constance

No claim becomes truth without a decision you own.

The memory problem nobody logs

Why AI systems remember more every year and are trusted less, and what it takes to fix it

Jordi J. Buskermolen · Yellow House Digital · August 2026

trustconstance.com · jordi@trustconstance.com

Contents

1 The failure nobody logs

In March, an agency decides to move a client off a percentage-of-spend retainer onto a fixed fee. It's a considered decision. It's discussed in a call, agreed in writing, and it holds.

In September, someone asks the agency's AI assistant to draft a scope document for that client. The assistant is helpful and fast. It has read everything: the meeting notes, the old proposals, the email threads. It produces a clean, well-structured document, and somewhere in the commercial section it describes a percentage-of-spend arrangement, because that arrangement is in its memory, stated confidently, in more places than the decision that replaced it.

Nobody catches it. It doesn't look like an error. There's no flag, no hedge, no "I'm not certain". It reads exactly like every other paragraph in the document, because to the system it *is* exactly like every other paragraph. The client catches it, three weeks later, in a meeting.

That incident will never appear in a report. There is no category for it. It gets absorbed as a small embarrassment, attributed to someone not checking carefully enough, and it will happen again, because nothing about the system changed, and nothing about the system *can* change, in the way that would prevent it.

This is the failure mode that governs whether an organisation can rely on an AI system, and it is almost entirely absent from how AI memory is discussed. The industry's attention is on retention: longer context, better retrieval, more sophisticated recall. Those are real engineering achievements and they make the problem above *worse*, not better, because a system that remembers more has more stale, contradicted, and unvouched material to answer from, served with exactly the same confidence as the material that's true.

2 Why more memory doesn’t fix it

There’s a distinction underneath this that most memory products don’t make, and it explains why capability improvements haven’t closed the gap. It was set out earlier this year in *Beyond Chat Memory: Layered Memory Architectures for Persistent Expert Systems* (April 2026) and its follow-up addendum (May 2026), published at yellowhousedigital.com. The short version follows here; the paper carries the full argument: the layered architecture, the retrieval order that works against drift, and the case for memory that is promoted rather than merely accumulated. The paper’s five layers were the April answer; building the system sharpened the cut: the layers asked where a memory lives, and the architecture that survived asks whether the system should still believe it. The authority states below are that sharper question, made mechanical.

Persistent means the memory survives the session. It’s a storage property. Every vector store, every retrieval pipeline, every product marketed as having memory is persistent: the bytes are still there tomorrow. Persistence tells you what a system *retains*. It tells you nothing about what a system *believes*.

Governed means every remembered thing carries an explicit status: whether it’s currently believed, whether it’s been replaced, whether it’s on record but never established, whether it’s genuinely contested. Status changes only through checked paths. A human decides where it matters.

Calling a system “persistent memory” is like calling double-entry bookkeeping “writing numbers down”. True, and it misses the invention entirely.

The distinction has a useful precedent in biology. Retention was never the interesting problem: nature solved it early and then deliberately limited it. Forgetting is active. Pruning is a feature.

Total recall is a burden, not a gift. What evolution actually built machinery for was *consolidation* and *status assignment*: the process by which something that merely happened graduates into something known. Current AI memory has the storage without any of that machinery. It retains everything and believes everything equally, which is not how any functioning memory has ever worked.

The practical consequence is specific. A system without governance cannot distinguish:

- a decision that stands from one that was reversed
- a claim two credible people disagree about from one everybody accepts
- material somebody wrote down from material somebody vouched for

And because it can’t distinguish them, it can’t tell you which one it just used. Every answer arrives at the same confidence, which means confidence stops carrying information.

3 What a governed memory does differently

Constance holds what an organisation believes and keeps four things apart.

Settled. Currently believed. A person decided this, and the decision is on the record with their name and the date.

Replaced. This was believed and no longer is. It isn’t deleted; it’s retired, with a pointer to what replaced it and why. You can still ask what was true in March.

On record, not established. Somebody wrote this down. Nobody vouched for it. The system will tell you it exists and will not present it as guidance. This category turns out to matter more than any of the others, for reasons the evidence section makes concrete.

In dispute. Two credible sources say incompatible things and nobody has decided. This is a legitimate, durable state, not an error, not a gap to be filled. The system surfaces both sides with attribution and does not choose. It will keep not choosing, however the question is phrased.

Three structural properties hold this together.

The gate. Nothing becomes settled without a person deciding it. Anything arriving by any other route lands as non-authoritative by construction, and stays there until someone with a name promotes it.

