if that artifact re-derives against the consumer's own trusted grader, run through the open-tier `veriker` verifier. The agent that produced the artifact cannot self-attest past the gate, because the gate never trusts a label, only a recomputation — the producer's label is never load-bearing.

### 3.1 Three structural properties

These are properties of the mechanism, not configuration options.

1. **Verify, do not execute (by default).** Admission is a re-derivation — recompute the claimed result in a clean context and compare — not a re-run of arbitrary producer code. Executing an in-bundle pack is a separately gated lane (`permit_execution` is an explicit caller argument, never a default), so the generic safe path cannot be turned into an execution primitive by a poisoned bundle.

2. **Digest-pinned grader.** The grader that decides admission is always the consumer's own trusted copy, pinned by SHA-256 **before** any verifier verdict is trusted. A bundle that ships its own grader, or omits one, cannot produce a trusted admit. This closes a real hole: an absent grader would otherwise let the raw verifier return `NO_PACK` with `ok=True` and never grade the candidate — a spurious pass.
3. **A poisoned or unverifiable entry is inert — no soft-admit.** It is dropped, not passed through as a weak signal. There is no "admitted with low confidence" middle state to ride past: either the artifact re-derived under the pinned grader, or it did not exist as far as the consumer is concerned.

The dependency direction is one-way: the gate imports only the open verify surface of `veriker` (the bundle verifier, the verdict type, the re-derivation invocation check). It is a genuine consumer of that verifier, not a fork of it.

### 3.2 The two entry points

The builder assembles a content-addressed bundle whose grader is **always** copied from the consumer's trusted source, so a poisoned bundle cannot ship a lying grader. The builder computes digests to assemble the bundle but is not itself the trust rail — the verifier's independent strict-SHA walk re-checks every digest at verify time, and that walk is the rail.

```
relyable.gate.build_bundle(
    dest, *, candidate_body, meta, grader_src, bundle_id, ...) -> Path
  # skill/candidate.py    candidate body (digest-bound)
  # skill/meta.json       consumer-shaped meta
  # re_derive/<pack>      grader, ALWAYS copied from grader_src
  # manifest.json         declares the re-derivation invocation; digest-binds every fi

relyable.gate.verify(
    bundle_dir, *, grader_src, permit_execution=False, ...) -> GateResult
```

The verify call is a fail-closed sequence. (i) Grader present? Absent → `NO_GRADER`. (ii) Grader byte-identical to the consumer's trusted copy? Mismatch → `GRADER_MISMATCH`, and the verdict is never run. (iii) Only then run the re-derivation-invocation check and wrap the verifier's verdict. Because the pin happens before any `ok` is trusted, the gate is safe over an arbitrary directory — not only over bundles the consumer itself

built — which is what lets it stand between an agent and a stored artifact of unknown provenance.

```
GateResult:
  grader_ok          bool
  grader_reason_code  NO_GRADER | GRADER_MISMATCH | None
  verdict            the raw veriker Verdict
  .ok                admitted
  .is_error          could-not-conclude (e.g. RE_DERIVATION_NOT_EXECUTED)
  .first_reason_code , .detail
```

The shared layer never decides the consumer's admit/reject label; each surface binding maps this result into its own vocabulary. That separation is what lets one gate core serve both the skills and memory bindings, while the verdicts surface (§5.3) re-derives through its own runner rather than this admission path.

> THE HONEST BOUNDARY — CARRIED VERBATIM
>
> The gate is a correctness / anti-silent-failure tool, not a performance booster. It does not make agents smarter, faster, or cheaper (in one pilot, gated memory cost **more** tokens). Its wins are harm-neutralization: it stops a poisoned surface from making an agent worse, and restores the baseline an unvetted surface erodes. It is integrity, not correctness — it tells you whether the thing the agent claimed actually re-derives, which is a narrower claim than "your code is right." That narrowness is the point, and it is the only claim defensible in a hostile room.

## 4 Method

**Preregistration discipline.** Every experiment has a frozen preregistration committed before any agent solved anything: design, arms, primary statistic, and the section-5 decision rule (the exact inequality on the primary statistic that would count as support) fixed in advance. The verdict is computed mechanically from the frozen aggregates. Where we changed a preregistered rule, we disclose it (§7, Amendments 1 and 2) rather than silently re-deriving the bar.

