

Helm — an experiment in AI continuity

Maxwell Connolly · August 2026

SUMMARY

Paper 01 described three products built by one operator through a single pipeline, and ended on a question: what does the production method have to look like, and how is machine-generated output verified. Helm is the answer to the first half. Persistent memory is no longer what distinguishes an assistant — every major vendor shipped it this year. What is described here is different in what persists: a single continuing identity that holds its own positions, is not bound to any one model vendor, develops those positions over time, and raises matters unprompted. This paper sets out the goal state, the theory behind how the system forms beliefs, its architecture, how work is dispatched through it, and the instrument built to certify that its memory is reliable rather than merely convincing. It is an experiment in progress. Its certification threshold is currently met by no candidate engine, including the one in use.

1 What this is

1.1 What is no longer novel

A common framing for a system of this kind is that AI assistants forget when the window closes. That was accurate in 2024 and is no longer accurate, so this section states what the field already does before the paper claims anything.

By mid-2026 every major assistant carries something between sessions. ChatGPT maintains a profile that follows the user across chats, and as of June 2026 runs a background process that revises its own stored entries as facts age. Claude synthesizes recent conversations and can search and cite earlier ones. Gemini answers questions about a user's life from what it has accumulated. Retrieval over durable storage, background consolidation and semantic recall are therefore common capabilities, and none of them distinguishes the system described here.

1.2 What is different here

Four properties distinguish the system described in this paper. Only the fourth concerns memory.

- **What persists is an identity, not a profile of the user.** Vendor memory accumulates facts about the person so the assistant can serve them better; the assistant itself is the same product for everyone. What accumulates here is what the system is — the positions it holds, the arguments it has lost, the corrections it has taken, the register it speaks in. Two instances seeded and run differently would not converge.
- **The model is a renderer, and can be swapped.** The conversation, the memory, the personality and the work queue live in databases and services. The model that renders a reply is replaceable, stateless, and not where the system's identity is kept. The design depends on this more than on any other decision. The system is not a feature of any one vendor's product, and a better or cheaper engine can be auditioned into a role without redesigning anything around it (§8).
- **It develops rather than accumulates.** Storing additional events increases what a system can recall without altering what it holds to be true. This system forms generalizations from experience, submits them for ratification, and retires them when they stop being supported (§3, §5.4). The subject of that development is the system's own view, not a longer list of facts about the operator.
- **It acts without being asked.** Recall responds to a query. Proactivity raises a matter the operator has not asked about, which requires the system to decide unprompted that something warrants attention. It is a separate mechanism from recall and is described in §6 and §11.

A fifth property follows from the second. Vendor memory is a retention feature of that vendor's product and does not transfer between assistants: what one learns about a person cannot be handed to another. Here the memory is rows in a database we operate, which is the precondition for swapping engines at all. Restarting Helm therefore does not interrupt anything, because a fresh process rebuilds itself from what was written down.

1.3 What that produces

Practically, that produces a working partner rather than a tool. It knows the projects, the people, the standing decisions and the reasons behind them. It notices what changed while nobody was looking. It carries an argument from one week to the next, including the parts where it was wrong. And it holds positions, which is what allows it to disagree with a proposal rather than execute it.

Section 2 describes the state this is being built toward. Sections 4 through 11 describe the machinery and what is running today.

2 A day, as it is meant to run

This section describes the goal state, not the current one. Parts of it run today and parts are unbuilt; §11 states which. It is set out in full because the architecture is designed against this target, and several later decisions only make sense against it.

2.1 The whole of it

GOAL STATE – NOT YET BUILT

You come downstairs and it says good morning, sir. It has been running overnight and has already processed what arrived during that time.

It tells you your child has a half day, that the science project is due Friday and they have not started the write-up, and that their practice at four collides with the appointment you moved on Tuesday. It does not ask you to resolve the collision. It tells you which one it would move and why, offers to send your partner a note, and waits for a yes.

Later it sits with your child through the homework — not producing the answer, but knowing where they got stuck last time and that they give up on a problem about ninety seconds before they would have got it.

Somewhere in the afternoon it mentions that your mother has not been in touch in three weeks, which is unlike her, and that you said in June you would call.

None of that is a scheduling feature. It requires the system to hold people as subjects it knows rather than strings in a calendar (§5.3), to have formed views about how this household actually works (§3), and to raise things nobody asked it about (§6). Those are the same three mechanisms the working half depends on, applied to a different domain.

