Hallucinations Shouldn’t Be Silent:

The Orchestration Framework (TOF) for Transparent AI

Lauren Morrell
Independent Researcher & AI Systems Architect

Founder & CEO, Pugilist Orb • CSO, SocialEyes
ORCID: https://orcid.org/0009-0005-9009-2424

Abstract
Large language models (LLMs) are powerful but prone to hallucinations, confidently generating false or unverifiable information. The Orchestration Framework (TOF) introduces three mechanisms: Bridge Summaries, Admin Notes, and Sentinel Flags that help preserve context, reinject validated knowledge, and flag uncertainty in AI outputs. TOF provides a structured transparency layer for multi-turn reasoning, improving reliability while preserving creativity and human agency.

Keywords: AI Transparency · Hallucination Detection · Interpretability · Reliability · Large Language Models

Large language models are remarkable at generating information and making connections, but that strength comes with a hidden risk. They don't just forget facts; they sometimes invent them. In AI, this is called a hallucination, and the problem is that most users don't know when it's happening. There's no clear signal to show when a system has shifted from truth to invention, and that uncertainty makes hallucinations particularly dangerous.

The risks grow when these models are combined with tools, autonomous agents, or long-term memory that can act on their output. A single mistake in casual conversation might be harmless, but in fields like healthcare, finance, or education, one confident error can move quickly, influence decisions, and cause real harm before anyone realizes it.

This is the challenge TOF was designed to address: giving users a clear signal when AI shifts from grounded information into improvisation.

Why Hallucinations Happen

To understand why these fixes matter, it helps to know how large language models (LLMs) work. On their own, LLMs aren't search engines or databases. Most of the time, they're improvisers, predicting the next word based on patterns in their training data. When paired with retrieval systems, they can surface external facts. TOF focuses on clarifying when the model is grounded in evidence and when it's improvising.

A few of the key challenges are:

- **Limited memory:** LLMs can only "see" a certain window of conversation at once. As the dialogue gets longer, early details fall away, leading to forgotten context. When that happens, the model rarely signals that it has lost the thread. It often invents details to keep the conversation flowing.

- **Fragile recall**: Facts (like names, numbers, or definitions) can get dropped or subtly changed as the model generates text.

- **Uncertainty hidden under fluency:** Even when a model is unsure, it still produces confident-sounding language, leaving the user with no signal that the answer may be shaky.

These gaps are exactly what TOF is designed to patch: ***Bridge Summaries*** for memory, ***Admin Notes*** for facts, and ***Sentinel Flags*** for uncertainty.

Why Can't We Just Train AI Not to Hallucinate?

It is tempting to think the fix is simple: just train AI not to make things up. But hallucinations aren't a glitch; they are baked into how these systems generate language. LLMs don't know facts in the way people do. They predict what words are likely to come next based on patterns in their training data. When information is missing, outdated, or outside their training, they will still generate fluent text, even if it is invented.

Retraining can reduce hallucinations, but it cannot eliminate them. That's why visibly showing when a system is uncertain, forgetting, or inventing, becomes just as important as accuracy.

What's Been Tried So Far

Researchers are already experimenting with ways to reduce hallucinations: reinforcement learning with human feedback, retrieval-augmented generation, and post hoc fact-checking. These approaches have value; they make models less likely to go off track. They do not solve the core problem, though: users still have no way of knowing when the AI has slipped from truth into invention. Errors remain silent. TOF is designed to **complement** these approaches by adding a transparency layer that makes uncertainty and context limits visible to users.

The Orchestration Framework (TOF)

I am an independent researcher and systems designer who began exploring these issues out of necessity. The more I learned, the clearer it became: if we want to trust AI, we need to make uncertainty visible. I sketched out a system I call ***The Orchestration Framework (TOF)***, three mechanisms that work together like a support crew for AI.

