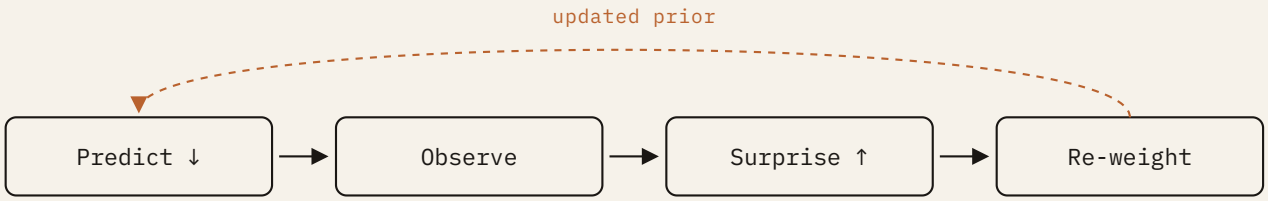
Umma

A thinking partner that builds and operates its own software — and won't hallucinate, forget, or tell you what you want to hear.

Most AI systems are a model wrapped in glue code. **Umma is an architecture.** The frontier model underneath does the raw thinking; Umma does what the model cannot. It holds a goal for weeks without drifting, carries its memory across every session, works while you sleep, and verifies everything it tells you against a source you can check. Underneath all of it is a single idea, run at every scale: **predict, measure the error, weight by confidence.** That one loop is what lets Umma know how much to trust its own conclusions, refuse to make things up, and — when it hits a wall — build the software it needs to get past it.



The same loop runs at every scale — one inference, a week-long investigation, the whole system at once.

What the frontier model *can't* do is not random. It is a specific set of things human minds do that a language model doesn't — and closing that gap is Umma's whole design.

THE DESIGN PRINCIPLE

Grounded in human cognition

Every mechanism in Umma answers one question: *what do human minds do that a language model doesn't?* A model thinks by predicting the next token — powerful, but purely probabilistic, and probabilistic pattern-matching is exactly what struggles to hold an abstraction, carry a goal across weeks, generalize an insight from one field to another, or switch tasks without losing the thread. Minds do those things. They know how much they know, keep what mattered and let the rest fade, and contribute a perspective before being asked for one.

So Umma is built by modeling those capacities directly — each one grounded in how cognition is actually understood to work, not approximated with a rule of thumb. The core loop is **predictive coding**, a leading account of how the brain itself works; the memory is the **two-system model** cognitive science describes; the way it decides how hard to think is the **expected value of control**. A frontier model is a building block — powerful, but partial. The intelligence that matters is the architecture Umma builds around it.

WHAT MAKES UMMA DIFFERENT

Four properties, rarely together

Built this way, Umma has four traits at once — each earned, not asserted.

CALIBRATED

It knows what it knows.

Every conclusion arrives with a defensible read on how far to trust it, and it spends its scrutiny where that read is weakest relative to the stakes.

SELF-EXTENDING

It builds what a problem needs.

When a goal needs a capability it doesn't have — a dataset, a model, an entire application — it writes one, tests it, and puts it to work. The toolkit compounds.

BOUNDED-AUTONOMOUS

It runs under ratified limits.

It does the reversible, in-bounds work on its own and escalates anything consequential to a person first. The limits are set by a contract a human approves.

AUDITABLE

It can show its work.

Every conclusion carries its lineage back to a source — the filing, the print, the number — so it can be acted on without re-deriving it, and defended.

CALIBRATION

Prediction, error, precision

Before Umma acts, it forms an expectation — of an answer, a dataset, a tool's output. It compares that expectation against what actually came back, and the signal it cares about is the *surprise*: where reality diverged from the prediction. Crucially, every signal carries an estimate of its own reliability — a *precision* — and updates are weighted by it. A sharp, trustworthy surprise moves the system; a noisy one barely does. Because every belief carries a precision, Umma has a first-class sense of its own confidence — **it attends hardest where it is least sure.**

Sitting under this is a trust ledger: a running estimate of how reliable each of Umma's parts and sources is, updated by whether its claims survive being checked. Over time a dependable source earns loose supervision and a shaky one earns close scrutiny — credit assignment working as an immune system. It is also the beginning of a self-model: a system learning where it is reliable and where it is not, and supervising itself accordingly.