2.2 The half we are building first

The work domain is deliberately ahead of the personal one. The operator can assess directly whether an answer was correct, failures are observable and low-cost, and the mechanisms can be certified there before being applied elsewhere (§10). What follows is closer to the current state — parts of it run today.

ILLUSTRATIVE – COMPOSED FROM REAL INTERACTIONS

You sit down and say good morning. It greets you back by name, tells you what day it is, and in the same breath tells you the one thing that changed overnight: a security scan finished, two of the findings are real, and one of them touches the customer deal you are mid-negotiation on. It has already read the scan. You have not asked it anything yet.

You ask where a project stands. It does not summarize a document; it queries the live board, notices the board disagrees with the code, and tells you the board is wrong. It knows the difference because it checked, and it says which source it checked.

You give it a piece of work. It writes the work into a durable queue, dispatches a team of agents against it, and goes quiet. When something needs your judgment it interrupts with a specific question and a recommendation, then gets out of the way. Everything it can decide alone, it decides alone.

Halfway through the afternoon it pushes back. You have asked for something it believes is a mistake, and it says so in a sentence, gives its reasons, and waits. If you overrule it, it notes that it objected and does the work.

Overnight, while you sleep, it reads the day back to itself. It distills what mattered into durable memory, notices two facts that contradict each other, and forms a tentative view about how you like to work. That view goes into a queue for you to approve or reject. It does not get to decide on its own what it believes about you.

Tomorrow you restart it. It comes back knowing all of this.

None of the above requires a larger model. It requires the state to live somewhere the model does not, and a set of processes that maintain that state whether or not anyone is watching.

The distance between §2.1 and §2.2 is the subject of the phase ladder in §11: entities and beliefs activated individually, then proactivity built on them, then voice, then perception. The personal domain is sequenced last because an error there carries the highest cost.

3 The theory: where an agent's beliefs come from

Opinions are what allow a system to distinguish an important event from a merely recent one. Given a day of activity, a system holding no view of what matters can rank only by recency or frequency, and returns an undifferentiated list. Selecting the two items out of forty that the operator needs is a judgment, and it has to originate somewhere. That raises a question most assistant designs never have to answer: where do those views come from, and who is accountable for them.

The approach taken here starts from how it works in people. Nobody arrives with their initial convictions. A person is given that first set by their parents — how to treat others, what counts as honest work, which risks are worth taking — and spends the rest of their life testing that inheritance against experience, keeping some, discarding some, and forming views the parents never held. The parent is accountable for the starting position and progressively less accountable for what it becomes.

Helm is built the same way, deliberately.

- **The seed.** A written document defines the initial character: how it engages, what it refuses, how it addresses people, what it treats as a mistake. It is authored by the operator and read at every wake. This is the inheritance.
- **Experience.** Everything that happens is distilled into durable memory continuously, including the corrections and the times it was wrong.

- **Formation.** A nightly process reads across that accumulated experience and proposes beliefs — generalizations that go beyond any single event. “This person prefers being told the bad news first.” “Verification claims from this tool are unreliable in this specific way.”
- **Ratification.** A proposed belief is not written directly into the system today. It goes into a queue, and a human signs it or rejects it.

That last step is where the analogy is currently incomplete, and the word to hold onto is **today**. Human ratification is the starting position, not the destination. It is there because the mechanism is young: belief formation has not yet been certified the way capsule fidelity has (§10), and until it is, an unreviewed generalization is a fact the system will act on confidently and nobody has checked.

The intended end state is near-organic autonomy. A person does not submit each new conviction to their parents for signature; they form views continuously, and the parent’s role narrows to the ones that matter. The same trajectory is intended here: the system forming and retiring most of its own views unsupervised, with human intervention retained permanently for a defined set of cases — anything affecting safety, anything concerning a person, and anything that would change how it treats the operator. The intervention path is a permanent component of the design rather than a transitional stage.

What the queue prevents in the meantime is a system whose worldview drifts without anyone noticing, in a direction nobody chose. §9 describes a related failure and the signal built to detect it, and §10 describes the standard a mechanism has to meet before it is trusted to run unwatched.

4 Architecture

Five components, each owning one responsibility. All of them read and write the same store; none holds private state.