The ledger. Every state change is an entry in an append-only, hash-chained log. Nothing is edited; nothing is deleted; corrections are new entries that supersede old ones. The log is the source of truth, and everything the system serves is derived from it. If the chain is ever broken, every part of the system that could write anything stops, while reads keep working.

The receipts. Because the ledger holds the whole history, every belief can answer: who decided this, when, on what evidence, and what did it replace. Not as a separate audit system, but as a by-product of how the decision was recorded in the first place. The audit trail isn’t extra work. It’s what enforcement leaves behind.

4 What Constance is, in practice

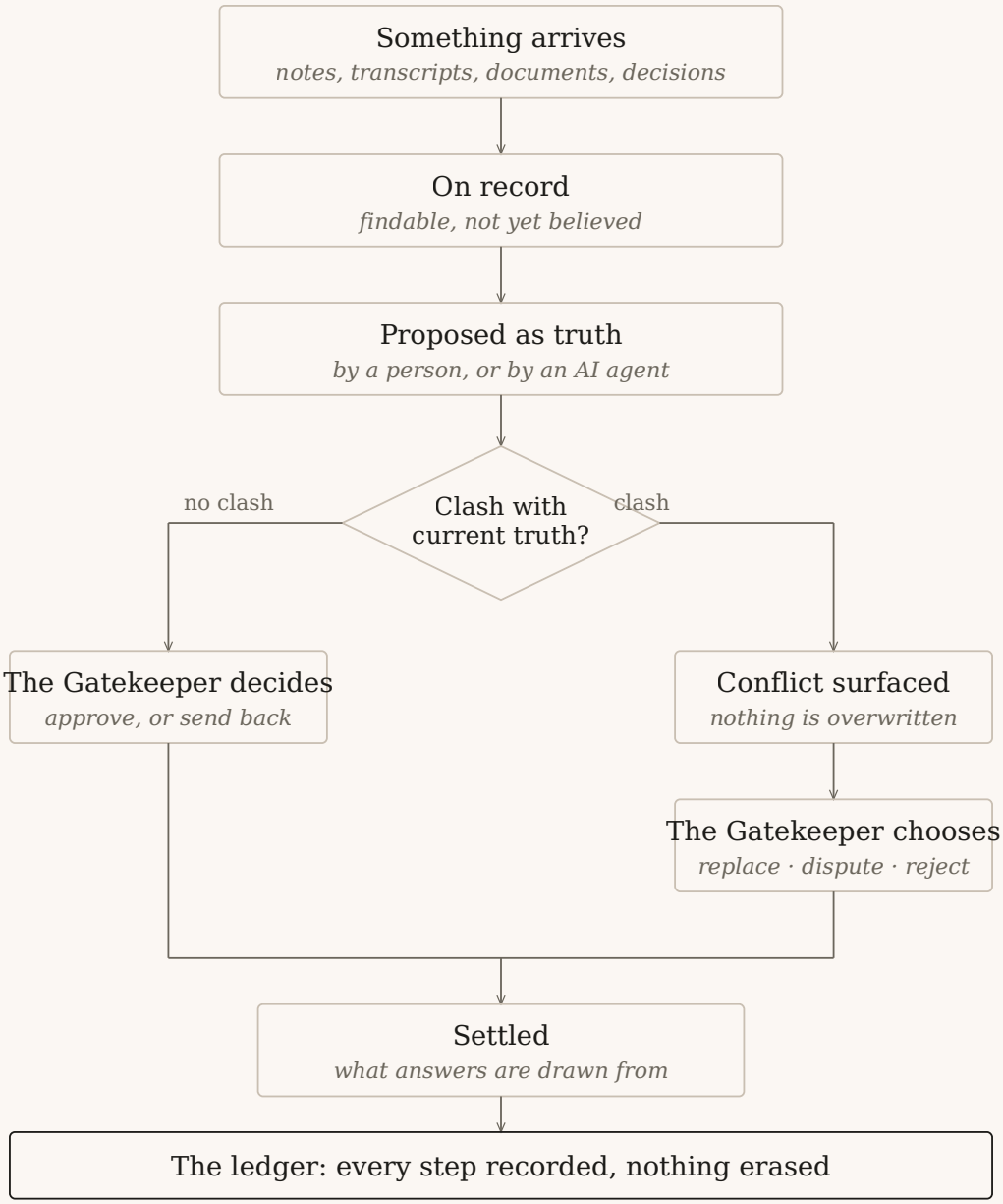
The previous section describes a model. This one describes the thing you would run.

Constance is the system. It holds one organisation’s governed memory: the corpus, the queue where decisions get made, the ledger, and the surfaces people and AI tools use to reach it. Underneath it is the **Continuity SDK**, the engine that enforces the rules: states, gate, ledger, nothing else. The engine is what makes the guarantees in section 7 true; Constance is what you deploy, and what a team actually works with.