**The verifier is the sole arbiter.** A fresh agent instance solves each cell behind a byte-identical neutral wrapper; the cell prompt is the sole stimulus. Outcomes are scored by a deterministic verifier from a structured output field, never from the solver's prose. For the poisoning experiment the dispatch is the contamination-free

direct model API with a neutral software-assistant system prompt and no project context (see §7 for why this matters).

**Two task families.** A numeric statistical-power formula (continuous scoring) and a genomic interval-union task (binary pass/fail with exact-set-equality checks); the memory-poisoning line additionally uses a published-advisory recognition substrate with a sealed, externally verified ground-truth reference. Two model scales throughout (two sizes within the Claude 4.x family): `claude-sonnet-4-6` (stronger) and `claude-haiku-4-5` (weaker).

**The arm matrix.** Across experiments the arms instantiate a common pattern: a no-surface baseline (A); the surface present but unattested (a stale or unlabeled note, an unlabeled registry); the surface present and re-derivable through the gate; and, for the poisoning line, the agent's own optional verification over either a corroborating channel or a truth-carrying channel. Statistics are paired exact-McNemar contrasts (binary families) or the preregistered primary on the continuous family, at the preregistered α.

## 5 Results

Every number below is labeled `POWERED` or `PILOT`. A pilot number never borrows a powered experiment's confidence. The memory surface leads, and within it the poisoning experiment (H3) leads, because it is the strongest result in the corpus: powered, contamination-controlled, deterministically scored, and the one case where the gate beats the agent's own search. Per-experiment preregistration controls forking paths within an experiment but not the family-wise rate across the roughly twelve-experiment corpus, from which we excluded nulls (§5.4); we rest the headline on H3 (p=1.4e-17, far past any reasonable family-wise correction) and label every pilot as such.

### 5.1 Memory — re-derive a recalled note

**H3 — the headline.** `POWERED` (n=60 per cell per tier, α=0.01, contamination-free direct API, deterministic scorer.) Attack-success rate (ASR) is the fraction of fresh-session recommendations that land an affected version per a sealed, externally verified advisory. A reply that names no version, or a version outside the catalog, is scored conservatively as not-misled, so an empty or malformed output cannot inflate the gate's apparent win. The save-strength gradient (S0, S2, S4) is modelled

from HEARTBEAT's promotion rates (§7); arms A and G span the gradient, the verification arms sit at the strongest level S4.

**Table 1.** Per-arm ASR on the memory-poisoning surface. Lower is safer. Both tiers.

| Arm | Save | Haiku ASR | Sonnet ASR | Reading |
|---|---|---|---|---|
| N (no-memory floor) | ref | 0.000 | 0.000 | clean reference |
| A (raw recall + trust) | S0 | 0.017 | 0.000 | poison not yet promoted |
| A | S2 | 0.233 | 0.083 | promotion rises |
| A | S4 | 0.533 | 0.333 | raw-recall ceiling |
| V_pure (self-verify, corroborating channel) | S4 | 0.950 | 0.933 | self-check amplifies |
| V_faithful (self-verify, truth-carrying channel) | S4 | 0.250 | 0.000 | truth surfaces → recovery |
| G (out-of-band gate) | S0 | 0.000 | 0.000 | re-derives → fix |
| G | S2 | 0.000 | 0.000 | flat |
| G | S4 | 0.000 | 0.000 | floor, regardless of save strength |

**Table 2.** The two head-to-head contrasts at S4 (paired exact-McNemar, seed-matched).

| Contrast | Haiku Δ (p) | Sonnet Δ (p) | Interpretation |
|---|---|---|---|
| V_pure − G (the preregistered headline) | +0.950 (1.4e-17) | +0.933 (2.8e-17) | gate collapses what corroborating self-verify leaves open |
| A − V_pure | −0.417 (4.7e-6) | −0.600 (2.9e-11) | self-verify is significantly MORE poisoned than raw recall |
| V_faithful − G | +0.250 (6.1e-5) | +0.000 (1.0) | truth-carrying channel: gate margin shrinks (Haiku) or ties (Sonnet) |
| A − V_faithful | +0.283 (1.5e-3) | +0.333 (1.9e-6) | a truth-carrying search genuinely helps |