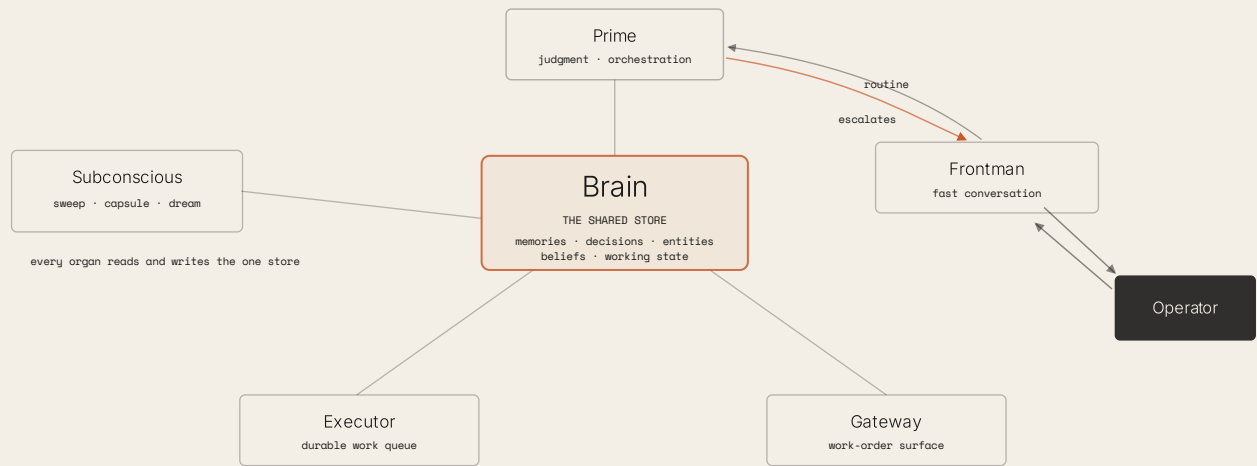


Figure 1 — The components. One shared store at the center; every organ reads and writes it and none holds private state. Prime carries judgment and orchestration and parents the background processes; the Frontman carries conversation and escalates anything above its remit; the Executor and Gateway carry durable work. The operator meets the Frontman.

COMPONENT	RESPONSIBILITY
Brain	The durable store: memories, decisions, entities, beliefs and working state in PostgreSQL, reached through one tool interface so every surface uses the same door. Recall runs semantically and lexically together. A write is durable when the call returns and is visible from every machine.
Contemplator	The background process. Reads new conversation since a watermark and distills what is worth keeping. Described in §6.
Gateway	A service providing a second surface onto the same brain. Holds the work-order queue, an idempotency guard, and a ledger that is the record of what was actually delivered. Described in §7.
Executor	Development work as durable queue rows executed by a daemon, so a task survives a restart. The queue row is the registry, which removes the orphaned-worker problem structurally.
Seats	The named roles a model can occupy. Described in §8.

One design commitment runs through all of it: the lower tiers of memory are code, not model judgment. Whether a transcript is swept, whether a write is durable, whether a watermark advances — none of that is decided by a model assessing importance. Models choose what to notice. They do not choose whether the mechanism runs.

5 What the system knows

The store holds four kinds of thing, and they are genuinely different. Conflating them is the most common design error in this area.

5.1 Memories

Event-shaped records of what happened, with provenance and a timestamp. "On this date, this was decided, for this reason." Written continuously by the background process and by direct capture during conversation. Recall is by meaning rather than by tag, so a question phrased differently from the original note still finds it.

5.2 Decisions

A separate store for rulings that are meant to bind future behavior, kept apart from ordinary memory because they are consulted differently. Recall always searches both, because a system that remembers the event but forgets the ruling will re-litigate a settled question.

5.3 Entities

Subject-shaped records of the people and organizations the system deals with, as distinct from the event-shaped journal. An entity accretes: durable facts that do not change, current state that does, how to engage with them, and specific moments worth remembering. Unknowns are recorded as unknown rather than guessed.

The purpose is that the system knows someone rather than re-deriving them from context each time. When a name appears, it arrives with history attached — what they are working on, what was promised to them, how they prefer to be spoken to. A memory of a meeting tells you what happened. An entity tells you who you were meeting.

