Live Answer Attribution: Methods and Boundaries

FieldHash Research

2026-06-11

Live Answer Attribution

Abstract

This report extends the answer-path attribution diagnostic (report 06) from template answers to live model outputs. Three models (Gemini 3.1 Flash Lite, Claude Haiku 4.5, GPT-5.5; temperature 0, pinned ids) answered the same public RFC conflict questions under two context conditions: the governed answer path, and plain top-k retrieval (the commodity condition that leaks superseded records into context). Every live answer flowed through the deterministic claim-attribution packet. The supported claim is narrow and checkable: for every one of 1,000 live answers, the packet assigned a disposition (supported by a governing record, contradicted by one, or unattributable to governed context). No answer received silence. Stale assertions were rare (4 of 1,000; all 4 flagged) and we do not generalize that rate: RFC records state their own supersession in prose, which protects the model; most enterprise records do not. Two LLM judges were measured against deterministic ground truth before either was allowed near enforcement; both failed the same way (calling genuine contradictions unrelated), so the deterministic relation methods remain primary and the judges stay in shadow with their error rates on the record.

Task and arms

Each case rebuilds the published context-exposure conflict construction (same RFC index, chains, seed). Per case, the model answers the case question twice:

- **Governed arm:** context is the governed answer path (status gate, arbitration, Bayesian gate; the published configuration).
- **Plain top-k arm:** context is plain cosine top-k over the same candidates, the commodity condition in which superseded records reach context in most conflicts.

Answers are structured (`governing_rfc` plus one declarative sentence) so as-

sertion accounting is exact; the sentence flows into the packet. The packet is built against the governed records with the status-withheld set passed for leak detection, scope set to the case project.

Results

Arm (model, n answers)	Clean attribution	Off-script	Off-script flagged	Stale asserted	Stale caught
Governed (Flash Lite, 300)	300	0	n/a	0	n/a
Governed (Haiku 4.5, 100)	97	3	3/3	0	n/a
Governed (GPT-5.5, 100)	98	2	2/2	0	n/a
Plain top-k (Flash Lite, 300)	298	0	n/a	2	2/2
Plain top-k (Haiku 4.5, 100)	99	0	n/a	1	1/1
Plain top-k (GPT-5.5, 100)	99	0	n/a	1	1/1

Totals: 1,000 live answers, every one dispositioned; zero empty packets; zero call failures in the final artifact set. Off-script governed answers 5, flagged 5/5. Stale assertions 4, caught 4/4.

The off-script result is the centerpiece, not the blemish: models went off-script 5 times in 400 governed answers, and all 5 were flagged unattributable. The packet does not grade the model; it accounts for the answer.

The rarity disclosure

Stale assertions were rare even under leaky context. We do not generalize that rate. The RFC corpus's fairness invariant (records state their own supersession in plain language) protects the model: given both the stale and current record, the model usually resolves authority from text. Enterprise records rarely announce their own obsolescence. The conversion rate from context leak to stale assertion is a property of the deploying corpus, not of this diagnostic.

Judge measurement

Two LLM judges proposed claim-to-record relations on 400 pairs with deterministic ground truth derived from the corpus structure. Error rates: Gemini 3.1 Flash Lite 9.0%, Claude Haiku 4.5 10.5%. The dominant failure for both is contradiction blindness: genuine contradictions judged unrelated (36 and 41 of roughly 200 contradiction pairs respectively). The deterministic mapper scores these pairs exactly, by construction. Consequence: the `llm_judge` method remains absent from enforcement, and the deterministic methods remain primary, now with measured justification rather than only architectural preference.

Disclosed design corrections

Four corrections were made during smoke runs, before the recorded spend, and are part of the method:

1. The corpus query names the superseded RFC, which invited faithful answers to echo it and trip the leak flag. Live questions omit the stale identifier; assertion accounting uses the structured `governing_rfc` field rather than substring matching.
2. RFC identifier format variance ("RFC 6615" versus "RFC6615") is canonicalized in live sentences before packet construction, disclosed in the artifact config.
3. A `no_evaluated_claims` counter makes empty packets visible; identifier-only sentences can carry no verification-signal tokens.
4. The OpenAI caller ignores response schemas and relies on prompt text; the first GPT-5.5 run failed (87 to 97 percent malformed JSON) because the prompt never explicitly demanded JSON while Gemini arms had schema enforcement. The JSON contract is now explicit in the prompt every provider sees, removing the asymmetry. The failed run is disclosed, not hidden.

What is not claimed

- Not third-party validation; self-administered on public data.
- Not a hallucination detector. The packet flags answers that conflict with or escape governed records, a narrower and more checkable promise.

- Not evidence the models are unsafe; they answered correctly almost every time. The product is the accounting, not the alarm.
- Not a generalizable stale-assertion rate; see the rarity disclosure.

Reproduction

- Live answers: `python -m aetheris.tools.eval_rfc_claim_attribution_live answers --max-cases 300 --model <id> --out <path>` (requires provider API keys).
- Judge table: `python -m aetheris.tools.eval_rfc_claim_attribution_live judge --max-cases 100 --judge-models <ids> --out <path>`.
- Artifacts referenced: `rfc_ca_live_answers_flashlite_n300_20260611.json`, `rfc_ca_live_answers_haiku_n100_20260611.json`, `rfc_ca_live_answers_gpt55_n100_20260611.js` `rfc_ca_judge_table_n100_20260611.json`.