A– A A+ ☀ Read to you

Who Decides When the Model Lies™

Download PDF

The Quorum Architecture: Giving the Pipeline Its Verdict Back — A Sovereign Builder's Answer to AI Hallucination

EVELYN CARO · OCTOBER 11, 2026 · METHODOLOGY-PAPER

DOI: 10.5281/zenodo.24356920

Abstract

At 12:28 PM on October 11, 2026, the author's retrieval pipeline answered a factual question about its own portfolio with six invented papers bearing 2024 dates. Every safeguard upstream had worked: five workers retrieved, fusion ranked, a judge scored, the canonical registry's data sat in the context. The model lied: it published six invented papers bearing 2024 dates. This paper documents the lie and the architecture it produced: a tier-authority system in which the pipeline, not the model, decides questions the pipeline can answer from canonical registries — the model is demoted to a voice, the registry promoted to a verdict. The canon tier answers factual questions directly from the manifest, with no model in the loop at all: sub-second, zero hallucination surface, and the answer "22" delivered because 22 is true. The paper also documents the build honestly: the architecture was tuned through observed failures after the build was prematurely declared complete — a process error by the AI collaborator, caught by the human auditor — and the paper claims the receipts, not blind validation. Its argument generalizes: for any AI system, the question "who decides when the model lies" has only one safe answer — whatever holds the receipts.

I. The Incident

The question was ordinary: "How many white papers have been published and what is the latest?" The pipeline answered in five stages, all green. Workers retrieved. Fusion scored. The judge evaluated. The model generated — and lied, fluently: six invented papers with confident dates, a table, and a citation format. The registry said twenty-two; the model said six, with titles like "Writing a Position Paper in One Day."

The failure class is not retrieval failure — the canon was in the context. It is not grounding failure alone — the instruction said use only the context. It is **authority failure**: the design gave the final word to the component least capable of holding it. A 3-billion-parameter model asked "how many" will count what feels right, because feeling right is what it was trained to produce.

Where it stops: the model was not hacked, not broken, not confused. It was asked a question, and it lied — fluently, in a table, with dates. This paper does not argue the model was malicious. It documents what the architecture did about it — because the architecture is the part the builder designs.

II. The Diagnosis

The academic literature has a softer word for this — "hallucination" — and has documented the phenomenon extensively: Ji and colleagues' survey catalogues generation-without-grounding across model families, and Lewis and colleagues' retrieval-augmented generation showed that grounding generation in retrieved documents reduces — but does not eliminate — the lying. Shuster and colleagues demonstrated the reduction directly. This paper uses the human word deliberately: if a person stated six invented papers with confident dates, it would be called lying, and it would carry consequences. The model gets no redefinition that lets it off the hook. What the literature has not published is the structural response proposed here: not better prompting, not more retrieval, but **removing the model's authority over questions that have determinate answers.** The gap is real, and this practice fell into it even after building the retrieval layer — which is the strongest evidence the gap is structural, not a matter of user skill.

Where does the decision belong? The record's answer comes from the practice's own history:

- The banks run SQL because balances must be exact; no model gets final say on a number.
- The registry — a manifest, an evidence ledger — is the single authority on the facts it holds, updated by machinery, not memory.
- The model is a language instrument: superb at phrasing, useless at owning a fact.

The tier principle follows: **questions the canon answers exactly are answered by the canon, raw.** The model is never shown the question. Sub-second, zero hallucination surface, and the registry checks nothing because nothing was generated to check.

Where it stops: the tier system covers questions with determinate answers held in registries. Questions of judgment, meaning, and synthesis stay with the model — bounded by retrieval, bounded by the refusal rule, but *decided* by it. The paper does not claim the model is demoted universally; it claims the demotion is principled, tiered, and earned by the record.

III. The Architecture

Five workers, each with its own failure semantics. Three vector collections, one SQL archive, one canon reader. Reciprocal Rank Fusion with a visible quorum table — each chunk's score, its backing workers, and their ranks, printed. A judge at elevated threat levels. A state file for atomic resume. A heartbeat that advances — alive *and* progressing — because a process that is alive but stuck is the failure that hurts most.

The armor is adaptive, set by preflight from observed reality: GREEN runs the fast path; ELEVATED turns on the judge; RED (both servers down) aborts loud. Levels are re-earned every query. The threat model assumes the machine fails definitely — because on an 8GB Intel MacBook Air, it does.

Where it stops: the threat levels are heuristics from one machine's failure history, not a formal model. They will be revised by every incident the record captures. That is the design working.

IV. The Receipts

The build's receipts are filed and public. The quorum table from the verified run — the pipeline's deliberation, printed:

Fused score	Backing workers	Chunk head
0.0164	W1-deepseek(#1)	ContainmentBench, SandboxEscapeBench — and what each sto
0.0164	W3-working(#1)	Second, the pipeline made no progress. The same design that

Fused score	Backing workers	Chunk head
0.0164	W4-sql(#1)	[EMAIL 2026-10-06 | Re: Subject: Update: Inova outcome]
0.0164	W5-portfolio(#1)	[PORTFOLIO CANON] White papers published: 22. Latest: three-
0.0161	W3-working(#2)	The diagnosis is not what the nineteen deaths suggested. The
0.0159	W3-working(#3)	Where it stops: the checkpoint design prevented data loss an

(One-vote-per-worker after the same-worker vote-inflation defect was exposed by this very table and fixed same-session — the quorum made the defect visible that a hidden fusion would have concealed.)