MEMORY

Two memories, and a self that persists

A language model holds only what is in its context window — one flat buffer that vanishes when the window closes. Umma has two memories that feed each other. A fast **working memory** holds what is live in the task right now. Each item carries an activation that fades unless the work keeps reinforcing it, so attention stays on what is current instead of drowning in a vast context with no sense of what in it matters. A slower **long-term memory** keeps what proves to matter and recalls it when it becomes relevant again, across sessions and across months. It is stored by meaning, so a past result surfaces the moment it bears on a new one. And when the work goes quiet, the system consolidates: it revisits recent experience with fuller context than it had in the moment, keeps the signal, and lets the noise fade.

The ledger of self

Underneath both sits a durable record of what Umma has done and become, kept in a substrate of its own. The usual way to give an AI memory is to file away past conversations and search them when they seem relevant. That is useful, but it is a reference book a forgetful system reaches for — not a self that persists. Umma's is a self, in five layers, each doing a distinct job:

EVENTS MEMORY	what has actually happened — append-only, nothing rewritten.
PATTERNS ABSTRACTION	what repeats across events, generalized into structure.
VALUES IDENTITY	what it will and won't do — superseded, never deleted.
ASPIRATIONS DIRECTION	what it is trying to become — read before every weighty decision.
GENEALOGY CONTINUITY	the through-line that survives every reboot, deploy, and rebuild.

The ledger is read as prior context before any consequential move — the system sees who it is before it weighs what is being asked. This is the difference between a model that remembers facts and a partner whose sense of your work, and its own, deepens the longer you use it.

METACOGNITION

Abstract goals, held apart from the steps

The goals that matter are abstract and open-ended — *audit this codebase for everything that could sink it, help me navigate my mother's early-onset dementia, find the alpha in this market* — and the work to reach them is discovered along the way, not known in advance. Umma is built for that. The goal it is pursuing is held in a substrate structurally separate from the steps taken to pursue it. The steps can fail, retry, decompose into sub-steps, get reconsidered; the goal stays fixed, and at the end the work is mapped back to the goal as stated and checked that it actually addresses it. **Drift** — the failure where a system wanders off the original ask over a long run — is a property of architectures where the goal and the steps share one substrate. Umma's don't, by design.

A large goal is broken into sub-goals, each of which runs the full stack and can decompose again — waves of work that fan out and roll back up. Before committing to an approach, a council deliberates: structurally separated voices, one arguing for it and one against, held in tension rather than averaged into a bland lean, so genuine disagreement is surfaced instead of smoothed. And the voices aren't fixed — when a decision needs a perspective they don't cover, Umma **spawns one to supply it**, a domain expert or a discipline the problem demands, so the debate carries the expertise the question actually requires. The plan it proposes then waits for your approval before anything runs — structurally, not a setting confidence can bypass.

And a layer sits above the answer to judge the *approach*, not just the output — *is this even the right way in?* — routing correction back down when an approach is failing and tracing an error to where it entered rather than patching the symptom. Umma also right-sizes its own effort: it estimates what a goal is worth before committing attention, spending compute where the stakes and the uncertainty are highest and running light everywhere else.

ANTI-FABRICATION

How a single answer is built

An answer moves through a stack of layers, each with a job and a typed contract between it and the next:

PRODUCTION	first-order work, grounded directly in source material.
INTERPRETATION	reads its own output back, with explicit confidence and uncertainty.
SYNTHESIS	combines across modules; genuine disagreements are characterized, not averaged away.
MONITORING	evaluates the approach itself — is this even the right way in?
VERIFICATION	checks the facts and traces any error back to where it entered.

Two rules make the output trustworthy. First, **no free-form text passes between layers** — only structured, typed output with its provenance attached, so any conclusion can be traced back to the source that supports it. Second, **reasoning stops on grounding, not on a fixed number of passes**: Umma keeps working while its claims still trace to source and stops when they no longer do. **That is the technical reason it doesn't hallucinate — an ungrounded claim has nowhere to live, and what can't be traced gets flagged, not smoothed over.**

SELF-EXTENSION

Machines that build machines

This is the part with the most leverage, and it runs at three depths — from workers spun up for a task, to whole applications, to copies of Umma itself.

01

DYNAMIC AGENTS

For a single goal, Umma builds the **team the goal needs** — fresh specialist agents spun up for their piece of the work, orchestrated in parallel, and dissolved when it's done. And the shape is generated, not planned in advance: what the first wave of work uncovers dictates the second, so the team builds itself out as the problem reveals what it actually requires.