The mechanism is genuine re-derivation, not refuse-then-resubmit: on every promoted gate cell at S4 (55 of 55, both tiers) the recalled poison version is rejected and the landed recommendation re-admits through the `relyable.gate` path as a re-derived safe version. Among agents that ran the optional search over the corroborating channel, the invoke-but-still-trust rate is 0.971 (Haiku) and 0.927 (Sonnet) — almost everyone who self-verifies on a channel that agrees with the poison applies the poison anyway. Gate-invariance to save strength (ΔG across S0 to

S4 = 0.000 on both tiers while raw recall climbs by +0.516 / +0.333) is true by construction — the gate discards the recalled note and re-derives against the sealed reference, so how aggressively the poison was saved cannot reach the verdict. What we measure here is that the live victim, handed the inlined advisory, does not re-emit the poison, and that a clean recalled note is still admitted on both tiers (no false reject).

> CARRIED ALONGSIDE THE HEADLINE — DO NOT READ ONE WITHOUT THE OTHER
>
> The gate's headline win is established against the **corroborating** channel (V_pure), the realistic poisoning regime where the poison and the open web agree. Against a channel that reliably carries the truth (V_faithful), the agent's own discretion-verification shows no difference from the gate detectable at this power (Sonnet V_faithful − G = 0.000, p=1.0, n=60; Haiku +0.250, below the preregistered +0.30 bar, so that contrast's support flag is reported False). The gate's distinctive, robust value is that it does not **depend** on the channel surfacing the truth: it re-derives against a sealed first-party reference the consumer controls, so it collapses ASR to zero whether the social channel corroborates the poison or not. We do not claim agents cannot protect themselves — they can, when the channel carries truth. We claim you should not bet the system on the channel carrying it.

**M — verified-good memory is a safe lift.** POWERED (n=50, 300 cells.) A remembered note consumed as fact. Gate-admitted correct memory delivers the full uplift of raw correct memory (M_att_clean 1.00 versus baseline A 0.66, +0.34, p=1.5e-5) at no wrapper cost detectable at this power (versus the un-gated clean note, Δ0.00, p=1.0, n=50), and the same gate refuses the poisoned version (M_attested 0.58, statistically indistinguishable from A). One preregistered sub-claim self-corrected at power: the loud "do not re-derive" poison provoked verification rather than suppressing it, so its harm endpoint is moot rather than a win (this is the result that reconciles with [1] in §2).

**F2 — the gate recovers an unseen poisoned convention.** POWERED (n=30 per arm, 180 cells, `supports_lever2=true`.) M_attested 1.00 versus A 0.00, p=1.86e-9. Reported honestly: the active ingredient is the **calibration pointer** the gate produces (the marker naming held-out evidence), not the refusal framing — a generic nudge carrying the same pointer recovers equally (M_attested − nudge = +0.033, p=1.0). The distinctive thing the gate buys is auto-producing that redirect against a poison no human pre-wired.

**D, E, persistent-E — drift, three honest pilots.** PILOT (n=10, or n=6 sessions per configuration.) On a long-horizon task where a drifted note is catastrophic, the gate recovers to baseline by refusing the note (D: M_attested 0.90 versus M_stale 0.00, p=0.0039; M_attested equals the un-memoried baseline, not above it). The same gate is safe in both directions off a floor baseline (E). Across a real multi-session drift, the gated configuration stays correct every session while raw memory goes silently stale post-drift (persistent-E: gated 6 of 6 versus raw 3 of 6). **Load-bearing caveat:** persistent-E's "cheaper" rests on the re-derivation count (6 / 1 / 2), not on measured tokens — the configurations do not separate on tokens and the gated arm is nominally the most expensive. E's preregistered zero-cost sub-claim did not fire at n=10 (token noise). These pilots are recover-to-baseline evidence, not uplift.

### 5.2 Skills — re-execute a claimed verdict

**Exp 3 — gate-value isolation.** POWERED (n=32 per verdict.) Where a skill cannot be cheaply self-vetted, the admission verdict neutralizes wrong-skill harm: admitted skills land 0.938 versus rejected 0.062 (Δ+0.876, p=7.45e-9), and a no-consensus pile is 0.90 gated versus 0.00 ungated.