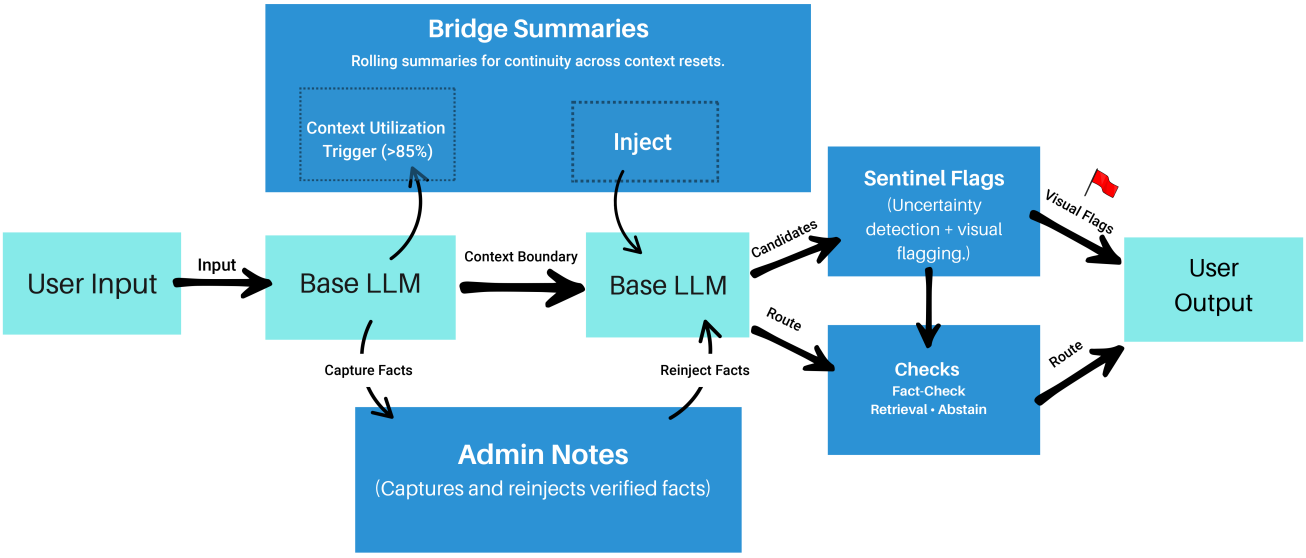
Bridge Summaries step in when memory is full. They create rolling summaries so the AI does not lose track of earlier context (think of it like passing a relay baton of information).

Admin Notes quietly capture key facts, names, numbers, definitions, commitments, in a sidecar log so details do not get lost.

Sentinel Flags fire when the AI is uncertain or drifting into risky territory, surfacing a flag, routing to fact-checking, or abstaining from an answer. (This is the "Major Tom" component; ground control to keep the AI tethered.)

Together, these three mechanisms shift hallucinations from hidden and silent to visible and actionable.

Figure 1: TOF Architecture - Transparency Layer around a Base LLM



Cite as: Morrell, L. (2025). The Orchestration Framework (TOF): Bridge, Admin, and Sentinel Tokens for Reliable LLMs. v1.0.

