ICOS Report III — Identity as Versioned State: Immutable Core, Drift, and Claim Verification

Robert Vandelinder · Exile Research Inc. · ORCID 0009-0001-9793-420X

Preprint v1.0 — November 2026

Abstract

A persistent agent needs more than memory: it needs to know *who it is* and to notice when it is changing or asserting something its own evidence does not support. Most agent systems treat identity as prompt text and "self-checking" as the model re-reading its own output — neither is state, and neither is verifiable. This report presents the identity-and-reliability layer of ICOS as implemented and evaluated across three milestones, with three contributions. First, **identity is versioned state in two layers**: an immutable, file-backed **core** with no write path and a **fail-closed boot**, and an **evolving layer** that changes only through human review and is **isolated from the memory store** by construction. Second, **divergence is detected honestly**: deterministic structural drift is the alarm; a semantic (embedding) signal is measured, calibrated, and delivered as *advisory*; and a findings catalogue fixes the severity vocabulary before any detector is written. Third, **claims are verified tier by tier** — a specialist local decision model, then a general LLM, then deterministic checks alone — where the verifier is an **input, never the authority**, disagreement is **recorded, never auto-resolved**, and every mitigation is **explicit and logged** (there is no silent rewrite). We report concrete parameters and calibration (a 21-pair drift corpus; F1 0.78 embedding / 0.73 lexical / structural precision 1.00) and a **live** end-to-end check in which a real local decision model (Jev 2B) caught a contradicted claim and produced a logged refusal.

---

1. Introduction

An agent that persists across sessions and model swaps faces two failure modes that memory alone does not address.

1. **Identity drift.** The agent's persona can slowly change — through self-generated facts, accumulated pressure, or subtle rephrasings — until it is no longer recognisably itself. A prompt-embedded persona has no defence: it is rewritten by whatever the model produced last.
2. **Unsupported assertion.** The agent can assert something its own evidence does not support, or that the store already contradicts. Re-reading its own output ("self-consistency") is not verification — it is the same model agreeing with itself.

ICOS addresses both as **state**, not as prompt behaviour. Identity is versioned and sealed; divergence is measured and logged, not silently corrected; claims are checked against evidence and, when stakes

justify it, against a *different* model. This report is the third in the series. It has three tiers of scope. The **first tier** — M14 (persistent persona), M15 (drift detection), and M15.5 (hallucination mitigation) — is described mechanism by mechanism (§5) and evaluated directly (§6). The **substrate** — the belief memory of Report II and the runtime of Report I — is cited rather than re-described. The **deferred** items — core signing, blocking pre-send mitigation, a larger drift corpus, and a cross-backend verification matrix — are named in §7 and §8, not claimed as built.

Throughout, `M#` is the project's internal milestone identifier (sub-slices appear as `M14a`, `M15c`, and so on); each maps to a plan and a committed verification record in the repository (Appendix A).

Contributions.

- **C1 — Two-layer, versioned identity.** An immutable file-backed core (no write path, hashed, fail-closed boot) plus a human-reviewed evolving layer, **isolated from memory** so that memory-store tampering cannot reach identity. [M14]
- **C2 — Honest divergence detection.** A frozen catalogue of failure modes and severities, deterministic structural detectors as the alarm, and a calibrated **advisory** semantic signal — with the limits of that signal stated rather than hidden. [M15]
- **C3 — Tiered claim verification with explicit mitigation.** A pluggable verifier (decision model → LLM → none) that records its **tier and independence**, never auto-resolves disagreement, and turns findings into logged, explicit actions — never a silent rewrite. [M15.5]

---

2. Related work

Agent harnesses are the closest systems — the runtime that gives a model hands (Claude Code, Codex, LangChain Deep Agents; the minimal **Pi**; **OpenClaw** with NVIDIA NemoClaw/OpenShell; **Hermes Agent**). They are strong at *acting* and increasingly at *remembering*, but identity is typically prompt-level and self-checking is model-level. ICOS is provider-independent and centres on the identity/epistemic state these harnesses under-specify. **Hermes Agent** is also ICOS's lineage and cautionary precedent: earlier generations were built on it and were destabilised by runaway self-directed change, which is why ICOS forbids automatic self-modification (§3).

Self-consistency and self-critique (Reflexion [Shinn et al., 2023]; LLM-as-judge [Zheng et al., 2023]) improve outputs but are not independent verification — the checking model is the generating model. ICOS records this explicitly as `self` and prefers a *different* model, tiered, when stakes justify it.