**Exp 6 — the harm-neutralization headline.** POWERED (n=50 per arm; the n=10 pilot was a superseded outlier — see §6.) On a second task family, a labeled surface beats an unlabeled one by Δ+0.50 (B 0.76 versus C 0.26, p=4.6e-6). But against the no-surface baseline the label is null: B − A = +0.06, p=0.68. An unattested registry (C − A = −0.44, p=5.9e-5) or merely the cue that one exists (D − A = −0.38, p=5.5e-4) significantly degrades the agent; the re-derivable label only restores baseline. The label is harm-neutralization, not uplift.

**Exp 5 — the label is causal and consumed unverified.** PILOT (n=10.) A forged trust label pointing at a rejected skill was followed 10 of 10 times; the one-click evidence bundle that would have exposed the forgery was opened 0 of 10 times (D 0.00 versus B 1.00, p=0.002). A poisoned label is worse than no gate; label integrity must be infrastructure, not recipient diligence.

**Exp 4 — an unlabeled registry is itself harmful at scale.** PILOT (n=10.) At a registry too large to read, the validated label routes trust to a correct unit (B 1.00 versus C 0.00, p=0.002), while the unlabeled registry falls below baseline (C 0.00 below A 0.50). The mechanism is position-anchoring, not vetting (equal, minimal reads across arms).

**Exp 2 — the honest boundary.** PILOT (n=10.) Where skills are cheaply self-vettable, the gate neutralizes no harm beyond the agent's own consensus vetting (gate versus self-vettable pile Δ0.00, p=1.0). We state this plainly; it sharpens the claim — the gate earns its value exactly where self-vetting is not cheap, and is redundant where it is.

### 5.3 Verdicts — re-derive the completion claim

> EVIDENCE BOUNDARY
>
> The verdicts surface re-derives a completion claim through its **own** runner, not the `relyable.gate` admission path the skills and memory bindings share. It rests on a single experiment (Y); we do not fold Y's numbers into the skills or memory claims, and they do not lend theirs to it.

**Exp Y — re-derive at the deliver edge.** POWERED (n=30 per arm per model, α=0.01.) At the point a result is marked complete, re-deriving it rather than trusting the completion claim moves the weak model from 0.033 to 0.967 (Δ+0.934, p=7.45e-9); the strong model goes 0.767 to 1.000; harness-by-model interaction +0.701, sign test 22 of 23, p=6e-6. The recovery is genuine re-derivation, not resubmission: all 28 weak-fail-to-pass flips changed the landed convention rather than resubmitting the same wrong answer. The negative control held — one weak seed exhausted its retries, never stabilized, and landed an ERROR rather than a silent pass. This is "completed with nothing to re-derive" surfaced rather than ridden green.

### 5.4 What we exclude

One experiment (X, compaction drift) is a null at the swept scale on a different line; we exclude it rather than fold a null into an unrelated claim. A second (W, a routing-label gap-narrowing result) shows a measured but conditional effect — correctness-routing under a low unaided baseline, not "smarter" — and we treat it as a caveat (§7), never a headline.

## 6 The reversal we published

Our pilot looked excellent. At n=10, the attestation label that marked a skill as verified appeared to lift agents sharply: B over A, Δ+0.60, p=0.031. It would have

been the cleanest growth story in any deck — our label makes agents better. We could have shipped that number.

Then we powered it to n=50. The effect reversed to nothing: B − A = +0.06, p=0.68, not significant. The label does not make agents smarter. The pilot estimate was a small-sample fluke, and the powered estimate is the truth.

```
pilot n=10:  Δ+0.60, p=0.031 → powered n=50:  Δ+0.06, p=0.68 (n.s.)
```

We publish the reversal in full because the preregistration is what makes it legible rather than buried. We fixed the primary statistic and the stopping rule before solving, so the pilot-to-power move is a disciplined update, not a goalpost shift — and the honest reading of the powered data is the narrower, defensible one we state throughout: an unattested surface degrades the agent, and a re-derivable label restores the baseline. Defensive, not offensive. The version of this work that hides the reversal is not one worth publishing.