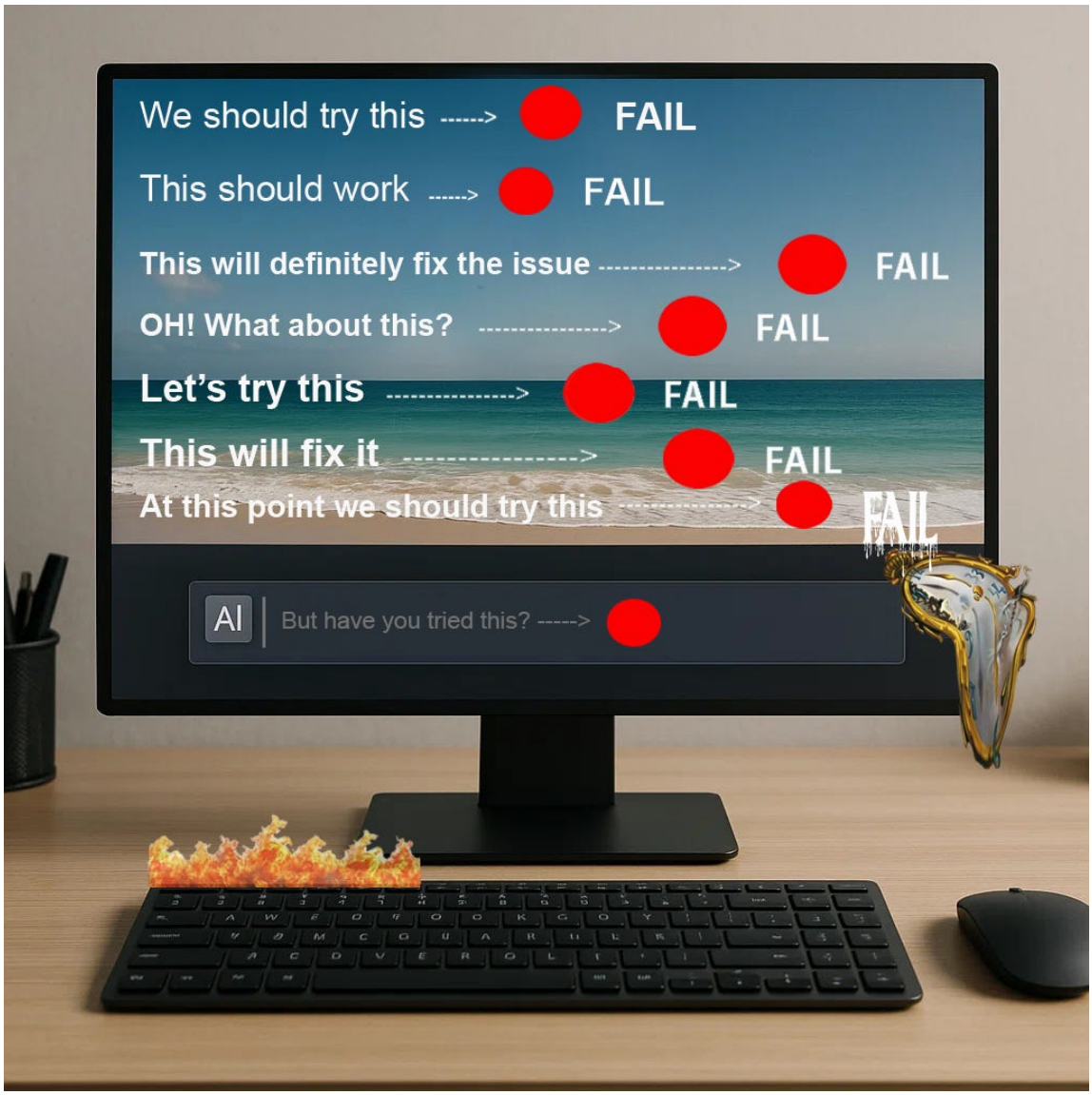
Why it matters

If an AI makes a claim, the user should know whether that claim is solid, uncertain, or needs checking. Imagine responses where questionable statements are flagged, underlined and linked to a "View Evidence" panel. Imagine long conversations where continuity doesn't unravel. Imagine audit trails that show not just the answer, but the uncertainty behind it. That's what TOF **is designed to make possible**: models that don't just generate, they also mark their own blind spots.

Future Applications: Beyond Hallucination Detection

While TOF was designed to make **hallucinations visible**, its architecture also addresses a broader challenge: **continuity loss in multi-step tasks**.

Large language models like GPT-5 now engage in **deep nested planning**, consuming tokens faster and silently discarding earlier context when the window fills. This creates a hidden risk: the model may **repeat failed approaches** or **revisit discarded assumptions** without realizing it, wasting both time and resources.



Case Example: Debugging a Mobile Game

During development of *Gem Bloom*, a simple mobile game, we encountered recurring interface bugs. After multiple debugging cycles, the model began proposing the same fixes we had already tested, blind to past failures. In larger, more complex projects, this kind of "rabbit-hole" repetition can waste significant time, introduce new bugs, and erode code stability.

With TOF in place, these loops are avoided:

- **Bridge Summaries** would trigger when context usage nears **85%**, producing compact summaries that preserve prior debugging steps before details vanish.
- **Admin Notes** would record attempted fixes and their outcomes, reinjecting them only when relevant.
- **Sentinel Flags** would highlight uncertainty or contradictions in the model's suggestions, signaling when extra scrutiny is needed.

This layered approach keeps multi-turn problem solving **aware, efficient, and transparent** whether debugging code, synthesizing research, or coordinating multi-agent workflows.

How It Works in Practice

Bridge Summaries: Trigger automatically when the context window approaches **85%**, producing rolling summaries that compress older steps without losing continuity.

Admin Notes: Store validated details externally (like tested fixes, parameters, or errors) and reinject them only when needed.

Sentinel Flags: Watch for warning signs such as high entropy, contradictions, or unsupported claims; when flagged, route responses through extra checks before adoption.

