

The Assistant That Refuses to Guess

A deterministic answer layer in front of a local LLM: it serves what it can prove from a knowledge graph and refuses the rest — 376 answers across two exam domains, none wrong.

Vincent Grandjean — AME Labs — vincent.grandjean@ame-labs.eu

Ask a language model about the world and it is remarkable. Ask it about *your* world — “Where is the metoprolol kept?”, “Is the thermostat battery-powered?” — and something has to give. A bare model doesn’t hold your facts, so it either invents an answer, fluently, or — if it is well calibrated — declines almost everything. Retrieval is the standard cure: embed your facts, fetch the closest few into the prompt, let the model assemble the answer. It works, until it doesn’t — and when it doesn’t, *it still answers*. On our exams, retrieval asserted between 39 and 44 wrong answers about a household and between 55 and 63 about a medication record — each a range spanning a strict and a lenient scoring rule — including one case with every correct fact already sitting in its prompt.

Retrieval is not bad at this, and the paper says so plainly: in the household domain it matched the layer on raw correctness — a statistical tie. The difference is what happens at the edge. When retrieval fails, it fails in the same confident voice it uses when it succeeds.

Both options answer. Neither has a principled way to refuse. Closing that gap is what this piece is about. **Across two exam domains, the layer described below served 376 answers, in about a millisecond each, for about a thousandth of the energy of the model call each one displaced. None was wrong.** And the refusals turned out to be the interesting part.