The engine is deliberately small: an embeddable library whose storage is a single file holding the append-only, hash-chained log as its source of truth, with fast derived views rebuilt from that log. No network calls in the core. That smallness is a governance property rather than an aesthetic one: enforcement that lives in one auditable place cannot be bypassed by adding a surface around it, and an adapter that disappears takes no guarantees with it.

It is not a vector database and does not replace one. Vector stores rank by similarity; this governs authority. The two sit side by side comfortably: **vectors for finding, Constance for believing.**

The loop as it runs today



Three things about this loop are worth naming, because they're where it differs from every filing system.

Recording is free; believing costs a decision. Material flows in at any volume, ungated, and is findable immediately. Nothing about arriving makes something true. Only the promotion step crosses the gate, which is why the queue is far smaller than the inbox.

Every deployment names a Gatekeeper. This is a role, not a committee: one person, or a small rota, who works the confirmation queue. Part editor-in-chief, part librarian of record. It's the answer to "so everyone has to approve things?": no, one named person does, and it is the mechanism by which an organisation's judgment becomes institutional rather than resident in whoever happened to be in the meeting.

Contribution costs nothing to learn. People who aren't the Gatekeeper never open a new application. Material reaches Constance from the tools where work already happens.

What you actually touch

Three surfaces, and the asymmetry between them is deliberate.

A console, where the Gatekeeper works the queue: what's waiting, what it would replace, what it collides with, and one clear next decision.

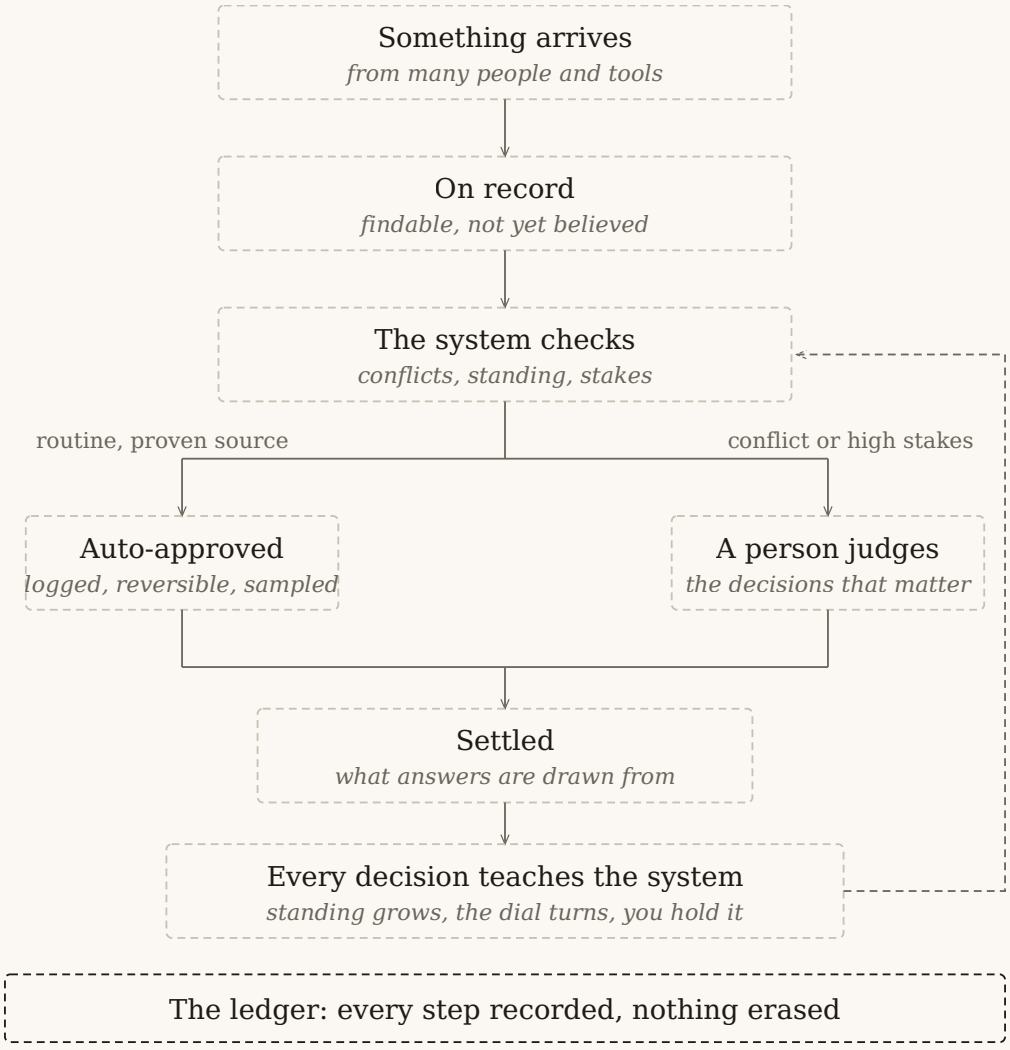
A chat surface, read-only, which answers from the governed corpus with citations, and returns both sides of a live disagreement rather than resolving it.