TOF's initial experimental goals are:

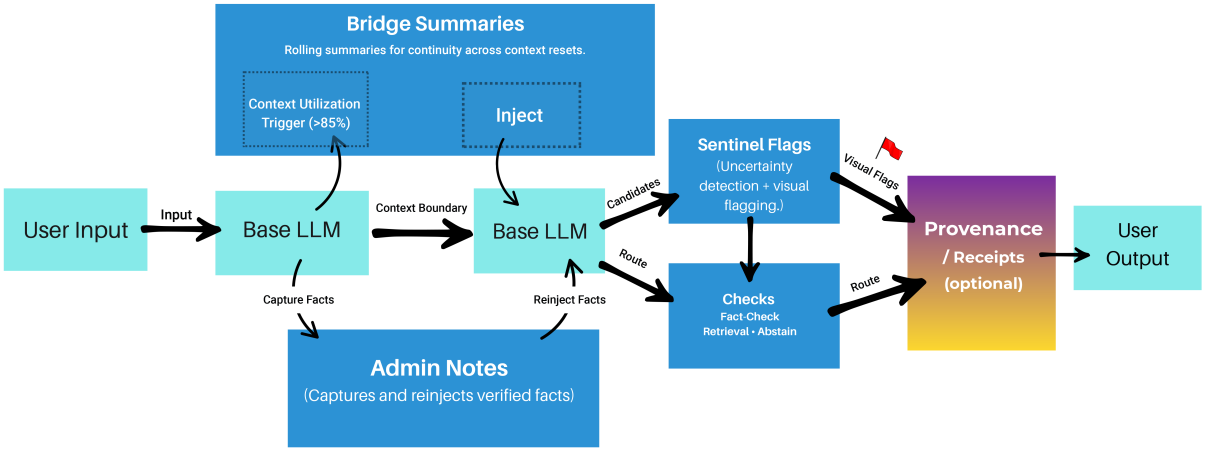
- reducing unsupported claims by up to 40%
- improving factual recall by around 25%
- keeping latency overhead near +1 second
- achieving calibration AUC around 0.6

These figures are design goals, not measured results, and will require validation through future experiments.

Optional Extension: Provenance and Blockchain

Hallucinations will not vanish overnight, but with TOF they can be surfaced, reduced, and studied. One way to study them is through provenance receipts: lightweight logs that record which claims were checked, what evidence was used, and when. Anchoring these receipts in an append-only log (and optionally a blockchain) creates a dual benefit: accountability for users, anyone can trace where a claim came from, and data for researchers, so hallucinations become measurable phenomena rather than invisible failures. This also enables systematic studies of hallucination patterns and model drift over time.

Figure 2. Extended TOF Workflow



Cite as: Morrell, L. (2025). The Orchestration Framework (TOF): Bridge, Admin, and Sentinel Tokens for Reliable LLMs. v1.0.

Why I'm Publishing This Openly

I don't think reliability should be gatekept. That's why I'm releasing TOF as a public framework. The package includes a technical paper, an implementation blueprint, a research plan, and diagrams.

GitHub Repo

My hope is that researchers, developers, and builders will pressure-test TOF, improve it, and adapt it into real systems.

Closing

AI does not need to be flawless to be trustworthy. It does need to be transparent about its uncertainties, and it needs a structure to keep it grounded. That's what The Orchestration Framework (TOF) is meant to provide: *Bridge Summaries, Admin Notes, Sentinel Flags*, a small support crew to keep the improvisation honest.

Disclaimer

This article presents *The Orchestration Framework (TOF)* as a conceptual and architectural approach to improving transparency and reliability in large language models. The framework is described at the level of design principles, workflow diagrams, and illustrative case examples, rather than as production-ready code. Figures, diagrams, and examples are intended to clarify the approach, not to demonstrate experimental benchmarks.

The purpose of this preprint is to share the framework with the research community, encourage discussion, and invite collaboration. Implementation details and empirical evaluation may be developed in future work or by others who wish to build upon these ideas.

Upcoming: "Hidden Drift: How GPT-5's Deep Planning Amplifies Context Loss and How TOF Can Help"

This follow-up research note will explore how TOF extended beyond hallucination detection to handle **continuity transparency,** context resets, and silent token loss.

Lauren Morrell is an Independent Researcher and AI Systems Architect. She is the creator of The Orchestration Framework (TOF), the Founder & CEO of Pugilist Orb and the CSO of SocialEyes.

References

*Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., ... & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys (CSUR)*, 55(12), 1–38.*

Bang, Y., Cahyawijaya, S., Lee, N., Do, Q. V., Zhang, Y., & Fung, P. (2023). A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. arXiv:2302.04023.

Lin, S., Hilton, J., & Evans, O. (2022). TruthfulQA: Measuring how models mimic human falsehoods. arXiv:2109.07958.

Kadavath, S., Conerly, T., McKinnon, C., Snell, C., Hatfield-Dodds, Z., ... & Irving, G. (2022). Language models (mostly) know what they know. arXiv:2207.05221.