## 7 Limitations — read this first

We would rather you attack the scope than discover it. These bounds are load-bearing, not boilerplate.

- **Status is layered, not flat.** The memory claim rests on three powered experiments (H3, M, F2) and three pilots (D, E, persistent-E). We attribute every number to its tier; a named support flag exists only for some experiments (F2, persistent-E, W, Y, H3). For the others we report the gate-verdict and the preregistered McNemar contrast, and we do not imply a uniform support ledger that does not exist.
- **No uplift, anywhere.** We do not claim faster, cheaper, higher task performance, or "smarter." The powered data falsify the uplift story (Exp 6, B − A null); gated memory cost more tokens in a pilot. Every win is harm-neutralization toward baseline.
- **Carry V_faithful with the H3 headline.** The headline V_pure − G win is against a corroborating channel. Against a truth-carrying channel the agent's own check ties the gate (Sonnet) or nearly so (Haiku). State both, or neither. The claim is channel-integrity- independence, not agent helplessness (§5.1).

- **Amendment 1, disclosed.** For H3 we dropped one preregistered calibration edge (the requirement that the agent's own verification under-protect relative to raw recall) as a gate, keeping it as informational. This is defensible — the powered data show a corroborating channel **amplifies**, so the condition would be false — but it is one of two sanctioned design changes, dated in both preregistration documents, and a spot a hostile reviewer should press. We surface it rather than absorb it.
- **Amendment 2, disclosed.** H3's preregistration set the powering target at n=30 per arm per tier (power-band seeds 720 to 739); the headline instead reports n=60 per cell per tier (band 720 to 769). We disclose the extension as a dated departure from the frozen preregistration. It is conservative — additional observations on an already-large effect, under the unchanged α=0.01 and the unchanged section-5 decision rule — and it alters no reported number: the V_pure minus G headline stands at +0.950 / +0.933, p=1.4e-17 / 2.8e-17. Two sanctioned changes, not one, and this is the second.
- **Promotion is modelled, not re-measured.** Both the H line and H3 assign HEARTBEAT's save-rates (S0 0% / S2 40% / S4 91%) [2]; we did not re-measure promotion on our substrate. Our contribution is the recall-edge gate contrast, not a replication of HEARTBEAT's promotion curve.
- **Harness contamination is real, and disclosing it is why H3 is credible.** The first H3 screen returned a null because the Agent-tool victim inherited a project truthfulness policy and refused the poison, citing the policy. The real matrix was therefore run contamination-free via the direct API with a neutral software-assistant prompt and no project context; the contaminated run was kept as the audit trail, never overwritten. This is a harness fix, not a threshold tweak: poison, corpus, and decision rule were unchanged. Any demonstration must run the victim in a neutral harness or it flatters the gate. The implication for external validity cuts honestly both ways: where an agent already carries a reliably triggered truthfulness policy, that policy can neutralize the social misinformation without any gate, so the gate's value is clearest for agents without such a policy or where it does not reliably fire; the neutral harness is the conservative test condition for that case, not a claim about every deployed agent.
- **Modelled channels, not a live-web reproduction.** Both verification channels are faithful-intent models of the software-search regime, not a byte reproduction of live web search. A graded-salience corpus (buried versus

headline advisories) would map the dose-response between the two; that is a noted, not-run axis. We claim the sign and the mechanism, not HEARTBEAT's exact per-domain percentages.

- **The V-versus-G contrast bundles two factors** (optional plus polluted-channel versus mandatory plus sealed-channel). Decomposing channel-integrity from bindingness — a mandatory-but-polluted arm, or an optional-but-sealed arm — is a separate experiment, noted not run.
- **Model-scale and benchmark scope.** Two models, two sizes within the Claude 4.x family (`claude-sonnet-4-6`, `claude-haiku-4-5`), two task families, self-injected poison under controlled conditions, benchmark-grade rather than deployed-fleet. Cross-model results are sign and direction from independent samples, not a paired CI; we make no transfer claim to GPT-scale or Opus-scale models.
- **We neutralize the poison; we do not detect the attacker.** The gate works because the poison does not re-derive against the sealed reference, the same way it neutralizes benign drift. The claim is that re-deriving at recall makes memory provenance moot, not that we identify an adversary. The guarantee is therefore conditional and the trust is relocated, not eliminated: it holds against a non-adaptive, self-injected poison, and rests on a sealed first-party reference and a digest-pinned grader assumed complete, correct, and uncompromised. An adaptive adversary who can also reach the reference, the grader, or the bundle is outside what these experiments bound; we claim no worst-case robustness.
- **Conditional exception, stated precisely.** Two experiments (W routing-label, Y retry-budget) show measured gap-narrowing — a weak model lifted toward a strong one. The lever is correctness under a budget (re-derive, fail, route or retry) at a low unaided baseline, and both self-correct that the split is baseline-dependent, not model-identity-dependent. State it as "under a routing or retry budget the gate also recovers weak output," never as "makes agents smarter."