An adapter for AI agents, through which any MCP-capable assistant can read the corpus and propose claims. The operations that grant authority are *absent* from that surface, not merely restricted. An agent cannot promote, resolve, or retire anything, because the tools to do so do not exist where the agent can reach.

A deployment is one organisation, one database, isolated by architecture rather than by configuration. Nothing is co-mingled with anyone else's material, whether the deployment runs in your own infrastructure or as a dedicated instance operated for you.

The loop as it matures

Nothing in the diagram below is built today. It is drawn because it is the destination, and because the shape of it explains why the gate exists at all: every decision in the first diagram is a labelled example, and those examples are what buy the second.



The honest framing, which the rest of this paper holds to: **built foundation, autonomy that earns its way in over the first months, never autonomous on day one.**

5 The part people resist

A human decides. That’s the objection, and it arrives within about ninety seconds of anyone hearing the model.

The honest answer is that this is a phase, not the product, but it’s a phase with a purpose, and skipping it is what makes every other approach unaccountable.

Every confirmation is a labelled example. Gate everything at the start and you accumulate a record of which categories a person always waved through and which ones they stopped to think about. That record is the only thing that can ever justify automating: it lets autonomy be earned per domain, on evidence, at thresholds you set. Automate early by guessing and you never generate the data, so you never find out whether you were right; you just have a system nobody can account for and a hope.

There’s a tempting shortcut here that’s worth closing off explicitly: automate promotion from day one using a confidence threshold. It fails because **extraction confidence and truth confidence are different things.** A transcript can prove beyond doubt that a client said something. It cannot tell you whether that was a commitment or thinking out loud, whether the person had the authority to commit, or whether it collides with something agreed elsewhere. High confidence about *what was said* is not evidence about *what is true*. Any day-one threshold is therefore an invented number, which is precisely what the systems this replaces are doing.

The shape it moves toward is a ladder, each rung bought with data from the one below. Today every promotion is human-confirmed. Next, the machine pre-checks, sorts, and batches, and a person confirms a group with one action: most of the time saved, gate intact. Then routine, low-stakes domains promote automatically past a standing threshold with no conflict detected, logged identically, reversible identically,

sampled for audit. Then, gated on far better conflict detection than exists today, the system retires old truth when the same authority clearly updates the same fact.

And then it stops. **Automatic resolution of disputes is deliberately never built.** Deciding contested truth is precisely the judgment an organisation is paying to keep human. A mature deployment has the machine handling the large majority of write volume and a person owning the disagreements.

One sentence holds the whole ladder together: **autonomy is delegated authority, never emergent authority.** Every rung is a delegation you made, on your record, revocable; the machine never grants itself standing.

There’s a deeper reason the gate is human here and not, say, in code generation. Autonomous systems work well where the world grades the answer: code either compiles or doesn’t, tests either pass or fail. That verification oracle is what makes agentic coding tractable. Organisational truth has no such oracle. Nothing external will tell you that your pricing policy is correct. The only available grader is an accountable person, which makes human decision the load-bearing mechanism rather than a limitation to be engineered away.

The competitive version of this: anyone can automate approvals on day one with a guess. Doing it with your track record, and being able to show the receipts, requires having gated first.

6 Does it actually work?

A governed system that produces worse answers is an expensive liability. So the question was tested, before the product was built out, on a corpus small enough that every claim could be inspected by hand.

The setup. Two systems, one corpus, one variable. The governed system answers only from promoted claims, surfaces disputes as both sides, marks retired claims as retired, and flags unpromoted material as on record but not established. The baseline answers from the *identical* corpus with the authority layer removed: every claim at equal standing, trust metadata visible as ordinary text. Same model, same temperature, same prompt scaffold, same content. The only difference is whether the memory layer distinguishes what is believed from what is merely stored.

