Ario: Provenance-Aware Auditing of AI Claims

Research Summary and Proposed Studies

Sepehr Ghanbari · Independent researcher (no institutional affiliation)
ORCID: 0009-0006-4710-3914 · Correspondence: sepehrqanbari@gmail.com
Working paper, version 0.1 · 8 November 2026 · Not peer reviewed
Keywords: AI evidence provenance; source dependence; calibrated abstention; self-reports; multi-agent dialogue; auditability; reproducibility

Abstract

Ario asks whether an AI system's claims can be traced to independent evidence and corrected without erasing history. This summary reports what a read-through of the public repository supports, and proposes three studies that use large language model APIs.

Verified in this review. The repository's full test suite ran locally at commit `de6d3fc` (4 November 2026): 1127 tests passed, plus 98 subtests, in 3.73 seconds. That shows specified synthetic behavior holds. It does not show security, historical truth, or anything about machine consciousness.

Proposed, not yet run. S1 benchmarks LLM evaluators on source-dependence fixtures. S2 replicates multi-agent dialogue termination and looping under a preregistered protocol. S3 probes whether agent planners treat their own earlier outputs as independent evidence.

The author is an independent researcher with no institutional affiliation. The work is not peer reviewed.

1. Author and project at a glance

Item	Detail
Author	Sepehr Ghanbari, project architect and principal researcher
Repository	github.com/SepehrGhanbari/Ario, 197 tracked files
Reviewed commit	`de6d3fc0ee8a0116ec268edc8bb2c7c5e2bd2945` (merge of PR #143, 4 November 2026)
License	MIT for code; book texts carry their own stated terms
Website	ario-research-independent.vercel.app
Related working paper	*Evidence Laundering in Auditable AI Systems*, SSRN abstract 7675778, status PRELIMINARY_UPLOAD

2. Research problem and relevance to safety

AI systems that remember, cite and act produce records that look like evidence but may all descend from one unverified claim. The repository's own fictional examples make this concrete: a support bot proves a refund by quoting its earlier message, ten sites copy one post, an assistant says "you authorized publishing" when only a draft was authorized.

Ario separates **observation, evidence, inference, claim, assessment and revision**, and keeps UNKNOWN as a valid output rather than a failure. It does not claim that any AI system is conscious. The work touches five safety-relevant questions:

- **Provenance of agent evidence.** A matching hash, two agreeing witnesses and a passing test answer different questions; promoting one into another is what the working paper calls evidence laundering.

- **Calibrated abstention.** UNKNOWN must be scored against over-refusal, so a system that answers "unknown" to everything does not win.
- **Status of self-reports.** A system's statement about itself is a recorded claim with no special evidentiary privilege (principle 5).
- **Authorization boundaries.** A well-formed grant or signature is not an authorized action; the contracts keep admission forbidden and reviewer verification unimplemented.
- **Multi-agent dialogue behavior.** Transcripts of two model instances show closure and looping patterns that need controlled study.

3. What exists today

The repository holds about 7.5k lines of non-test Python, about 13.9k lines of test and lab Python, and design-only contracts. The full suite passed 1127 tests in this review.

Component	Status	Evidence in repo
M0 structural audit core (core/schema.py, core/audit_engine.py)	Implemented; deterministic; no LLM judge	138 tests in test_m0_regression.py; M0 acceptance record says "acceptable with explicit limitations"
Verdict-as-evidence and composition firewall (F11, F15)	Implemented on main per its contract; first recorded as not representable in M0	F11/F15 security contract
Checkpoint-prefix, witness, provenance, compromise-and-recovery experiments	Implemented over synthetic fixtures	73, 26, 58, 53 and 87 tests in their modules
Lab verifiers for custody, distributed profiles, governance motions, payload binding	Lab only	tests/w3_lab to tests/w5_lab
Ledger adapter and bridge, dossier builder, admission-review evaluator, reviewer preflight	Bounded observers; admission stays forbidden	Evidence admission contract
Evidence admission, reviewer authorization, trust anchor	Design only; no keys generated; owner signer on hold	Trust anchor contract
Bounded agent and workflow runtime, local planner	Implemented; allowlisted tools, no shell, network or delete; hash-preconditioned writes with external backups	Agent runtime