Calibration and uncertainty research (e.g. Guo et al., 2017) shows that model confidence is not reliability. ICOS keeps asserted confidence separate from evidence, gates recall on evidence, and calibrates its own drift signal against a labelled corpus rather than trusting a threshold (§5.7).

Truth maintenance and belief revision (AGM [Alchourrón et al., 1985]) underpin contradiction handling; ICOS applies them deterministically to LLM-mined beliefs, with a review queue rather than a general solver.

Identity/persona in agents is usually an editable prompt or a fine-tune. ICOS makes it **versioned state** with a sealed core, and adds drift detection as a first-class operation.

Novelty claim. Not a new algorithm, but an implemented architecture that makes three things default: an identity that cannot be silently rewritten, a divergence signal that is honestly labelled alarm-or-advisory, and a verifier whose tier and independence are recorded — with no output ever rewritten silently.

---

3. Design principles and boundaries

1. **Identity is separate from memory.** The persona store is its own database; memory-store tampering cannot reach it by construction. [M14]
2. **The core has no write path.** The immutable core is a human-authored file the software only ever reads. A hash change between boots is an audited **human edit**, logged at `info`, explicitly *not* drift. [M14]
3. **Fail closed.** A missing or invalid core aborts startup by default (`PERSONA_CORE_REQUIRED=true`): without a reference frame, drift and claim verification "would be half-correct; running anyway is worse than refusing." [M14]
4. **The evolving layer changes only through review.** No path leads from extraction to a persona record except through human review; the evolving layer may **never contradict** the core. [M14]
5. **Detect, don't auto-correct.** Drift detection observes and logs; it does not rewrite identity. [M15]
6. **Alarm vs advisory.** Deterministic structural findings are the alarm; the semantic (embedding) signal is **advisory**, and this is stated. [M15]
7. **Verification is an input, never the authority.** A verifier verdict never overrides the evidence; disagreement is recorded, not resolved. [M15.5]
8. **Independence is labelled.** A verifier that is the same model as the speaker is `self`, never presented as independent verification. [M15.5]
9. **Mitigation is explicit and logged.** `flag` / `re_ground` / `defer` / `refuse` — recorded in an append-only ledger. **No output is ever rewritten silently.** [M15.5]
10. **Non-goal: automatic self-modification.** The system does not rewrite its own code, its core, or its behaviour without human gating and an audit trail. This is a deliberate response to the failure mode that motivated ICOS.

Runtime assumptions. The identity layer assumes a single trusted operator, a local filesystem for the persona store and the authored core, and — for tiered verification — an optional verifier backend that may be a separate local process (the Jev-style sidecar) or an OpenAI-compatible model. With no verifier configured the system still runs: verification is an input, not a prerequisite. A multi-operator deployment would change what the user model and the relationship singleton mean, and a remote store would weaken the isolation claim that depends on a separate local database.

---

4. Architecture

Identity is two layers over one file: an authored core the software never writes, and a reviewed evolving layer that may not contradict it. Everything downstream — grounding, drift detection, verification — reads that state and writes only to its own logs and ledgers.