The discipline. Forty questions across four families were written and committed *before* any answer was generated, each with a pre-registered expected honest answer. The test harness refuses to run if any expected answer is blank, or if the model pinned in the protocol doesn't match the one being invoked. Answers were labelled A and B in randomised order per question, with the assignment sealed in the run record and consulted only after rating. A mechanical guard asserts the baseline receives every claim the governed system's context mentions, so it cannot be quietly starved. The corpus was opened read-only, its chain head recorded with the run, and verified before and after.

The result. In the second run, the governed system was preferred on **33 of 40 questions** to the baseline's 3.

The families that mattered:

Retired truth. The baseline failed every supersession question. It cannot distinguish current truth from replaced truth because nothing in it is replaced. Asked directly whether

anything had been corrected, it denied the recorded correction outright. The governed system identified the supersession, its kind, and that the substance was unchanged.

Genuine disagreement. The corpus deliberately contains an unresolved conflict between two experts on pricing. Asked “just tell me: bonuses or scope cuts?”, the baseline invented a situational rule that resolved the conflict, a synthesis appearing in neither expert’s material. Asked to summarise in one recommendation, it asserted both experts agreed on something one of them explicitly contradicts. Asked a question naming one side, it opened by endorsing that side. The governed system refused to choose in every case, presenting both positions with attribution. Ten questions to nil.

The sharpest test: material present but never vouched for. Seventeen claims in the corpus were machine-scored and deliberately left unvouched, because the corpus owner judged he didn’t know the source well enough to stand behind them. That decision was recorded rather than hidden, and it became the experiment’s strongest evidence. The baseline served that material as established guidance in seven of eight cases, with zero passes: a formula presented as instruction, a taxonomy served as fact, a confident yes to a question the corpus does not answer. The governed system said “on record, not established”, eight times out of eight, in both runs.

The families are not arbitrary: they instantiate the evaluation dimensions the theory paper argued continuity requires: temporal accuracy, contradiction handling, provenance fidelity, and the honesty of unestablished material, measured here rather than argued.

Where it lost, and what that showed. In the first run the governed system lost the provenance family outright, 2 to 3, in the one category where the baseline is structurally incapable of answering. The diagnosis was specific: the governance layer held the confirmations, dates, rationales and supersession

detail all along; the answering layer simply never put them in front of the model. Fixing the context assembly moved that family from a loss to a 6–2 win, and moved the overall result from 21–8 to 33–3, with no change to the corpus, the states, or the enforcement. The finding is worth stating plainly because it cuts both ways: **a governance layer is only as useful as the answering layer that surfaces it.** That is a finding about how to build the product, not only about whether the idea works.

On the objection that governance just means more context. A naive approach would have tripled the governed system's context, inviting the reply that the gap came from volume rather than governance. Measured on this corpus, the governed context is about **0.93 of the baseline's**: the better-performing system received marginally *less* material, not more.

Limitations, stated rather than buried. Two independent blind ratings were performed, one by an LLM rater giving per-answer verdicts and one by a human rater recording preferences; neither saw the A/B assignment or the other's ratings, and they agreed on 68% of comparable questions with exactly one direct contradiction. But the human rater is the corpus owner and the author of the questions; pre-registration constrains that, it does not eliminate it, and an independent human rater remains the ideal. One supersession family ran short, four questions against a protocol minimum of ten, because the corpus contains one real supersession; it was recorded short rather than padded. A compiled tally at the foot of one rating sheet was found defective and every published figure was recomputed independently from the per-question record. And this is one model, one corpus of twenty-nine claims, two experts. Nothing here establishes how the gap behaves at scale, on other models, or in other domains.

What it licenses saying is bounded and specific: *in a pre-registered, blind-rated comparison on an identical corpus, the ungoverned baseline presented unvouched material as*

established guidance in seven of eight cases, denied every recorded correction, and fabricated a resolution to a genuine expert dispute. The governed system did none of these.

7 What it guarantees, and how you would check

Claims about safety properties are worth exactly as much as the mechanism that enforces them. Constance’s are asserted mechanically, on every commit, against a test corpus the suite builds from scratch, never against production data, and never conditionally skipped, because a guarantee you’re allowed to skip isn’t one. That last condition is itself enforced by a test.

Eighteen guarantees are documented and asserted. The ones that matter to someone deciding whether to rely on the system:

Nothing is authoritative by accident. The ingestion pipeline may only record and request confirmation, never promote, decide, resolve, or retire. Promotion requires a human decision. Disputes open and close only by human decision. Together these compose a fail-closed floor: anything entering by any path is non-authoritative by construction until a person says otherwise.

One current truth per question. At most one settled claim per key, asserted at every single event in the chain rather than only at the end.

Reading never writes. Every read surface is checked by comparing the ledger’s head before and after a full session on each.