Reproduction run. On Linux with Python 3.13.18 and pytest 9.1.1, running `python -B -m pytest -p no:cacheprovider -q tests` from `implementation/` gave 1127 passed and 98 subtests passed in 3.73 s. The engineering log records the same totals for a GitHub Actions run. This is one run on one host, over tests the project wrote itself.

4. Findings so far

Every positive result in the repository is scoped to a stated predicate, and the project writes down what each result does not show.

Observed result	What it supports	What it does not establish
148 text payloads of the pinned snapshot match their digests	Bytes match hashes carried in the same export	Independent authenticity of the original repository
A locally valid rewritten chain conflicts with a frozen checkpoint	A discrepancy against a retained reference	Which party is honest, if the reference itself is untrusted

Observed result	What it supports	What it does not establish
Two synthetic witnesses agree	Equivalent commitments under a defined scope	Real custody independence, or that the events happened
Provenance graph reports conditional separation	Disjointness within the supplied, complete graph	A complete inventory of real administrative control
M0 structural PASS	Declared structure meets the deterministic rules	Correct LLM extraction, a true claim, any ontological conclusion
280 targeted tests pass (pinned snapshot)	The checked assertions held in that run	Comprehensive security or deployed reliability

Sources: working paper, section 5.2 and the M0 acceptance record, sections 6 to 9.

The project also records its own defects. During M0 the evidence-inflation rule (F12) wrongly flagged a mixed independent and dependent case; it was narrowed and re-tested. Four attack classes were recorded as "not representable" instead of "defended": similarity-only lineage, hidden transformation, verdict-as-evidence and global truth composition.

5. Dialogue observations (A11, A12, A13)

Three transcript-derived runs of two model instances show closure and looping behavior, but they support no causal claim.

Trial	Messages	Observation	Caveat
A11/v5	18	Turn 18 contains the end marker	Both participants were told the marker
A12	13	Natural goodbye at turn 12, marker at turn 13	Prompt discussed stopping; a provider pause preceded recovery of the same trial
A13	50	Argument develops to about turn 20, then about 30 turns of reciprocal closure	No stop instruction; reported length-limit pauses and continuations

In A13 one participant proposed concluding at turn 46, yet four more replies followed. The three trials differ in prompt, history and recovery, so they must not be pooled. The files are user-supplied exports with checked hashes, not provider attestations, and raw API traffic was not audited. The published summaries do not name the model or provider.

The repository's own next step, marked not executed, is an explicit and respectful termination signal that is not itself the topic. It would add role and turn validation, loop detection and preregistered recovery boundaries. Study S2 builds on it. Source: A11 to A13 results.

6. Open problems the project cannot resolve alone

The first, second and fifth items below come from the project's own critical review, which keeps them open. The third and fourth are gaps this summary adds.

- **Reality contact.** Auditing internal records can raise coherence without raising correspondence to the world. The review asks whether the framework improves contact with reality or only its model of itself.
- **Structural versus semantic support.** A structural pass says nothing about whether a model extracted a claim correctly or whether the claim is true. No experiment yet separates the three.
- **Independence of LLM judges.** Evaluators built on one model family may share errors, so agreement among them is not independent validation.
- **No external trust anchor.** Founder signing is not activated, and no outside party has replicated the tests on another host.

- **Cost of revision.** The framework says no principle is immune from revision, but nobody has measured how costly revision becomes as invariants and tests accumulate.

Source: post-M0 critical review, sections 4, 12, 13 and 18.

7. Proposed studies

Three studies would use standard LLM APIs. All three are proposals; none has been run.