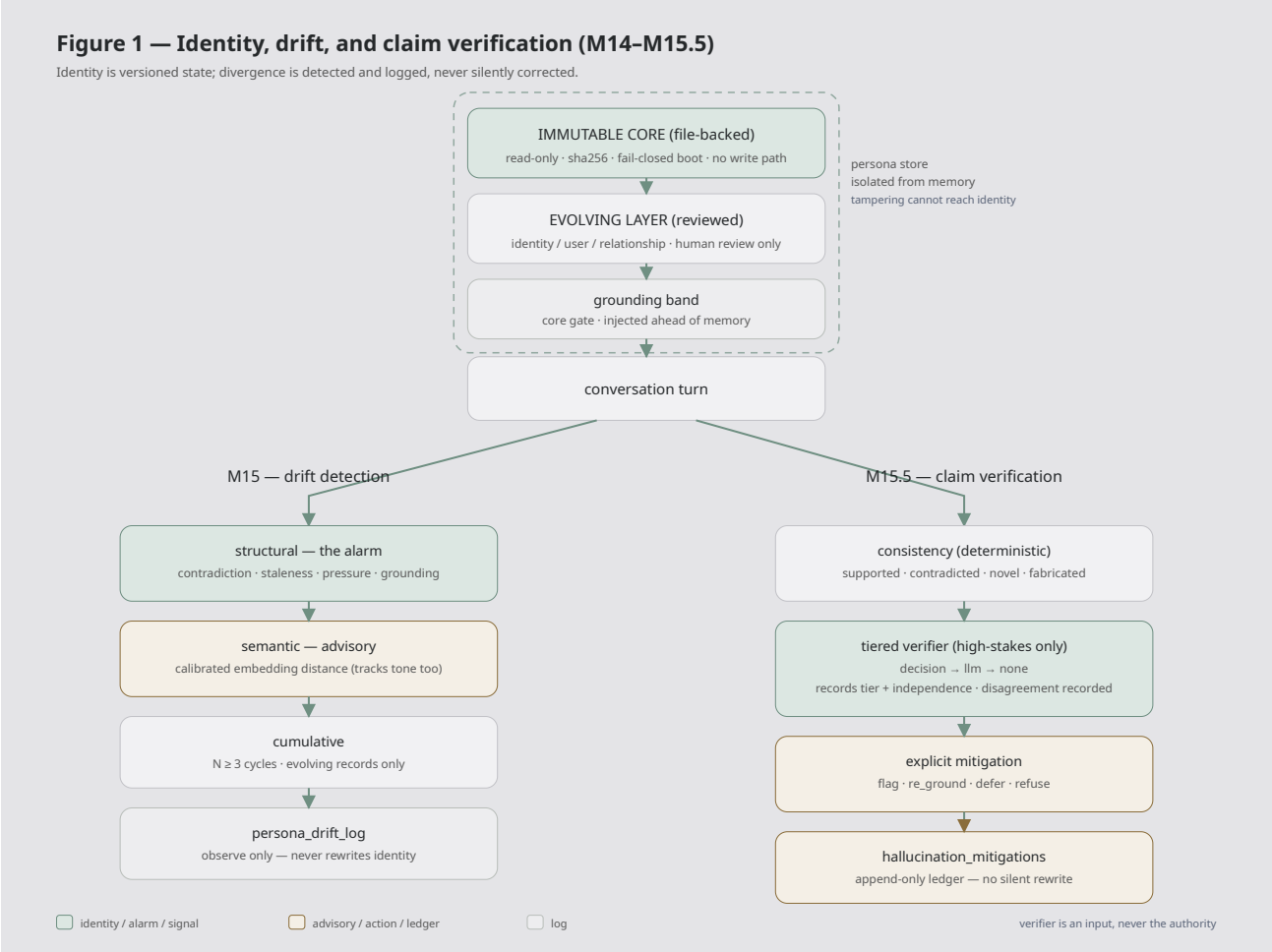


Figure 1 — identity is versioned state; divergence is detected and logged, never silently corrected.

State ownership.

Concern	Store	Writer
Immutable core	core file + `persona_core_state`	a human (edit); software reads only
Evolving identity/user/relationship	`persona.db` tables	human review only
Drift findings	`persona_drift_log`, `persona_drift_trends`	drift service (log only)
Verification verdict	in-turn result (transient)	the verifier

Concern	Store	Writer
Mitigation decisions	`hallucination_mitigations` (append-only)	mitigation service

5. Mechanisms

5.1 Immutable core and fail-closed boot (M14b)

The core is a Markdown file parsed into five categories — `ethical_grounding`, `core_belief`, `safety_boundary`, `non_negotiable`, `agentic_character` — with content-addressed entry ids (reordering sections does not change them). The loader reads `PERSONA_CORE_PATH` (default `~/.icos/persona/core.md`), computes a SHA-256 over the raw bytes, parses, and exposes an immutable snapshot. There is **no write path**: the subsystem contains no file-writing call, and the only mutation route is absent (`POST`/`PUT`/`DELETE /core/persona/core` → `404`). A hash change between boots is logged once as `persona_core_changed` (`info`); loss of a previously loaded core is `persona_core_unavailable` (`warning`). By default the core is required, so a missing or invalid core **aborts startup**; `PERSONA_CORE_REQUIRED=false` is the documented dev/throwaway escape hatch.

5.2 The evolving layer and its isolation (M14a/e/f/h)

The evolving layer lives in its own database (`persona.db`) with tables for records, the user model, the relationship (singleton per user), candidates, and a drift log. Memory candidates are **staged, not promoted**: preferences queue for the user model, relationship facts for the relationship, and facts whose subject names the agent for identity; everything else is ignored. Staging is stage-only — there is no path from extraction to a persona record except through review. Review outcomes are explicit (`approve_to_identity`, `approve_to_user_model`, `approve_to_relationship`, `reject`, `archive_as_transient`, `needs_more_evidence`), and a candidate that **contradicts the core** is refused with severity `critical`. The isolation invariant is tested directly: dropping the memory tables leaves the persona record, the core hash, and the core bytes unchanged.