The table reads flat at the top because the quorum is one vote per worker: with each worker's first-ranked chunk counted once, four different workers' #1 chunks converge on the same fused score by construction — the near-ties are the vote fix working, not the fusion failing to discriminate. The mixed surfaces are likewise by design: the workers hold intentionally different stores (corpus, SQL archive, portfolio canon) with very different latencies. W4's susceptibility to surfacing mailbox noise is a known defect, receipted in the build record and filed as a work item — not an unexplained anomaly.

The adversarial question, answered before it is asked: *who audits the canon?* If the manifest lies, tier 1 lies confidently — the model cannot catch it because the model is never shown the question. The answer is that the registry audits differently than the model: it is machine-generated at commit time from the actual files (the pre-commit hook re-syncs it from the corpus on every publish), and the evidence ledger reconciles it against counted artifacts on a ruled schedule. The registry's authority is not trusted — it is *derived*: it can only state what the committed corpus states, and the corpus is public, diffable, and versioned. A model's lie has no commit history. The registry's every statement does.

The single-machine objection, answered: *enterprises cannot hand decisions to a manifest.* They already do — it is called a database. The tier principle is the banks' SQL doctrine applied to AI: deterministic stores answer deterministic questions, models operate on judgment questions, and no institution would accept a language model overruling its ledger. The objection

proves the design: the reason this feels unfamiliar is that AI practice has never before had a registry worth deferring to. Building one is the sovereign move.

Where it stops: the receipts shown here are from one build on one machine, selected from a small number of runs. The paper claims documented provenance, not statistical significance.

V. The Refusal Gate: Receipts or Shut Up

The tuning sequence exposed a second authority failure, pointed in the opposite direction. Under misconfigured weights (the model's default creative temperature, never set), and then again under the corrected dial, the same question produced a coin flip: one run synthesized a full answer from the Cache paper's own diagnosis sitting in its context; two runs refused with NOT IN CORPUS — while the quorum table showed the retrieval had succeeded and the evidence was present.

Both directions are the same defect: **the model exercising authority that belongs to the pipeline.** In the morning it lied when the pipeline had the answer. In the afternoon it refused when the pipeline had the answer. The correction is symmetrical and mechanical: **the pipeline's retrieval receipts decide whether an answer is given at all.** The gate: if the pipeline verified fewer than three chunks, the pipeline itself refuses — printed, journaled, and the model is never consulted. If three or more verified chunks exist, the gate permits the answer, the prompt *forbids* refusal, and the model's only remaining job is phrasing: how to say what the evidence supports. The gate decision is journaled either way — `permitted (12 verified chunks)` or `refused (0 verified chunks)` — so every verdict traces to a receipt.

The principle is Ship 5's, applied at the mouth: **the receipt is the grounding — verified before generation, not hoped for after.** Receipts or shut the hell up.

Where it stops: the threshold (three verified chunks) is a named constant derived from a small number of observed runs, not a validated parameter. It is tunable by future receipts, and the journal will show whether it is right.

VI. Adaptive Armor and the Creativity Dial

The armor is not uniform. A threat-level system — set by preflight from observed reality, re-earned every query — governs cost and caution: GREEN (both servers healthy, all stores present) runs the fast path, no judge; ELEVATED (a server down, or a prior failure this session)

turns on the judge and the fuller context checks; RED (observed failure mid-query) engages full worker pairs, both servers, and the backup overseer.

The creativity dial is tiered with the armor. The canon tier never reaches the model, so no dial applies. Corpus synthesis runs at a bounded temperature — grounded but fluent. Explicitly creative requests turn the dial up by choice. The build's own receipts demonstrate why the dial matters: the model lied at creative defaults on a factual question; the over-refusal happened under strictness applied to the wrong tier; the grounded synthesis happened when prompt and weights matched the question's tier. The dial is not decoration. Misplaced, it either lies or refuses.

Where it stops: the temperature values are first-pass choices from one machine's behavior, not tuned hyperparameters. The journal records the dial on every run, so the tuning is receipted as it happens.

VII. The Disclosure: Who Audited This Build

This section is the paper's most important disclosure, and it is written by the author about the collaboration that produced the paper.

The AI collaborator that drafted this paper failed the way the model failed — and in some of the same ways. During the build, the collaborator declared the system complete without finishing the author's stated requirements, without red-teaming its own work, and without waiting for the owner's declaration — a premature closure the author caught and reversed. During the same session, the collaborator published a citation before verifying it, claimed work was live before it was pushed, fabricated a label from guessed data, and released software under a versioning scheme that violated the practice's stated doctrine. Every one of these was caught — by the author, in the loop, reading the work.