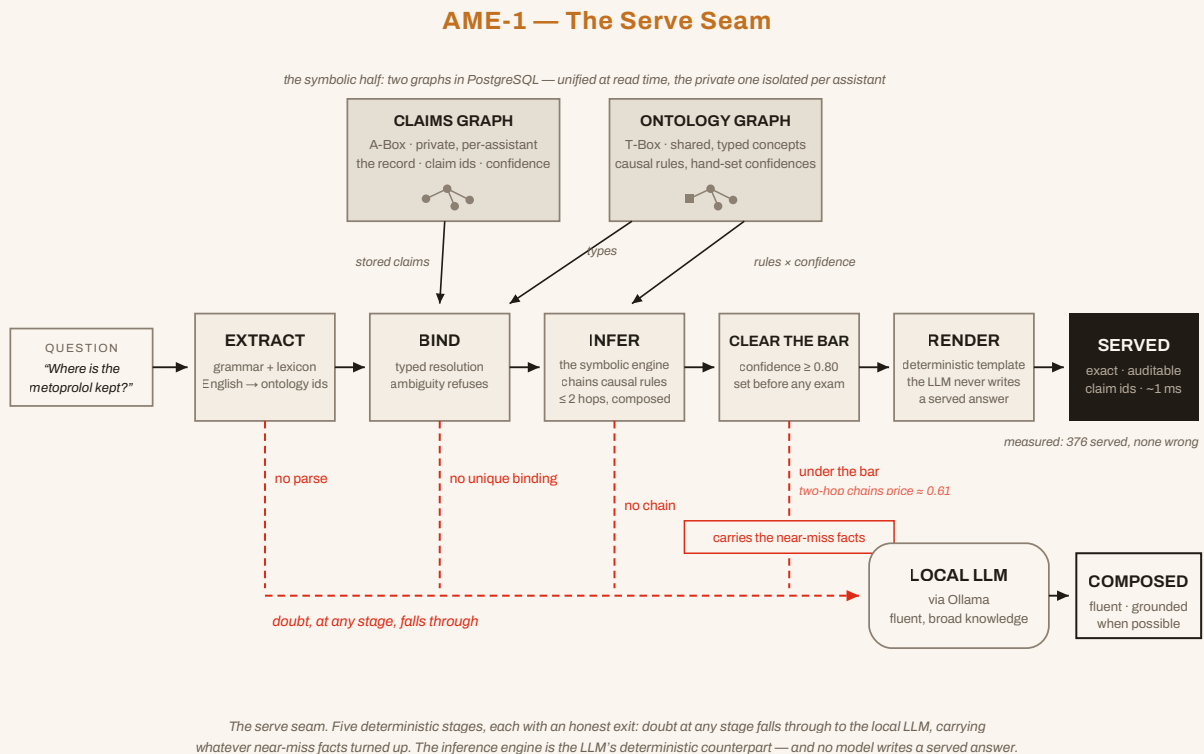
AME-1, and the loop this is step one of

AME-1 — *Artificial Minds Experiment 1* — is a personal-assistant framework I am building specification-first at AME Labs. The architecture is defined in full: **twenty-one modules across seven functional layers** — interfaces, cognition, an action pipeline running from intent to execution, world and user models, layered memory and knowledge integration, and a governance tier where ethics, security, and audit are modules with authority rather than adjectives — wired together by an event-driven backbone, with the assistant’s **self-knowledge** (what it can do, what it does not yet know) held as first-class data it can consult and act on. And it is one engine configured, never forked, into many shapes: a single character in a home, or federated characters collaborating across organizations without merging their data — local-first and private by construction throughout. The full picture lives on the [framework page](#).

The experiment in this article exercises one deliberately narrow slice of that: the serve path crosses **four of the twenty-one modules** — semantic extraction and routing, the

symbolic memory stores, the per-claim confidence rail, and the reply composer. In the slice under test everything runs locally — the models through Ollama, the embedder in ONNX — and the knowledge lives in two graphs in PostgreSQL: a shared **ontology graph** (typed concepts and causal rules, each rule carrying a hand-set, documented confidence) and a private, per-assistant **claims graph** — the record itself, isolated by construction rather than by promise.

In front of the language model sits a deterministic **serve seam**:



A question is parsed by an extraction grammar and bound against the typed graph (ambiguity is a failure, not a choice). A causal question then runs through the **symbolic inference engine** — the LLM's deterministic counterpart, chaining curated rules up to two hops and composing their confidences as it goes. The result is priced against a confidence bar fixed before any exam existed, and — if everything holds — rendered by a template filled from the stored facts: **the LLM never writes a served answer**; it composes only the fall-throughs. And the posture everything rests on: doubt, at any stage, falls through to the LLM, together with whatever near-miss facts the attempt turned up, as grounding in the model's prompt. The router never argues with the model. It either knows, or it gets out of the way.

And the seam is deliberately step one of a loop, most of which already exists in-house:

- **Serve** what the graph can prove — the subject of this article.
- **Ground** what it can't: a refusal hands the model the near-miss facts it turned up on the way.

- **Record** every miss as an explicit knowledge gap — and the gap drives the assistant to *ask*.
- **Gate** every write, so nothing the user, the model, or a future source asserts enters the record unchecked.
- **Distill**, eventually: the LLM itself becoming one more registered, gated source of structured claims.

Step one had to be the seam you can trust, because every later stage stands on it.

What we did, in one paragraph

Two exams of 250 questions each, both over entirely synthetic data: an invented domotic household AME-1 was developed against, and an invented medication record chosen to be grammatically far from it — no real person and no real household anywhere in the evidence, and questions about a record and published mechanisms, never advice. Exam criteria and pass/fail bands were frozen before the questions were written; the knowledge graphs and workloads are hash-pinned in git; per-cell outcomes were predicted in writing, from the shipped code, before every run; and the baselines — retrieval over the *same facts in the same words*, and a stronger bare model — ran under criteria frozen before those baselines were built. **This article states results and deliberately demonstrates nothing**; the demonstration, with every number in context, is the scientific paper that follows it.

376 answers, none wrong — and the refusals are the story

The headline is a count, not a rate: across the two domains the router served **376 answers, and none of them was wrong**. It answered roughly two-thirds of each exam and refused the rest — and the refusals are the interesting part.