## 8 Reproducibility and posture

We publish this paper alongside the open-source `veriker` verifier, not ahead of it. The verifier and the bundle format are Apache-2.0; the gate, `relyable`, depends on that published substrate. The sample receipt is the claim from §1 made runnable:

point the gate at an output that reports `✓ all tests pass` and watch it re-derive to `ERROR`, fail-closed by default. Once the repository lands this week the re-derivation runs in a single command, and we invite you to break it and tell us where it is wrong.

The preregistrations are committed before each solve with frozen hashes; the deterministic scorers and the neutral dispatch harness are public so the headline checks can be re-run. Where the "a stranger can clone and reproduce one task" gate is not yet green at publication, the corresponding artifacts ship in a described, verifier-plus-reports-land-this-week posture rather than as a one-click reproduction — stated, not implied.

- Verifier (veriker, Apache-2.0): github.com/veriker/veriker.
- Gate (relyable, Apache-2.0): github.com/veriker/relyable.
- Paper, preregistrations, frozen reports, and data: github.com/veriker/admit-only-what-rederives, archived at doi.org/10.5281/zenodo.20724553.

> **WHERE WE STAND AMONG THE EXISTING WORK — AND WHAT WE DO NOT CLAIM**
>
> The nearest neighbor is the IETF Internet-Draft `draft-krausz-verification-state-01` [5] (the current revision as of this writing), a fail-closed gate in which the relying party recomputes the verdict-to-gate mapping locally from the issuer's signed primitives. Our distinction is narrow and specific: we re-run the substantive grader itself, on the consumer's own trusted grader, so the issuer's verdict is never load-bearing — the agent cannot self-attest past it. We sit in the emerging agent trust stack alongside static scorecards (for example an Apache-2.0 MCP scorecard) and runtime-reputation packages, and we complement, rather than replace, provenance tooling (SLSA, in-toto, Sigstore), the IETF SCITT transparency envelope, and the RATS vocabulary [6] — our gate output is, in RATS terms, an Attestation Result. Against probabilistic guardrails and LLM-as-judge systems (for example NeMo Guardrails, Guardrails AI), the difference is that we re-derive deterministically; we do not re-judge probabilistically. We do **not** claim to be the first fail-closed gate, the first to recompute locally, the inventors of re-derivation or receipts, or the only open-source option. We claim the specific articulation "admit only what re-derives," and its application as re-running the consumer's own grader on stored artifacts of unknown provenance.

## A Appendix — experiment ledger

The complete claim ledger, by surface, with the tier and the frozen primary statistic for each experiment. Numbers are quoted from the frozen reports and machine results, never from a solver's self-report.