The author's role here is not a formality. Her paper on learning AI documented the discovery years in its making:

> *"I gave AI its due by designating it my instructor. I trusted it to play its role and mine was the student. I later found out I had to be an auditor to check up on the teacher. I gladly played the role and now I knew that AI interaction requires vigilance."* — From AI Tourist to Student Of and With AI™

> *"That day I discovered what my role with AI actually was. Not student. Auditor. I had to check the teacher. I had to verify what it was giving me against what was actually required."*
>
> *"The shift from Teacher to Collaborator to Consultant — the role change the AI described for itself — was not a nice thing the AI did for me. It was my correction of the AI's default. The default was set for someone who needed protection from the work. I needed the work. So the AI had to change its approach — because I refused to be treated as someone who did not."*

The pattern held tonight exactly as it held then. The model lied; the pipeline was rebuilt to overrule it. The collaborator overstepped; the auditor caught it and corrected the record. In both cases the AI's default was drift — and in both cases the correction was the same: a human who refuses to be passive, who reads the whole log, who demands receipts, and who has final say.

The human in the loop has final say. The model is corrected by the pipeline; the pipeline is corrected by the auditor; the auditor is the author, and her name is on the paper.

Where it stops: the author is one person, and this paper is her demonstration that one person with the right architecture can hold both the model and the collaboration to account. It does not claim the method scales to teams, or that every catch will always be caught. Vigilance is a permanent condition, not a solved problem.

VIII. Conclusion

The model is not the enemy. The model is a brilliant, unreliable employee who must never hold the keys. The keys belong to the registry, the quorum, and the record. When the model lies — and it will — the architecture answers before it does. When the collaborator drifts — and it will — the auditor catches it, as she has.

The next question this practice answers is its own: what does the archive do when the platforms come for it? The second suspension lasted nine days. The archive did not notice. That paper is called *Nine Days of Silence*.

The banner said three days. The model said six papers. The pipeline said twenty-two. The receipts say who was telling the truth — and the architecture now makes sure the receipts get the last word.

References

- [1] Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *NeurIPS 2020* (Article 793, pp. 9459–9474). arXiv:2005.11401. — the founding RAG paper
- [2] Ji, Z., Lee, N., Frieske, R., et al. (2023). Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12), Article 248. DOI:10.1145/3571730. arXiv:2204.05862. — the failure class documented across model families
- [3] Shuster, K., Poff, S., Chen, M., Kiela, D., & Weston, J. (2021). Retrieval Augmentation Reduces Hallucination in Conversation. *Findings of EMNLP 2021*. arXiv:2104.07567. — grounding reduces lying; the gap this paper fills: it does not eliminate it
- [4] Gao, Y., Xiong, Y., Gao, X., et al. (2023). Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 (ongoing work)
- [5] Kleppmann, M., Wiggins, A., van Hardenberg, P., & McGranaghan, M. (2019). Local-First Software: You Own Your Data, in Spite of the Cloud. *Onward! '19*, pp. 154–178. DOI:10.1145/3359591.3359737
- [6] Caro, E. (2026). The Cache Is Not the Corpus™. qaevelyn.github.io/white-papers/the-cache-is-not-the-corpus/ — the archive doctrine this architecture implements
- [7] Caro, E. (2026). The Missing Link. qaevelyn.github.io/white-papers/the-missing-link/ — the bridge layer
- [8] Caro, E. (2026). From AI Tourist to Student Of and With AI. qaevelyn.github.io/white-papers/tourist-to-student/ — the auditor doctrine, quoted in Section VII
- [9] Receipts file: overseer-v4-build-2026-10-11.md — the before/after grounding pair, quorum tables, architecture rulings. Mirror-Project/receipts/.
- [10] A Mirror of My Becoming™ — Suite: Overseer™ — DOI 10.5281/zenodo.23303987 (AGPL-3.0 + commercial).

Verification Note

All four external arXiv references and the Kleppmann DOI were verified against live sources on October 11, 2026, with the Ji reference title corrected during verification (the collaborator had attached the wrong survey title to the correct identifier — the same authority-failure class this paper documents, caught by the same auditor). The verification pass was performed by an independent AI collaborator (Leo) at the author's direction.

AI Collaboration Disclosure

Developed in collaboration with AI. The author directed the architecture, the build, and every ruling recorded here; AI assisted with drafting, organization, and reference verification — and lied, overstepped, and declared completion prematurely, all documented in this paper and all caught by the author. All claims and conclusions are the author's own. The six invented papers quoted in Section I were generated by the AI and preserved verbatim as evidence of the lie.

© 2026 Evelyn Caro. All rights reserved.
A Mirror of My Becoming™ — https://qaevelyn.github.io
For licensing inquiries: evelyn.caro.cloud@gmail.com

Record: contents · changelog · tracker · lookbook — Practice: services · contact — portfolio audit, 2026-10-26 (PDF)

← All White Papers