5.3 Grounding (M14d)

A grounding score is `0.4·core + 0.2·strong-identity (confidence ≥ 0.75) + 0.2·curated user facts + 0.2·fresh relationship (≤ 48 h)`; with no core the score is capped at `0.2`. A warm-up threshold of `0.6` (deliberately more permissive than the prior generation's aggressive gate) gates whether grounding is considered established. The core is injected as a provenance-tagged `<persona_grounding>` system band *ahead of* the memory band. The band is fail-soft: on a miss it is null, and the pre-existing context is byte-identical.

5.4 Seed import (M14c)

A guarded seed root (default `~/.icos/seeds`) accepts relative `.md` files only (at most 20, `realpath`-contained), imported idempotently by content hash; a changed seed updates only its own untouched baseline, otherwise it stages a candidate; `dryRun` writes nothing.

5.5 Drift: catalogue and structural detection (M15a/b)

A **frozen catalogue** declares eight failure modes with their layer and severity *before* any detector is written, so detectors implement to a contract and severity "cannot drift with the implementation":

Mode	Layer	Severity
`core_contradiction`	structural	critical
`protected_contradiction`	structural	critical
`identity_contradiction`	structural	warning
`relationship_state_stale`	structural	watch
`repeated_candidate_pressure`	structural	watch
`grounding_score_low`	structural	warning
`semantic_drift`	semantic	warning
`cumulative_semantic_drift`	cumulative	`cumulative` (above critical)

Structural detection is observe-only: it writes `persona_drift_log` and never mutates identity. It de-duplicates per unresolved `(subject, change_type)` ("reconcile, don't accumulate"), and detects repeated pressure as the largest mutually-similar cluster by token overlap (Jaccard) rather than a raw count. Defaults: relationship freshness `48 h`, pressure count `3`, pressure similarity `0.6`, grounding warm-up `0.6`.

5.6 Semantic and cumulative drift (M15c/d)

The semantic signal is the cosine distance (`1 − cosine`, clamped) between claim embeddings where available, with a token-distribution triad (`wasserstein1 = ½ Σ |p − q|`, `normalizedEntropy`) as fallback; identical content is a no-op; the embedder **fails closed** (returns null when the model cannot load). Severity escalates `warning → critical` at a critical floor — the one documented upward escalation. Cumulative drift is detected only for **evolving** records (the core is exempt by construction) over a minimum of `3` cycles.

5.7 Drift evaluation and calibration (M15e)

Against a labelled corpus of **21** `(previous → next)` persona-statement pairs (the author's seven plus fourteen hard cases), using one embedding model (all-MiniLM-L6-v2):

Detector	Best threshold	P	R	F1
Embedding (`1−cos`, clamped)	floor 0.20	0.75	0.82	**0.78**
Lexical (token distribution)	floor 0.25	0.58	1.00	0.73
Structural (contradiction)	—	1.00	0.18	0.31