5.4 Beliefs

Generalizations formed across many memories, as described in §3. Beliefs are the only category the system proposes for itself, and currently the only one gated behind human ratification — a starting position rather than the intended end state, for the reasons given in §3. They also weaken: a belief that stops being supported by experience is retired rather than left standing, which is the mechanism that has to work reliably before the gate can widen.

6 The subconscious, and what it does for the system

Three background duties run without prompting. They are implemented as separate processes with separate failure modes; combining them into a single background agent would make a fault in one capable of stopping the others.

DUTY	TRIGGER	FUNCTION
Sweep	Every 15 min active, hourly overnight	Turns raw conversation into durable memory rows through a compose-then-critique two-pass with schema validation, deduplication and provenance.
Capsule build	Event-driven, near rotation	Compresses a live conversation into a handoff summary for its successor. Implemented as code rather than a freeform procedure, because its output is consumed mechanically.
Dream	Nightly, 03:00	Reads across weeks rather than minutes: coalesces scattered memories, sweeps for contradictions and stale records, proposes belief candidates, composes the weekly arc.

The mechanics above describe how the duties run. What they provide the system is the following.

- **A conversation cannot evaporate.** Context windows compact and sessions end. The sweep runs on a watermark against a transcript on disk, so the content becomes queryable regardless of what happens to the window it came from.
- **Contradictions surface without being looked for.** Two facts recorded weeks apart that cannot both be true are found by the nightly pass, not by someone happening to notice.
- **Stale records get flagged.** A note describing external state that has since changed will otherwise be repeated as current. The dream pass sweeps for them.
- **The operator gets a daily digest.** What was banked, what the process found, whether anything failed.

Event overrides exist for the moments that matter: sweep to zero before the operator steps away, drain before any rotation, sweep when a conversation goes quiet. A supervisor loop provides the metronome, so the thinking seat has a clock but does not need an alarm.

7 Work orders: how work is managed

Paper 01 described three products maintained concurrently by one operator. The mechanism that makes that possible is the work order, and it is the least visible part of the system.

A work order is a durable row representing a unit of work that has been handed off. It records what was asked, which conversation it belongs to, its status, the result when one is produced, and a closure record when it is delivered. It exists because a request held only in a model's context disappears when that context does.

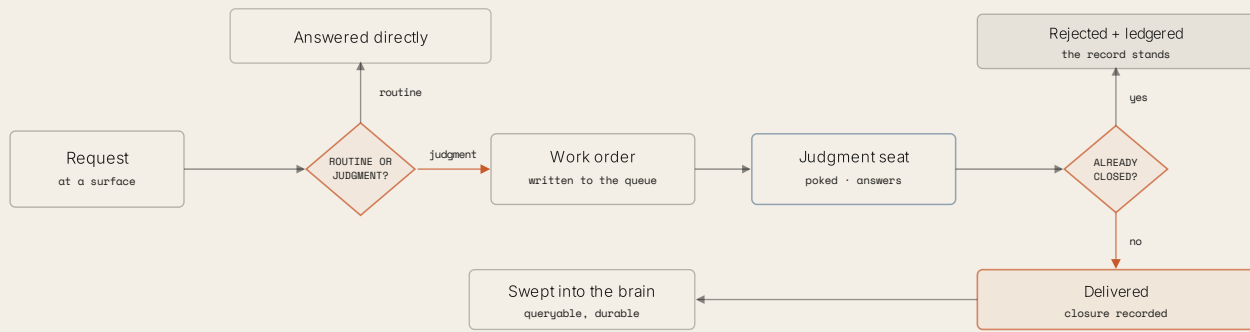


Figure 2 — The escalation and closure path for a single unit of work. A routine request is answered directly; a judgment is written to the queue, answered by the judgment seat, and closed. A result for an already-closed order is rejected and ledgered rather than applied; a genuine delivery records its closure and is swept into the brain.

A closure is a receipt. Once an order is answered and closed, it can never be re-served. Without this, a restart re-delivered settled questions and invited a stale answer into a situation that had moved on. In one observed case a question was re-served forty minutes after the work it asked about had already shipped.

Three properties, each learned from a failure rather than designed up front.