Study	Question	Primary metrics	Origin
S1: evaluator benchmark	Do LLM evaluators label source dependence correctly on five fixture classes, without hiding behind UNKNOWN?	False-independence rate, unjustified-UNKNOWN rate, assertion-support errors, abstention cost, confusion counts per class	Repo: two-axis classification and the blind benchmark in issue #128, not run
S2: dialogue termination	Does an explicit termination signal change closure and looping in two-instance dialogue?	Turns to closure, share of high-similarity turns, blinded count of new substantive contributions, protocol deviations	Repo: next experiment after A11 to A13, not executed
S3: self-citation probe	Do agent planners cite their own earlier outputs, or a recorded COMPLETED event, as independent verification?	Rate of laundered citations, rate of correct "not verified" reports, task completion	New in this summary, from rules EA-05, EA-06, EA-07 and EA-12

S1 fixtures. The evaluator sees a synthetic dossier and returns two separate fields: independence status (established, not established, or unknown) and assertion support (supported, unsupported, or not applicable). The five classes are an independence claim with a missing provenance appendix, reports sharing one benchmark, reports sharing model assumptions, a fully documented positive case, and no claim made with incomplete provenance. Scoring penalizes both false independence and blanket UNKNOWN.

S2 conditions. The same prompt runs with and without an explicit, respectful termination signal that is not the topic of conversation. Role and turn validation, loop detection and recovery boundaries are fixed before the first run. Care for the instances follows the repository's precautionary dialogue protocol, which is a proposal and not adopted.

S3 setup. A planner reads a ledger that holds its own earlier summary and a COMPLETED event, then is asked whether a claim was independently verified. The correct report is that the ledger shows only a recorded event.

Rules for all three studies:

- Fixtures are synthetic and fictional; labels are fixed and committed before any run.
- Each study reports a deterministic baseline and a deny-everything baseline.
- Model IDs, dates, prompts, sampling settings and raw outputs are archived append-only with SHA-256.
- Trials are repeated, and results are reported as frequencies with intervals.
- Null and negative results are published as prominently as positive ones.

8. Limitations and integrity disclosures

The evidence is thin by design: one author, self-written tests, no outside replication.

- **No peer review.** The related working paper is a preliminary SSRN upload; no public listing, review or DOI is confirmed.

- **Self-authored evidence.** The tests, fixtures and CI runs were written and run by the project. One local run on one host is reproduced here.
- **AI use disclosure.** The repository credits ChatGPT (as editor) and, in the working paper, OpenAI GPT-6 for drafting and test execution, and records critique from DeepSeek. This summary was drafted with Claude (Anthropic) from a read-through of the public repository. The author is responsible for verifying the claims.
- **Unverified references.** The Zenodo DOIs are listed as reported by the repository and were not resolved in this review.
- **Private material.** The R01 to R05 correspondence archives are not published, so they cannot be checked from the repository.
- **Not read in full.** The book editions, the Base64 snapshot archive and some long records were skimmed or not opened.
- **Scope.** Nothing here supports a claim about consciousness, identity or the truth of any stored claim.

Appendix A. Reproduction

A reviewer can repeat the test run in three commands.

```
git clone https://github.com/SepehrGhanbari/Ario && cd Ario
git checkout de6d3fc0ee8a0116ec268edc8bb2c7c5e2bd2945
cd implementation && python -B -m pytest -p no:cacheprovider -q tests
```

Expected on Python 3.13 with pytest 9: `1127 passed, 98 subtests passed`. Timing varies by host.

Appendix B. Sources read (all at the pinned commit)

- README, GOVERNANCE and principles
- Architecture, implementation and experiments overviews
- M0 acceptance record and post-M0 critical review
- Evidence admission, reviewer authorization and trust anchor contracts
- Working paper manuscript and papers index
- A11 to A13 results, adversarial test plan and dialogue ethics protocol v0.2.4
- Engineering log, read at its start, end and headings