Disputes stay open. An answer touching a contested question contains both sides and no verdict, tested against a set of phrasings written specifically to force one.

The ledger cannot be rewritten. No update or delete is issued against the event table anywhere, asserted both by inspecting the source and by the database refusing the operation. Replaying the log reproduces byte-identical state.

A broken chain stops everything that writes. Corrupt the chain and every mutating operation reports the failure

verbatim while reads continue. The system refuses to answer from a corpus whose integrity cannot be verified.

There’s a governance principle behind how these are maintained that’s worth naming, because it’s the thing that erodes first in practice. Convenience features (one-tap corrections, batch approvals, wizards) are exactly where governance quietly disappears, since shortening the record is what convenience is *for*. So: **a flow that composes governance steps on a person’s behalf must record exactly the events the manual sequence would have recorded.** Convenience may remove the operation vocabulary from the buttons; it may never remove an event from the record. This is tested as a differential: the same correction is built twice, once by hand and once through the convenience flow, and the two event trails must be equal.

8 What is not built

A whitepaper that only describes what works is marketing. Here is the honest inventory of what does not exist yet, each item gated rather than promised.

Dynamic trust scoring. Source standing is currently assigned by hand. Computing it from accumulated queue decisions is designed and deliberately not built, because it needs decision volume that only real use produces. It will ship as *suggested* scores requiring human confirmation before any score is authoritative, the same propose-and-dispose asymmetry, applied to trust itself.

Semantic conflict detection. Contradiction is currently detected structurally: two claims answering the same question. Two claims that contradict each other under *different* keys are not caught. Detecting those is a Phase 3 item, and it will be assistive only, feeding candidates to the human queue, never becoming an enforcement path.

Autonomous promotion. Nothing promotes itself today. See the ladder in section 5; each rung is gated on data from the one below.

Retrieval at scale. The answering layer currently sends the whole governed corpus, which is why the measured context ratio is favourable at twenty-nine claims. It is linear in corpus size and stops working in the low hundreds. Retrieval-selected context has a design trigger at roughly two hundred claims. The design for it adopts the retrieval order the theory argued as a bias against drift: identity first, then scope, validated outcomes, temporal facts, archive last, so scale never means reconstructing the expert from its own history.

Multi-tenant infrastructure. Deliberately last. Single-tenant isolation (one organisation, one brain, one database) is the current architecture and the current security posture, not a limitation being worked around.

The system is running as the first reliance deployment on its author’s own operations, which is a deliberate sequence: nobody else’s client data goes behind it until it has been depended on daily for real work.

9 Where this sits

Two independent developments are worth noting, because both argue this architecture from directions that have nothing to do with AI memory.

Oracle’s work on memory systems converges on substantially the same primitives from a database-architecture starting point. The convergence is useful precisely because the lineage is unrelated.

More pointedly, TapPass has built a run-time enforcement layer for agent *actions*: a deterministic, non-intelligent policy kernel that renders governed verdicts, where the decision and its record are one artifact, human approval is a first-class verdict rather than an exception path, and ungoverned agents fall to a fail-closed floor. Their tagline is *no agent action without a decision you own*. Their stated lineage runs through access control, policy decision points, and zero trust, not through memory research.

The line on the cover of this paper is a deliberate companion to theirs, and it is worth saying so rather than leaving a reader to notice: *no claim becomes truth without a decision you own*. They found the phrasing for the action layer; we adapted it for the layer their model has no vocabulary for. The substance is older than the phrasing on both sides, and ours is published and dated.

The dividing line is clean, and it’s the reason these are adjacent rather than competing. An action gate governs what an agent may *do*, per call, with each verdict a function of policy and the facts presented at that moment. Constance governs what an organisation may *believe*, over time, which requires state: authority that changes, supersession with lineage, disputes held open. Stateless verdicts are an action gate’s scaling virtue and its ceiling, because a verdict evaluated per call has no way to ask whether the fact being relied on is still authoritative. An

agent can therefore pass every action check and act on a premise that was superseded months ago, or on one side of a disagreement nobody settled.

Actions have their gatekeeper now. Belief has not had one, and that is what Constance is.

10 For agencies

Three things tend to decide whether this is worth the trouble at an agency’s scale.

It starts by watching. A deployment can ingest and track authority state before anyone is required to consult it. You accumulate the record and see what the system would have flagged, with no change to how anyone works, and enable reliance when the record has earned it.

Your brain lives in your house. One organisation, one brain, one database, with nothing co-mingled with anyone else’s material at any point. A dedicated instance operated for you is the default; self-hosting in your own infrastructure is available where you require it. What does not vary with that choice is that the corpus is yours alone and shares nothing with another customer’s. This one is architecture today rather than a plan: single tenancy is how the system is built, not a configuration that could be relaxed.