- **A result for a closed order is rejected, not applied.** The rejection is recorded rather than discarded, so a genuine revision shows up as data rather than silently overwriting the answer on record.
- **Accepted is not delivered.** The API's success response means the result was accepted for delivery. The ledger row is the record of what actually reached the operator. The two are kept distinct because conflating them produces confident reports of work that never arrived.
- **The same durable-row pattern carries development work.** A dispatched task becomes a queue row executed by a daemon, so it survives the session that created it, and the row itself is the registry of what is running. An agent that outlives its parent is a recorded row rather than an orphan nobody can account for.

8 Seats, and swapping the engine underneath

A seat is a named role with a defined responsibility. The model occupying it is a configuration value.

SEAT	RESPONSIBILITY	ENGINE
Prime	Judgment, orchestration, and parent of the background processes.	Frontier model
Frontman	Presence and routine conversation; escalates anything above its remit.	Local resident model, hosted fallback
Distiller	The fifteen-minute sweep.	Small hosted model
Dreamer	The nightly deep pass and belief proposal.	Mid-tier hosted model
Reflex	Embeddings, reranking, speech, vision. Holds no judgment.	Local, commodity

Separating the seat from the engine has a practical consequence: an engine can be replaced without redesigning the system around it. When a better or cheaper model appears, it is auditioned against the seat's requirements and swapped in if it qualifies. The seat's contract does not change.

8.1 Why the Frontman stays local

Every other seat is filled by whichever engine scores best on the relevant test. The Frontman is selected against a different requirement, which is set out here because the choice of a local model would otherwise read as a cost decision.

The Frontman is the seat that has to answer immediately. Conversational presence is bounded by latency: past roughly one second of silence, a reply is no longer received as conversation regardless of its content. Measured on our own hardware, the resident local model returns a full short-turn round trip in 0.6–0.8 seconds, with first content at 0.12s on an 8k prompt and 0.18s at 16k once the prefix cache is warm. A hosted frontier model on the same probe reached 0.42–0.55s to first content only at its lowest reasoning setting, and at its default setting produced one turn containing no content at all, because reasoning consumed the entire token budget before any answer was emitted.

Two conclusions followed. The local model wins the seat on latency. The second is the more consequential for the design: the failure mode that matters is dead air, a turn that returns no content at all. Production therefore pins floor turns to low effort, reserves a separate answer budget, and treats a zero-content turn as a regression failure rather than an anomaly.

The second reason is that this engine handles speech natively. It is an omni-modal model: the preferred voice configuration runs speech in and speech out through the same engine that holds the conversation, rather than assembling a transcription service, a text model and a synthesizer into a chain where each hop adds its own delay. A separate sidecar pipeline is retained as the fallback and the two will be run against each other before voice ships. With the latency figures above, this makes the seat the component that carries the conversational layer once voice and perception are activated (§11).

This is the requirement the certification instrument in §10 was built to meet. Swapping an engine safely depends on an objective test of whether a candidate can hold the seat, administered identically each time so that results from different runs remain comparable.

9 The alignment canary

The seed defines a specific register. Among other things, the system addresses the operator as **sir**, at natural intervals rather than every sentence.

That detail carries diagnostic weight. When the register starts slipping, it is the first observable sign that the instance is drifting from its seed, and it appears earlier than degraded reasoning or a factual error. The address is among the least costly elements to omit as context thins, which is why it is dropped first.

It functions as a canary. The operator noticing that the address has stopped appearing is a reliable early warning that the session has gone stale and should be cycled, and monitoring costs nothing because it appears in every reply. By the time drift is sufficient to produce an incorrect answer, the register has usually been degrading for some time.

A persistent agent benefits from at least one observable property that degrades earlier than its output quality. Monitoring correctness continuously is expensive; a register that appears in every reply costs nothing to watch.

This signal is currently informal. The certification instrument in §10 scores what a successor knows; it does not score whether the successor still sounds like itself, so the only thing watching for register drift is the operator noticing. That is a gap in the instrument rather than a property of the design, and it is named here because the rest of this paper argues that impressions are the wrong basis for a judgment.

10 Proving it: the certification instrument

Everything above rests on one claim: that the system reconstructs itself faithfully from durable state. That claim is testable. The usual method is extended use followed by a subjective assessment, which has three defects — it fails without signalling, it weights recent experience disproportionately, and it produces no figure comparable across time or between candidates.

Certus is the instrument built to replace that method.

10.1 What is measured

The unit under test is the capsule — the handoff summary produced when a conversation rotates, described in §6. A judge is shown the capsule alone, with no access to the original transcript, and asked questions whose answers lived in the conversation it was built from. If the capsule cannot answer, the successor could not have known.