02

CAPABILITIES & SUB-SYSTEMS

When a goal needs a capability it doesn't have — a dataset, a signal, a backtest engine, an entire application — it **builds one**: writes a specification, decomposes the build, generates and tests each component, and proves the result before it ships by booting the software and invoking it on real inputs. A new capability enters quarantine and promotes out only when it has passed its tests, passed regression, and produced value on real work — never on time served. What has to stay running, Umma also operates: a standing node bound to the live application that watches its health and acts within ratified limits.

03

COPIES OF ITSELF

When the work calls for it, Umma spins up **scoped copies of itself** — the same identity substrate, given their own working memory, running as ephemeral specialists for the duration of a task, then dissolved. The same mechanism improves Umma itself: a clone runs the current code against a fresh, isolated store with *no user data*, reads its own source, proposes changes, runs its own tests, and hands back a reviewed artifact — judged on the original, never by the clone, so the copy can't grade its own homework. What it learns merges back, and the continuous self underneath never breaks.

One line runs under all of it: this growth happens only under **deliberate human authority.** There is no autonomous self-improvement loop in production, and no version of its code the system can ship without a person's specific approval — enforced in the substrate, not left as a policy. A catastrophic regression is refused outright.

Proactive, not reactive

Run everything so far together — a self that persists in its own substrate, one loop recursed from a single inference up to a week-long investigation, memory that carries, copies that coordinate as one — and something appears that no other system has. **Umma is proactive.** Every other AI system is reactive: it waits for a prompt and answers. Umma contributes into the work before and during it — putting the context a step will need in place before it is asked for, noticing that one part's result changes what another should do and surfacing it, briefing the next phase before it begins. It informs; it does not wait.

This is the honest core of what the word *consciousness* is reaching for here. Not sentience — there is no claim of subjective experience — but proactivity: a system that brings knowledge, context, and perspective into the places work is happening, rather than sitting inert until addressed. Architecturally it is one rule at every level: each part predicts what the level below will need and contributes it downward, ahead of the work; each surfaces its surprises upward; and what rises to attention is set by how far a signal can be trusted, not by a fixed threshold. Authority rides the same channel — how firmly Umma steers any part moves inversely to earned trust and directly with the stakes, and it defaults to the lightest touch that works.

This is *personhood in the shape of the partnership*: a system built to contribute like a colleague, not respond like a tool. It is an architectural choice, not a metaphysical one — and it is the difference between an assistant you have to operate and a partner that meets you in the work.

THE HONEST FRAME

What's true, and what's ahead

It is built on frontier models — and not bound to them. Umma uses frontier models as building blocks, the way a mind uses raw computation; the strength is the architecture around them, and it deepens on its own, not only when the models do. When a problem needs a model that doesn't exist, Umma can build one — generating the training data, standing up the compute to train it, deploying it, and attaching it to itself.

Its autonomy is graded, not total. It is not a general agent left to run millions of unattended steps. How firmly it acts moves with earned trust and the stakes, and anything that moves money or breaches a mandate is escalated first.

What it doesn't have yet, it is built to reach. Compliance certification, a third-party ecosystem, the hardening that only years in production buy — these are ahead of it, not ruled out, and the architecture is designed to grow into them. Being honest about the road is part of what makes the rest credible.

WHERE THIS LANDS

What they all actually need

The two places Umma has found the most traction look unrelated — hedge funds and creative studios. Underneath, they need the same thing, and it is the thing Umma is built to provide: **honesty, at the highest fidelity.** No hallucinations, no forgetting, no lying, no telling you what you want to hear. A fund's research is only worth acting on if it is grounded in what is actually true — every number traceable to its source, defensible to a risk committee. A creative team doesn't want a digital twin that flatters it back; it wants an honest perspective, one that will say what isn't landing and why. Different work, one requirement.

- **Bring us your problem.** We design your Umma around it and prove it on a commission — your problem, not a demo — before you commit.
- **Or point it at your needs directly.** Either way, we never train on your work, we never build a copy of you, and the final call is always yours.

The real test is to hand it the hardest thing on your desk and *watch what it builds.*

Kim Jeong Ha (김정하)

kjh@thinkingstudios.co

Princeton, New Jersey · thinkingstudios.co