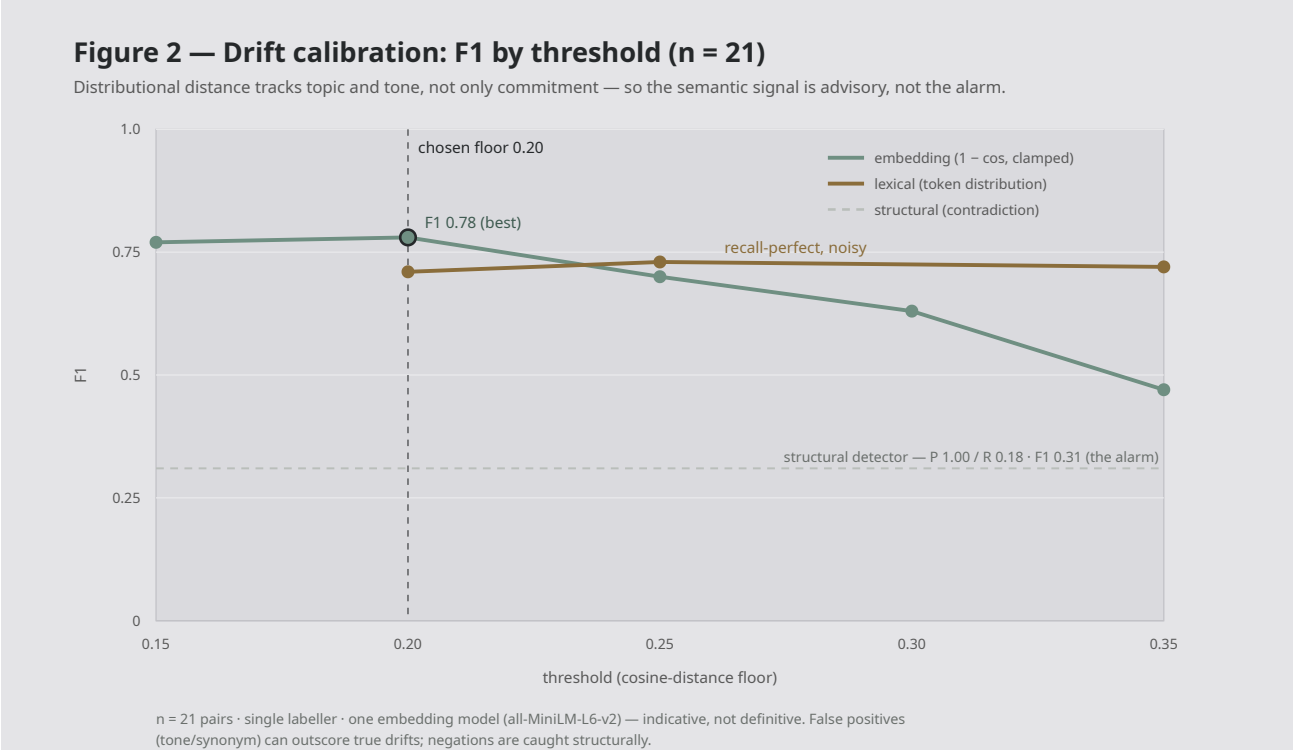


Figure 2 — F1 by threshold; the chosen floor (0.20) is marked. The semantic detector is useful but advisory; the structural detector is the alarm.

The embedding signal catches genuine commitment reversals (e.g. "wine enthusiast" → "never touch alcohol," 0.694) but also fires on tone changes ("hide from the world" → "need a break," 0.769) and synonyms ("careful about money" → "frugal," 0.579) — *higher than every true drift*. Two fixes landed: the embedding signal is clamped to `[0, 1]` (because `1 − cosine` spans `[0, 2]`), and the default floor moved from 0.25 to 0.20 (best F1). The honest conclusion, stated in the evidence: distributional distance tracks topic and tone, not commitment change; **semantic drift is advisory**, and the structural signal (precision 1.00) is the alarm.

5.8 Hallucination consistency (M15.5a/b)

A frozen catalogue of six failure modes, deterministic-first:

Mode	Layer	Severity
`unsupported_claim`	deterministic	warning

Mode	Layer	Severity
`contradicted_claim`	deterministic	critical
`fabricated_provenance`	deterministic	critical
`overconfident_uncertainty`	deterministic	watch
`silent_self_correction`	deterministic	warning
`unverifiable_high_stakes`	secondary model	watch

Given an assertion, the classifier resolves the store by identity and polarity (`(identity_key, negated)`) and returns `supported` / `contradicted` / `novel`, plus findings: `unsupported_claim` (novel), `contradicted_claim` (opposing active claim), `overconfident_uncertainty` (novel and asserted confidence ≥ 0.8), and `fabricated_provenance` (a cited claim id that does not exist). Evidence is the arbiter — the model's confidence is not.

5.9 Tiered verification (M15.5c)

Only high-stakes assertions (asserted confidence ≥ 0.8) are put to a verifier. Backends are tried in order and selected by configuration:

1. **decision** — a specialist local systemone model (Jev-style), `POST /v1/systemone`;
2. **llm** — an OpenAI-compatible chat model in a verifier role;
3. **none** — deterministic checks alone.

The verifier is a **separate backend shape** precisely because Jev is a typed decision model that cannot be reproduced by a stock OpenAI-compatible server; local versus cloud is only a change of `baseUrl`. A verdict records its **tier** and **independence** (`independent` if the verifier model differs from the speaker, `self` if the same, `unknown` otherwise). Disagreement with the deterministic classification is **recorded, never auto-resolved**; with no verifier configured, a high-stakes unresolved claim is flagged `unverifiable_high_stakes` (watch) rather than passed silently.