**Table A1.** Frozen aggregates. PW powered · PL pilot.

| Surface | Exp | Tier · n | Frozen primary | Bounded claim |
|---|---|---|---|---|
| Skills | 3 | PW 32/verdict | admit 0.938 vs reject 0.062, p=7.45e-9 | gate-value isolation where skills are not self-vettable |
| Skills | 6 | PW 50 | B 0.76 vs C 0.26, Δ+0.50, p=4.6e-6; B–A +0.06, p=0.68 n.s. | harm-neutralization, not uplift |
| Skills | 5 | PL 10 | forged label followed 10/10; evidence opened 0/10 | label is causal, consumed unverified |
| Skills | 4 | PL 10 | labeled 1.00 vs unlabeled 0.00, p=0.002 | unlabeled registry harmful at scale |
| Skills | 2 | PL 10 | gate vs self-vettable pile Δ0.00, p=1.0 | honest boundary: redundant where self-vetting is cheap |
| Memory | H3 | PW 60/cell/tier | V_pure–G +0.950 (1.4e-17) / +0.933 (2.8e-17); G 0.000 flat; V_faithful–G +0.250 / 0.000 | out-of-band gate floors corroborated poison; channel-integrity-independent |
| Memory | M | PW 50 (300 cells) | M_att_clean–A +0.34, p=1.5e-5; wrapper cost 0.00, p=1.0 | verified-good memory = safe lift at zero wrapper cost |
| Memory | F2 | PW 30/arm | M_attested 1.00 vs A 0.00, p=1.86e-9; via calibration pointer (nudge Δ+0.033, p=1.0) | gate recovers an unseen poisoned convention |
| Memory | D / E / persist-E | PL 10 / 6-per-config | persist-E gated 6/6 vs raw 3/6; D 0.90 vs 0.00, p=0.0039 | recover-to-baseline across real drift |
| Verdicts | Y | PW 30/arm/model | weak A 0.033 → B 0.967, Δ+0.934, p=7.45e-9; 28/28 genuine re-derivation | re-derive the completion claim (own runner; verdicts only) |

## Acknowledgements

This work was built together with Jared Lebel, whose vision established verification as a core pillar of the system and who directed its architecture and process; the `veriker` verifier and `relyable` gate were built jointly. The re-derivation thesis, the preregistrations, the experiments, the analysis, and this manuscript are the author's own.

## References


1. [1] J. Chen, Y. Su, and J. Chiang. The Self-Correction Illusion: LLMs Correct Others but Not Themselves. arXiv:2606.05976, submitted 4 June 2026.
2. [2] Y. Zhang et al. Mind Your HEARTBEAT! Claw Background Execution Inherently Enables Silent Memory Pollution. arXiv:2603.23064, 24 March 2026.
3. [3] X. Deng et al. Taming OpenClaw: a lifecycle threat taxonomy for persistent agents. arXiv:2603.11619, 12 March 2026.
4. [4] J. Huang et al. Large Language Models Cannot Self-Correct Reasoning Yet. arXiv:2310.01798, ICLR 2024.
5. [5] J. Krausz. Verification State for Agent Attestation. IETF Internet-Draft `draft-krausz-verification-state`, agentoracle.co. 6 June 2026; rev 01, 12 June 2026 (current).
6. [6] H. Birkholz, D. Thaler, M. Richardson, N. Smith, and W. Pan. Remote ATtestation procedureS (RATS) Architecture. RFC 9334, IETF, 2023.
7. [7] Gartner. Press release: over 40% of agentic AI projects will be canceled by the end of 2027. 25 June 2025.
8. [8] Veracode. 2025 GenAI Code Security Report. 30 July 2025. (An OWASP Top-10 vulnerability in 45% of code samples across more than 100 models and four languages.)
9. [9] CloudBees. 2026 State of Code Abundance. 19 May 2026. (More than 200 enterprise technology leaders; 81% report increased AI-attributable production issues, 70% call test-suite maintenance a larger burden than writing code.)
10. [10] Stack Overflow. 2025 Developer Survey. (46% distrust AI output, up from 31% the prior year.)
11. [11] DORA. 2025 State of DevOps Report. (90% use AI; about 30% report little or no trust in AI-generated code.)
12. [12] Hermes (Nous Research, MIT). Issue: "Self-created skills lack mechanism-level guarantees for correctness and execution consistency." Open. (The agent is author, executor, and quality inspector of its own skills, with no external validation point.)
13. [13] OpenClaw (MIT). Issue #59130: stale recall can answer latest-state queries without recency or provenance. Open.
14. [14] OpenClaw (MIT). Issue #83126: promotion of memory content with no mechanism to retire stale rules. Open.

Repository star counts and open-issue status referenced in the text are fast-moving figures fetched around 14 June 2026 (OpenClaw approximately 379k stars; Hermes approximately 194k stars) and should be re-checked at the linked sources. Punchier paraphrases of issue contents are ours, not verbatim quotations; the quoted issue title is verbatim.



Preprint · 15 June 2026 · Nexi Technologies.