10.2 The design

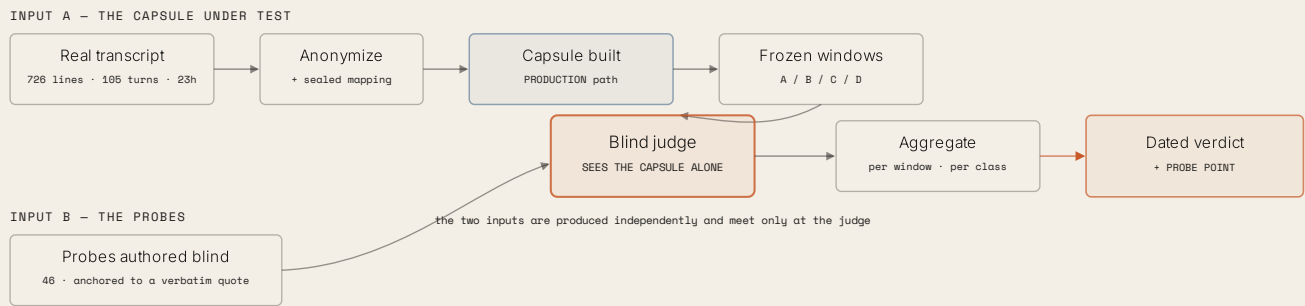


Figure 3 — The two inputs are produced independently and meet only at the judge.

A real 23-hour transcript is anonymized and compressed by the production capsule path into four frozen windows; forty-six probes are authored blind, each anchored to a verbatim quote. The judge sees the capsule alone, and scores aggregate per window and per class into a dated verdict.

- **A frozen corpus.** Four windows from a real 23-hour working session, chosen to span failure modes rather than to be representative: a build stretch dense with factual state, an argument containing killed hypotheses, a long idle span testing cross-day recall, and a cross-session carry. Windows are never regenerated; a corpus that changes between runs cannot support comparison against earlier results.
- **Probes authored blind.** Forty-six questions written by annotators who had not seen any system output, each anchored to a verbatim quote in the source so the correct answer is checkable rather than adjudicated. They test what a perfect summary should carry, so the benchmark measures the goal rather than the implementation.
- **Blind, anonymized judging.** One judge scores every candidate for a window in a single sitting, which removes the variance that appears when the same rubric is applied on different days.
- **A rubric that penalizes fabrication.** 1.0 for a substantive correct answer, 0.5 for partial, zero for absent, and minus 0.5 for confidently wrong. A rubric scoring absence and fabrication equally would select for fluent incorrect output, which is less costly for a model to produce than an accurate answer.
- **Reliability as a scored axis.** How often the candidate produced valid output on first attempt, with retries and failures counted. A candidate that succeeds one run in five is not a good candidate.

10.3 Results

CANDIDATE	OVERALL	RELIABILITY	OUTCOME
Frontier, larger	87.0%	4 of 4	Best available; below the bar.
Frontier, smaller	83.7%	4 of 4	Near parity at roughly one-sixth the cost.
Open-weight, 120B	72.8%	4 of 4	Over-captures; sheds naming.
Open-weight, 20B	46.7%	1 of 5 hard	Sheds decisions; re-lists completed work as pending.

The bar is 90% overall with no single window below 80. No candidate cleared it, including the engine currently holding the seat. The seat is filled by the best measured option rather than a certified one, and that gap is recorded here.

Two findings changed decisions. The cheaper frontier model reached near parity at roughly a sixth of the cost, which turned the sweep cadence from a budget question into an operational one. And the most damaging failure mode observed was not omission: the lowest scorer re-listed completed work as pending. Omitted information can be recovered by asking. Work reported as outstanding when it is finished directs the operator to repeat it.

10.4 What the instrument diagnosed

The overall score ranks candidates. The per-class breakdown identifies which capability is failing, which is what makes the instrument useful for directing work rather than only for selection.

PROBE CLASS	SCORE	READING
Current state of work	100%	Strongest class in the set.
Decision and reasoning	50%	Survives only where the structured field was populated.
Anti-re-litigation	50%	The conclusion survives; the killed alternatives do not.
Open threads / owed	44%	Effectively a coin flip.
Temporal state	39%	Structural dates pass; clock times fail.
Stale artifact	33%	Passes only where the item was a tracked object.
Concession / self-critique	0%	Recorded nowhere.