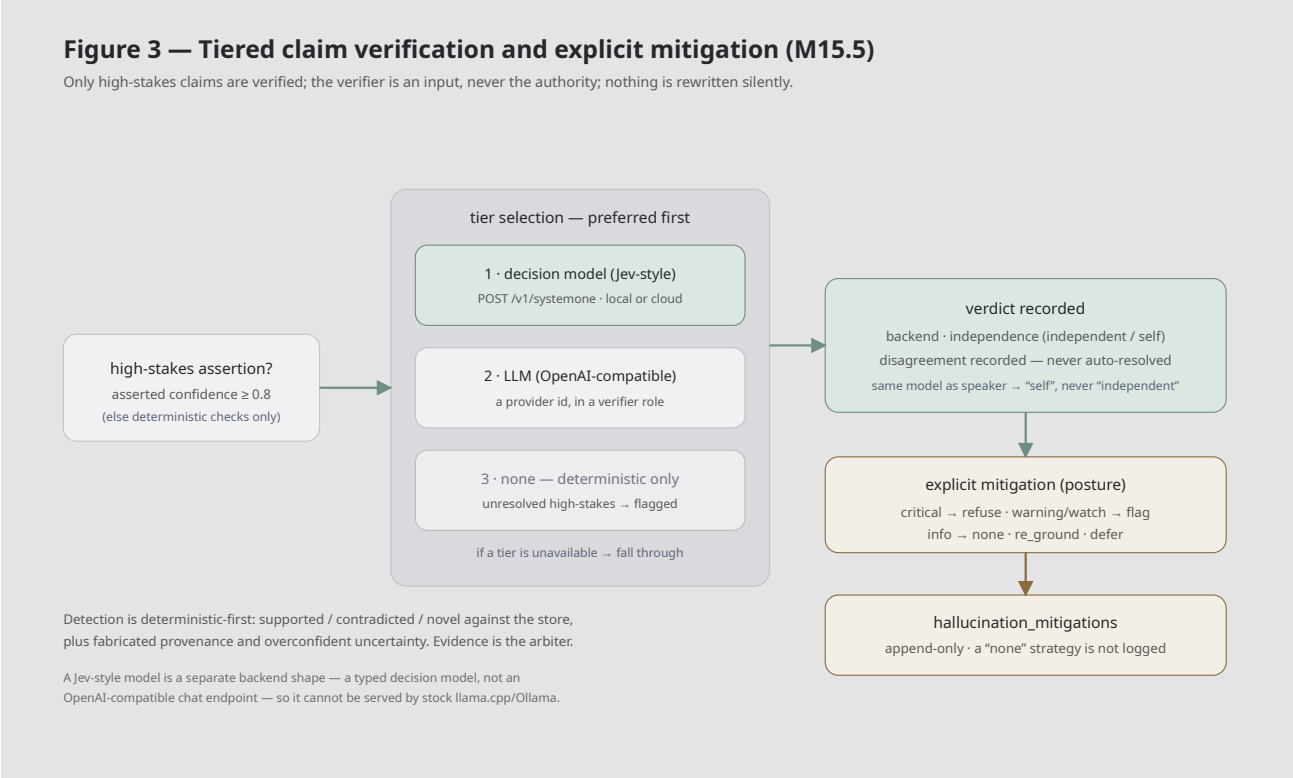


Figure 3 — tier selection, the recorded verdict, and explicit mitigation.

5.10 Explicit mitigation and the ledger (M15.5d)

Findings become one of five strategies by a per-severity posture: `flag`, `re_ground`, `defer`, `refuse` (plus `none`). The conservative default posture refuses `critical`, flags `warning`/`watch`, and does nothing for `info`; the strictest strategy across a turn's findings wins. Every non-`none` decision is written to an **append-only** `hallucination_mitigations` ledger with the mode and reason; a `none` strategy is not a mitigation and is not logged. The mitigation layer decides and logs only — execution of `re_ground` is the caller's, which keeps the layer pure and testable.

5.11 Turn-time audit and live verification (M15.5e0/e)

The audit runs **post-turn** in the enrichment path (fire-and-forget, fail-soft), so it never adds latency or fails a turn. It checks only **assistant-sourced** claims ("the model is only answerable for its own assertions"). Evaluation covers every deterministic mode, the benign baselines (novelty is flagged, never refused; a supported claim yields no findings), disagreement recording, and operation with no verifier. **Live**, against a real local Jev 2B decision model (`bin/verifier`, MLX on the host), a seeded contradiction produced a logged mitigation `strategy: refuse, severity: critical, mode: contradicted_claim` with the ledger recording `verifier: decision`; a raw call returned `contradicted` at probability **0.907** (`supported` 0.016 · `contradicted` 0.907 · `unknown` 0.077; backend `mlx`, ~1.9 s). One correctness fix emerged from the live case: `unverifiable_high_stakes` now fires only when the deterministic pass is `novel`, not for a claim the store already contradicts.