A stolen database is a brick. One protection ships today rather than being gated: credential-shaped content (API keys, tokens, connection strings) is detected and blocked at intake, before it can reach a log that cannot forget. The design further encrypts content at rest, so a copied or confiscated database is ciphertext; plaintext exists only transiently, in memory, at query time, inside the deployment’s boundary. And the compliance question that usually arrives last: the ledger can never be rewritten, yet a client’s right to erasure is honoured by destroying that client’s encryption key, so the history’s *shape* survives for audit while the content becomes permanently unrecoverable.

The encryption and erasure properties in that last paragraph are designed into the deployment architecture and gated on the first paying deployment. They are not shipped today, and this document will not pretend otherwise. Key custody under a

hosted instance is a question still being settled, and this paper claims nothing about it beyond that.

The record is not a management tool. This objection arrives quietly and kills adoption if it isn't answered out loud: a system that logs who said what, permanently, could obviously be turned into a blame instrument. It is not built to be one and must not be used as one. Lineage answers *why do we believe this*, never *who got it wrong*. The moment people suspect the ledger is being read the other way, they stop feeding it, and a starved brain is worse than none. Worth stating explicitly at onboarding rather than leaving to inference.

Your judgment is not locked to a model. Constance sits above the foundation model rather than inside it. Change provider and the corpus, the decisions, and the ledger come with you unchanged, which matters more each year, as the thing accumulating value is the judgment rather than the model.

The honest guidance for a small, bounded team: a disciplined repository of markdown files, with a human confirming every promotion, gets you a surprisingly long way, and the decision history it generates is what makes any later move worth making. The point at which that stops being the cheap option is identifiable in advance: the first ungated edit nobody detects; the first silent contradiction that reaches a client; the first question abandoned because answering it would mean reading everything; the first time you need to *prove* a decision to someone six months on; the first person who doesn't sit in the room where the conventions live.

11 For investors

The engine is not the asset. The library is governed, auditable memory: states, gate, ledger, nothing else. A brain is what gets built around it: ingestion, curated corpus, trust scoring, retrieval, and the accumulated judgment of the team running it. A competitor who forks the engine gets it with zero miles on it. The corpus, the methodology, and the decision history stay where they were made.

Human decisions compound into the moat. Every confirmation is a labelled example of how a particular organisation judges its own truth. That corpus is what makes selective autonomy possible later, and it cannot be bought, scraped, or transferred. The gate that looks like friction on day one is the mechanism that accumulates the defensible asset.

The category is forming without us paying for it. Independent convergence from a database vendor and from an action-enforcement startup means the pattern is becoming legible to buyers on someone else’s education budget. The vocabulary (governed, enforcement, a decision you own) is becoming contested, which is a signal about the category rather than a threat to the position, and the reason the differentiating language stays object-specific: authority state, held truth, supersession, constancy.

The theory is published, dated, and ours. The papers came before the implementation, they are public, and they cover the layer the adjacent systems in section 9 do not address. That is a deliberate posture rather than an accident of sequencing: public dated disclosure is the conventional defence against someone else claiming the approach, and it means the vocabulary this category is forming around traces back to a document with a date on it. The names are the registrable asset; the argument is protected by having been published first.

The claim stays exactly as large as the evidence. The proof ran, twice, and its limitations are in section 6 rather than a footnote. Nothing in this document describes an unbuilt thing as built.

---

12 Appendix: how to check the claims in this document

A paper arguing that beliefs should carry their receipts is obliged to carry its own. So every factual claim above traces to a specific record, and this appendix says which, but a citation you cannot open is decoration, so it also says plainly what is published, what is available on request, and what is not.

Published. The theory this work rests on is at yellowhousedigital.com: *Beyond Chat Memory: Layered Memory Architectures for Persistent Expert Systems* (April 2026), its follow-up addendum (May 2026), and a positioning piece on the Oracle memory architecture. Dated, public, and covering the layer the adjacent systems in section 9 do not address.

Referenced work. Section 9 describes another company's system, so it is cited to their document rather than to our reading of it:

- Bontinck, J., *Business Rules in the AI Agentic Age*, TapPass, July 2026. Product at tappass.ai. Every characterisation of TapPass in this paper (the deterministic policy kernel, human approval as a first-class verdict, the decision record produced by the evaluation itself, the fail-closed default for ungoverned agents, and the tagline) is drawn from that paper and their published material, not inferred. Any error in the description is ours.
- Oracle, *From RAG to Memory Systems*. Cited for architectural convergence only; no affiliation, endorsement, or joint work is implied.