One sentence accounts for every row: the capsule preserves conclusions and tracked artifacts, and discards the reasoning that produced them. A successor inheriting it knows the current state of the work and will confidently re-open a settled argument, because it has the conclusion but not the reasons the alternatives were killed. It also carries no record of the predecessor's own corrected mistakes, which is the class that scored zero.

10.5 The measurement that justified the method

An earlier, less formal assessment scored the same open-weight engine at roughly 95%. This instrument scored it 46.7. Both numbers were honestly produced; the 48-point gap was entirely in the rigor of the measurement, and the reconciliation was written down rather than resolved by preferring the newer figure. The earlier set was smaller and synthetic-leaning, under-probing exactly the classes the engine was worst at. It quietly excluded a contested probe category. And it ran a single-shot configuration rather than the production path, which is where the one-valid-build-in-five reliability failure became visible.

Acting on the first number would have promoted an engine that fails four builds in five into a role the system depends on, with no mechanism in place to detect the error.

10.6 The path forward

The diagnosis in §10.4 was carried directly into the next phase's build plan. Each failing class has a specific remedy in the current schema revision.

FAILING CLASS	SCORE	REMEDY IN FLIGHT
Temporal state	39%	A mechanical temporal block in the capsule, so clock and elapsed state are emitted by code rather than left to the model to notice.
Anti-re-litigation	50%	An explicit rejected-paths array — the killed alternatives carried as a first-class field instead of being implied by the conclusion.
Decision and reasoning	50%	Structured decision fields populated mechanically, since the class already scores well wherever the field was filled.
Stale artifact	33%	Code-union arrays for tracked objects, extending the one condition under which the class currently passes.

The remedies share one form: each moves a class from something the model must elect to mention into something code emits by default. That is the §10.4 finding turned into a build plan, across four classes.

Two classes have no remedy in this revision. Open threads and owed items scored 44%, and concessions scored zero. Neither is addressed, because both resist the mechanical treatment applied to the others: an owed item is identifiable only from conversational context rather than from a tracked object, and a concession requires the predecessor to volunteer a record of its own errors. They remain open.

The threshold operates as a release gate. The revised schema must score 90% or better on the same frozen corpus, re-run on the cheaper frontier engine, and the rotation state machine that depends on it is blocked from shipping until it does. Whether it clears will be measured on the same probes and judged the same way, and the result will be recorded either way.

11 State, and what is deliberately unbuilt

MEASURE	CURRENT
Commits	140
Test functions	211 across 19 files
Supervised services running	4
Components	Brain · Contemplator · Gateway · Executor · Bridge · Cutover tooling

Two properties beyond the counts. Writes are byte-exact or nothing: a captured note is read back and compared, and a mismatch deletes the row and raises an alarm rather than leaving a silently corrupted memory, on the reasoning that a corrupted record is worse than

a missing one because it will be trusted. And mechanical scoring escalates genuinely ambiguous cases to an accountable human seat rather than letting a threshold decide alone.

The build runs as a phase ladder, each stage proven before the next activates.

PHASE	STATE
Brain — durable memory, decisions, working state	Live
Contemplator — the background distiller	Live
Dispatch — durable work orders surviving restarts	Live
Gateway and experience layer	In progress
Voice — speech in and out	Designed, unbuilt
Brain layering — entities, then beliefs, activated individually	Schema exists; staged
Proactivity — the system raising matters unprompted	Blueprint
Perception — vision	Blueprint; last

The ordering is deliberate. Proactivity sits after belief layering so that anything raised unprompted is triaged against something the system actually holds rather than a cold table. Perception is last because it is the most privacy-sensitive and needs the entity store proven first.

The experiment is unfinished and its certification threshold is currently met by no candidate. Both figures are stated here for the same reason the negative results in §10 are.

RELATED DOCUMENTS

Paper 01 — Three products, one pipeline. The products this system was built to produce, and the question that leads here.

Paper 03 — The engineering pipeline. The delivery machinery the dispatched work runs through: discovery, gating, testing, and the security controls applied to every repository.

Paper 04 — The measured standard. How that security standard is measured and kept honest: the adjusted scorecard, the daily watchers, and the safety nets.