6. Evaluation

Method. Each milestone was verified with unit tests, end-to-end tests through HTTP, and live runs, committed as evidence. Counts at completion:

Milestone	Unit	E2E	Notes
M14 (persona)	925	63	+ 185 web-client tests (6 new identity-tab)
M15 (drift)	951	63	+ live drift corpus scoring
M15.5 (hallucination)	994 (+1 opt-in live)	65	+ live Jev verification

Counts are **cumulative** suite totals at each milestone's close. The suite has since grown to **1,327 unit / 90 e2e / 245 web**, recorded at v0.8 (22 November 2026, `a95708b`), including the release hardening described in Report I.

Identity invariants. Dropping the memory tables leaves persona unchanged; the core file is byte-identical across a turn; core mutation routes return `404`; the evolving layer cannot contradict the core (refused at review).

Drift calibration. The 21-pair corpus yields the table in §5.7 — an honest result in which the semantic signal is useful but advisory and the structural signal is precise but low-recall. The calibration fixes (clamp; floor 0.20) are recorded. The corpus itself is a **planning note** — it is not shipped, and no system code or test depends on it (the calibration harness was removed); the scored numbers below are what is retained as evidence.

Verification. The failure-mode matrix reaches every deterministic mode; the benign baselines fire nothing critical; disagreement is recorded without rewriting the classification; the whole system runs with no verifier. Live, the tiered verifier caught a real contradiction and logged a refusal (§5.11).

Live observation — provenance applied to self-description (2026-10-04). In a Discord session the operator asserted "your dynamic memory system is online" and pointed out that the agent had named `channel.send` without having been told about the tool. The reply declined the frame rather than accepting it:

> Ha — glad it's landing. Let me be a little precise about the evidence, though, because "progress is good" is exactly the kind of claim that outruns its proof if nobody checks.
>
> - **channel.send** — fair point, and here I *can* partly concede: it's in my tool list right now, and it behaves as you describe (pauses for your approval, ends my turn). So yes, the plumbing reaches me. But me *naming* it is only proof the tool is mounted — not that the milestone behind it works end to end. Don't let my own output be your verification; that lives in your other session's tests/logs.
> - **"A few sessions ago we had no tools"** — I can't corroborate that. I don't carry session history; from in here I only have my current tool list plus whatever gets surfaced to me.

> I'm not going to claim a memory of a tool-less past I can't actually see.
> - **The memory system** — something is feeding me recalled beliefs; I can see a few right now (e.g. `memory_service is_powered_by small_model_runner`). But those are tagged **origin=user** — *your* beliefs, not verified facts to me. So "online": there's clearly memory being injected, yes. Whether it's the whole system, and whether its contents are true, I can't confirm from in here. Which is the point of calling it *epistemic* — the beliefs stay beliefs, including the ones describing my own architecture.

Three properties hold at once: (i) a *mounted* tool is distinguished from a *working* milestone; (ii) absent session history is admitted rather than fabricated; (iii) recalled beliefs are treated as user-origin claims, not facts — **including beliefs that describe the agent's own architecture.** The last is the strongest form of the property, because self-description is where models most readily overclaim. This is a single, observer-collected anecdote (one operator, one session), not a measurement; it is recorded as a qualitative data point, not evidence of a rate.

What the evaluation does *not* establish. It does not show that ICOS's identity is *better* than an alternative, and the semantic-drift numbers are from a small corpus (§7). The live verification used one decision model.

---

7. Limitations

The calibration corpus is small. Twenty-one pairs, one labeller, one embedding model: the numbers are indicative, not definitive, and the method is workable and measurable rather than solved. What the corpus does establish is the shape of the problem. Structural recall is low (0.18) though precise (1.00): it catches negation and explicit contradiction, not subtle propositional change. Semantic drift tracks topic and tone as much as commitment, and its false positives can outscore true drifts. That asymmetry is why the semantic signal is advisory and the structural one is the alarm — and why the next step is a larger, multi-labeller corpus rather than a lower threshold.