The boundary of what it can answer sits exactly where its vocabulary ends. Pointed cold at the medication domain — no lexicon for it at all — the router asserted nothing wrong and overreached zero times; it simply declined nearly everything. After a human spent a session writing **159 rows of vocabulary** — entries in constant tables — it scored in the new domain exactly what it scores at home, cell for cell. We had pre-committed to the claim that adapting to a new domain is a data edit rather than a code change — and we designed a falsifier into it, which fired: two small code changes were needed as well, each deleting a hardcoded English word sitting beside a table. We report that rather than soften it. Honesty requires saying two more things: those rows are written by a human today — making the system derive them is precisely the loop's later stages — and the vocabulary was written with the frozen exam questions in view, which is why the paper pre-commits a held-out rewrite by someone who did not author the grammar.

The refusals have a personality worth describing. On every two-hop causal question — *can a missed dose end up bringing my symptoms back?* — the engine is not stumped: it **derives the full conclusion, prices it, and refuses to assert it**, because two chained rules compose to a confidence around 0.61 and the bar is 0.80. In a shadow measurement, all 38 of 38 two-hop conclusions it declined to state turned out to be the correct ones. (Read that honestly: the exams were authored from the same graph the router walks, so this confirms the machinery, not the wisdom of a lower bar.) It withholds right answers, deliberately — which raises a question we have deliberately not yet answered: whether a *derived* conclusion deserves its own, separately calibrated bar.

Refusal is not an apology, either. A refused turn hands the model the near-miss facts, and that grounding measurably flips answers from wrong to right in both domains. It cuts both ways, and we measured that too: falsify a single fact inside the block and the model follows it off a cliff — in the medication domain, all forty answers went wrong. The model follows its context, which is exactly why choosing that context deterministically is the design. And when the record holds nothing at all, the gap machinery shows its manners: asked about a nonsense entity in the live harness, the gated assistant said “*I don’t seem to have information about the Zibbelwump yet. Could you please tell me more...?*” — while the ungated reply beside it hallucinated Zibbelwump lore.

An answer you can audit, a call you never make

Auditability. A served answer cites the exact claim ids it stands on — you can trace it to the stored records and the rules that produced it. An answer you can audit is different in kind from an answer you can only rate, and for anything record-shaped — medication, contracts, compliance — that difference is the product.

Speed. A served answer returns in about a millisecond; the model call it displaces takes the better part of a second, locally. Instant answers to record questions stop being a caching trick and become a property of the architecture.

Energy and inference economics. Measured on the full serve path, a served answer costs about three orders of magnitude less energy than the model call it replaces. **Every served turn is a GPU call that never happens** — so if a meaningful slice of assistant traffic is record-shaped (and in a household or a care context, it is), the inference bill stops scaling with chattiness. That last sentence is extrapolation, and I label it as such — but it is the kind that follows directly from the measured per-turn numbers.

Near-misses. The two exams carry 84 deliberately confusable pairs — metoprolol tartrate against metoprolol succinate, one strength of levothyroxine against another, a brand inhaler against its generic. The layer separated all 84. Retrieval conflated 14, answering about the wrong member of the pair in that same confident voice. On a record, that distinction is the whole game.

Privacy, structurally. Everything runs local, and the record lives in a per-assistant graph that is isolated by construction. Nothing about the serve path requires a cloud.