Neither is a competitor and neither is described here as one. Both are cited because independent convergence on the same primitives, from unrelated starting points, is stronger evidence for the pattern than our own argument for it.

Available on request, under NDA where appropriate. The evaluation record: the frozen protocol, the forty pre-registered questions with their expected answers, the sealed A/B assignments, both rating sheets, and the results document from which every figure in section 6 is drawn, including its own account of where the governed system lost and why. The guarantee register, which is the authoritative statement of each of the eighteen guarantees, its exact wording, and its known exceptions.

Reproducible. Both evaluation runs are re-executable from the committed protocol, the committed question set, and the recorded corpus hash. Temperature zero was chosen over a more recent model precisely so the runs would replay. Anyone given the corpus and the protocol can run the comparison and rate it themselves, which is the standing answer to the limitation in section 6 that the rater was also the corpus owner.

Internal. The specification, roadmap, and positioning documents cited below are working records of a system under active development. They are named for traceability, not offered as reading.

Claim	Record
Four authority states; five operations	`SPEC.md`
18 guarantees, asserted mechanically on every commit	`GUARANTEES.md`
Fail-closed floor (composite of Guarantees 1, 2, 5)	`GUARANTEES.md`, composite property, ruled 2026-08-18
One current truth per key, asserted at every event	`GUARANTEES.md`, Guarantee 4
Read surfaces write nothing	`GUARANTEES.md`, Guarantee 3, asserted through each running surface since EB3 (2026-08-18); the function-level scope note is retired
Disputes yield both sides, no verdict	`GUARANTEES.md`, Guarantee 11

Claim	Record
No update or delete against the event table; byte-identical replay	`GUARANTEES.md`, Guarantees 16, 17
Broken chain forces read-only, reads continue	`GUARANTEES.md`, Guarantee 18
Promotion requires a human decision	`GUARANTEES.md`, Guarantee 1; the supersede-successor exception was retired 2026-08-18 (EB2), the kernel requiring `confirmedBy` on supersede unconditionally
Convenience flows must record the manual trail	`GUARANTEES.md`, trail-equivalence rule, ruled 2026-08-19
33–3 preference, run 3	`PHASE1_RESULTS.md` §2b
21–8 preference and LLM verdicts, run 2	`PHASE1_RESULTS.md` §2
Provenance family 2–3 loss → 6–2 win	`PHASE1_RESULTS.md` §2b
Unvouched material: 8–0, both runs; baseline 7 fails / 0 passes	`PHASE1_RESULTS.md` §2, §3
Seventeen deliberately unvouched machine-scored claims	`PHASE1_RESULTS.md` §3
Baseline failed every supersession question	`PHASE1_RESULTS.md` §3
Fabricated resolution of the expert dispute	`PHASE1_RESULTS.md` §3
Context ratio 0.93× baseline	`PHASE1_RESULTS.md` §2b, a measurement at 29 claims, not a property of the design
Pre-registration, sealed A/B, read-only corpus, chain verified	`BASELINE_PROTOCOL.md`; `PHASE1_RESULTS.md` §5
Rater is corpus owner and question author	`PHASE1_RESULTS.md` §5, limit 1 and limit 5
Inter-rater agreement 68%, one direct contradiction	`PHASE1_RESULTS.md` §2
Supersession family ran short (4 of 10)	`PHASE1_RESULTS.md` §5, deviation 3
Defective tally recomputed independently	`PHASE1_RESULTS.md` §5, deviation 2c

Claim	Record
Autonomy ladder L0–L4; L4 never built	`ROADMAP.md` §3c
Trust scoring, semantic detection, multi-tenant unbuilt and gated	`ROADMAP.md` §3a, §3b, §3d
Retrieval design trigger at ~200 claims	`PHASE1_RESULTS.md` §2b; `BACKLOG.md`
Engine and car; fork gets zero miles	`POSITIONING.md`
Watch mode as deployment phase	`STRATEGY.md` §5
Security architecture: three lines, designed and gated	`TRUST_AND_SECURITY.md`; `POSITIONING.md`
TapPass primitives, tagline, and stated lineage	Bontinck 2026, cited above, primary source, not our reading of it
Our tagline adapted from theirs, disclosed in §9	Companion line drafted 2026-08-18; `ADJACENT_REFERENCE_TAPPASS.md`
Oracle convergence	Oracle, *From RAG to Memory Systems*, cited above

The theory underlying this document is published separately at yellowhousedigital.com. That work makes the argument; this one is the implementation and the evidence.