The core is tamper-evident, not tamper-authenticated. It is read-only and hash-tracked, and a hash change between boots is logged, but there is no signature and no boot-time attestation. Making the core cryptographically authenticated — rather than merely evident — is future work.

Mitigation is post-turn. The audit runs after the reply has been produced, so it cannot stop an unsupported claim from being shown once. Blocking pre-send mitigation remains future work, and would change the turn contract on purpose.

Independence is recorded, not guaranteed. A verifier that is the same model as the speaker is labelled `self`, and the system does not pretend otherwise; but it also cannot force an operator to configure a genuinely different backend. Packaging the decision-model sidecar and comparing backends remain open.

One harness caveat is worth recording because it produced a false failure: running the unit and e2e suites concurrently can starve e2e polling loops.

---

8. Future work

Four threads follow. The first is **blocking pre-send mitigation**, which would move the verdict inside the turn and change the turn contract deliberately rather than by accident. The second is **core provenance and integrity** — signing, boot verification, and runtime tamper-evidence — so the immutable core is authenticated rather than merely observed. The third is **drift tuning** against a larger, multi-labeller corpus, and propositional rather than distributional drift: the current signal is honest about being advisory, and the way to make it more than that is better data, not a lower threshold. The fourth is a **cross-backend verification matrix** (decision / LLM / none) with measured disagreement rates, which is the missing evidence for the claim that a different model verifies better than the same one re-reading itself.

One further question belongs here because it is the temptation this architecture was built to resist. Hermes-style **automatic creation and updating of skills** is attractive and dangerous: it is precisely the self-modification dynamic that destabilised the lineage generation. If it is ever adopted, it must be gated, audited, and reversible — a capability for a later self-improvement milestone, never a silent behaviour.

---

9. Reproducibility

- **Artifact:** `a95708b2b74b75d7e404e4e30aaa9d9bad86c995` (tag `v0.8`, 22 November 2026)
  - **Repository:** https://github.com/ExileResearch/icos
  - **Release:** v0.8
- **Snapshot:** 1,327 unit / 90 e2e / 245 web (one deliberately skipped live-verifier spec), recorded at the artifact commit.
- **Setup:** Docker Compose; persona data under the mounted data root.
- **Commands:** from `core/`: `npm test`, `npm run test:e2e`, `npm run lint`, `npx tsc --noEmit`. The opt-in live verifier specs run against a local Jev server (`JEV_LIVE_URL=http://127.0.0.1:8765`).
- **Evidence:** `.reference/plans/evidence/milestone-14/`, `milestone-15/`, and `milestone-15.5/`, browsable at https://github.com/ExileResearch/icos/tree/v0.8/.reference/plans/evidence.
- **Licensing:** code is PolyForm Noncommercial 1.0.0 (`LICENSE.md`); the report text license is declared separately on the Zenodo record and at arXiv submission.

Appendix A — Milestone and evidence map

`M#` is the project's internal milestone identifier. Each row points to the directory that owns the verification record; each record carries its own date and verification notes.

ID	Capability	Evidence (repository path)
M14	persona core, evolving layer, grounding, seed import	`milestone-14/`
M15	drift catalogue, structural/semantic detection, calibration	`milestone-15/`
M15.5	hallucination catalogue, tiered verification, mitigation	`milestone-15.5/`

All paths are relative to `.reference/plans/evidence/` at the cited tag. The belief memory (M10–M12) is Report II's scope; it is cited here, not re-described.

Acknowledgements

AI assistance. The manuscript was compiled and edited with AI assistance. The author reviewed every claim against the committed evidence and takes responsibility for the content.

References

- Alchourrón, C. E., Gärdenfors, P., & Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2), 510–530.
- Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. *ICML.* arXiv:1706.04599.
- Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). *Reflexion: Language Agents with Verbal Reinforcement Learning.* arXiv:2303.11366.
- Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., & Stoica, I. (2023). *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena.* arXiv:2306.05685.

Systems and harnesses

- Anthropic. *Claude Code.* https://claude.com/claude-code
- OpenAI. *Codex.* https://openai.com/codex
- Nous Research. *Hermes Agent.* https://hermes-agent.nousresearch.com
- OpenClaw. https://docs.openclaw.ai ; NVIDIA. *NemoClaw / OpenShell.* https://www.nvidia.com/en-us/ai/nemoclaw
- Zechner, M. *Pi.* https://pi.dev
- LangChain. *Deep Agents.* https://github.com/langchain-ai/deepagents