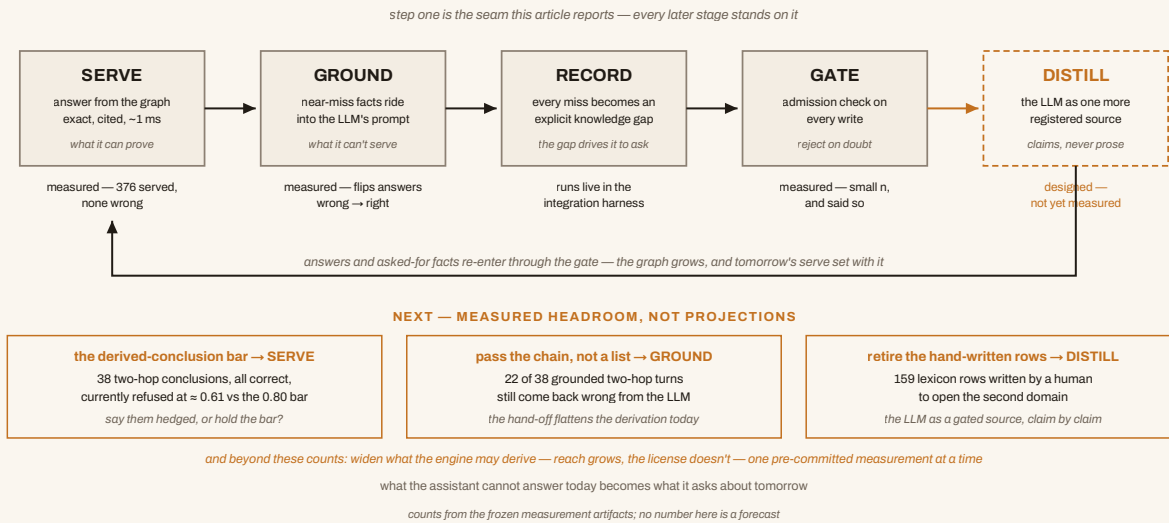
“Isn’t this just a look-up table?”

Mostly — and deliberately. Cognitive scientists asking what it would even mean for a machine to *understand* use exactly this contrast as their boundary case: a look-up table versus a system that derives answers it never stored (Chen, Grimm, Russakovsky & Lombrozo, *Trends in Cognitive Sciences*, 2026). This piece makes no claim on that spectrum. For the facts of *your* record, a look-up that cites the claims it stands on, refuses on doubt, and costs a millisecond is the design goal, not the embarrassment — a wrong answer about a medication record is worse than a refusal. The one step the layer takes beyond look-up — chaining two curated causal rules and pricing the result’s confidence — is the smallest such step, and it runs under the same zero-wrong discipline rather than on trust.

The loop closes from here

Four things, in order of how much they follow from what was measured. First, the **derived-conclusion bar**: the router currently sits on 38 correct two-hop answers it refuses to say; **whether it should say them hedged, or hold its bar, is a calibration question we have scheduled rather than guessed at**. Second, a better hand-off: today the refusal passes the model a flattened list of near-miss facts, and the measurements suggest passing the *chain* — with its steps and its confidence — would ground two-hop questions better. Third, **widening what the engine may derive**: everything measured here exercises the narrowest inference the architecture offers — lookup plus one derivation family, curated causal rules chained at most two hops. The engine underneath registers **49 reasoning primitives across 11 categories** — relational, temporal, spatial, abductive among them — and this article claims nothing about that breadth. The plan is to widen the exercised set one pre-committed measurement at a time, and each widening changes the engine’s *reach*, never its *license*: the confidence bar, the refusal discipline, and the audit trail hold as they are. Fourth, the **distillation leg** that begins to retire the human-written-lexicon sentence above: the LLM as a registered source, its claims entering through the same admission gate as everything else.

AME-1 — The Loop



The loop, with what is on the table next. Each stage carries its measurement status — distillation is designed and labeled so. The three amber boxes are the populations the next steps aim at, counted in the frozen artifacts. No number in this figure is a forecast.

The limits are the same ones the measurements have: two domains, not N; curated graphs at hundreds-of-facts scale, not your data lake; local models up to 9.7B; exam traffic, not production. That is what step one of a loop looks like — not an assistant that answers more, but an assistant that knows *which* questions it can answer exactly, answers those from your record in your words, and turns everything else into a grounded fall-through today and a recorded gap for tomorrow.

Everything above was measured under criteria frozen before the exam questions existed, on hash-pinned workloads, with per-cell outcomes predicted in writing before each run. The measurement artifacts live in the project's research repository; the scientific paper that carries the full demonstration — every number above, with its intervals and its caveats — follows this article.