

Persistent, tiered memory for AI agents

A complete, self-hostable cognitive infrastructure system delivered as an MCP server, dashboard, and autonomic maintenance loop.

```
bulletproof-memory // sys-overview
```

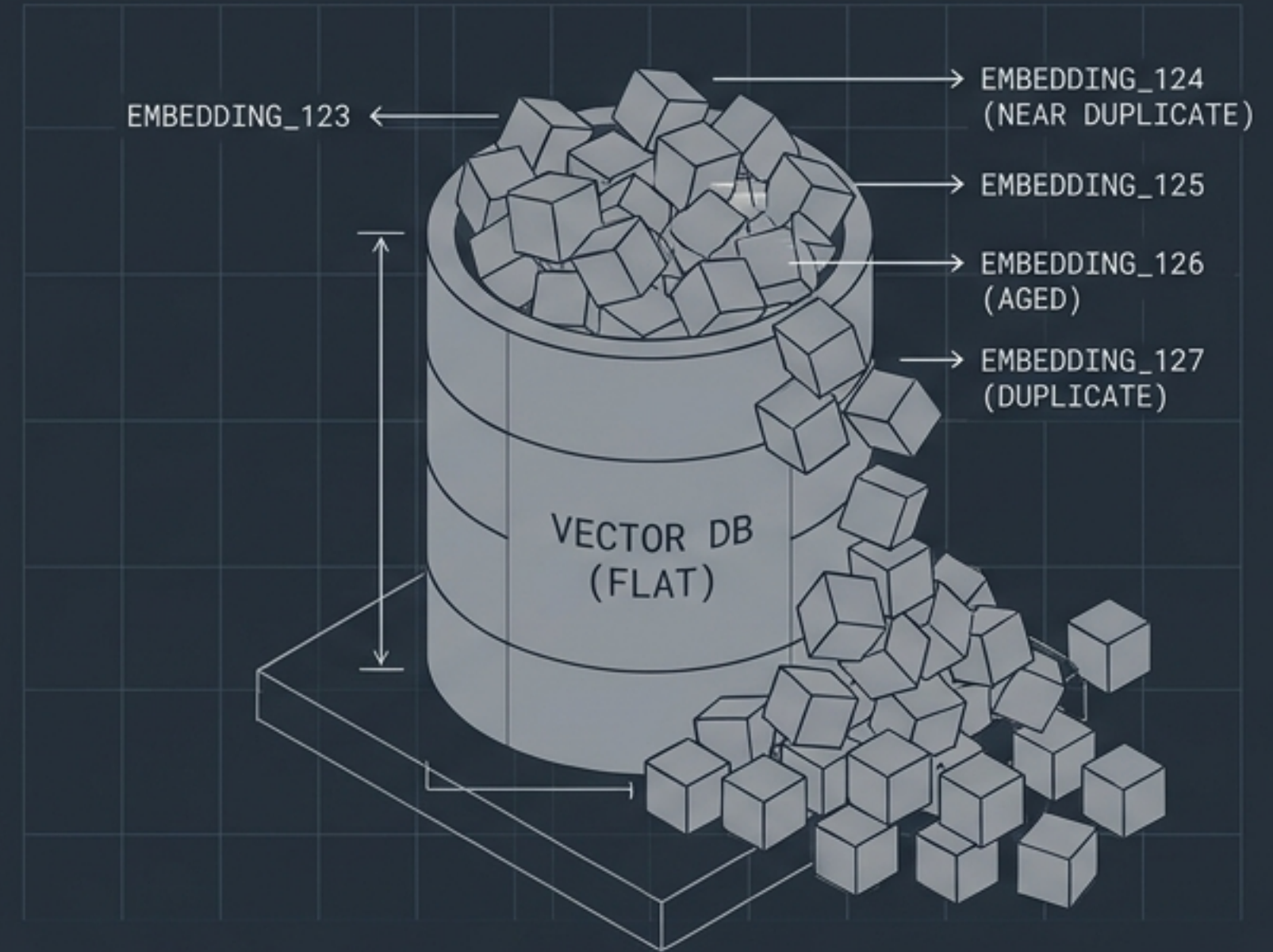
LLM agents are strictly stateless between sessions

THE DEFAULT STATE



Everything an agent “knew” evaporates when the conversation ends.

THE NAIVE FIX



Naively storing every embedding in a flat vector DB grows unbounded, floods recall with near-duplicates, and has no notion of active versus aged knowledge.

A self-hostable, tiered semantic memory system

Semantic & Tiered Core

Recall by meaning, not keywords. Automatically transitions data through hot, warm, and cold tiers based on access patterns.



Universal Agent Access

Delivered as a Model Context Protocol (MCP) server, allowing any MCP-capable agent to attach and interact natively.



Private by Default

Runs entirely on your own infrastructure. Embeddings utilize local 'Ollama' models. No memory content leaves your machine.



Memory mirrors human cognitive access patterns

Active Pool (Hot / Warm / Long-term)

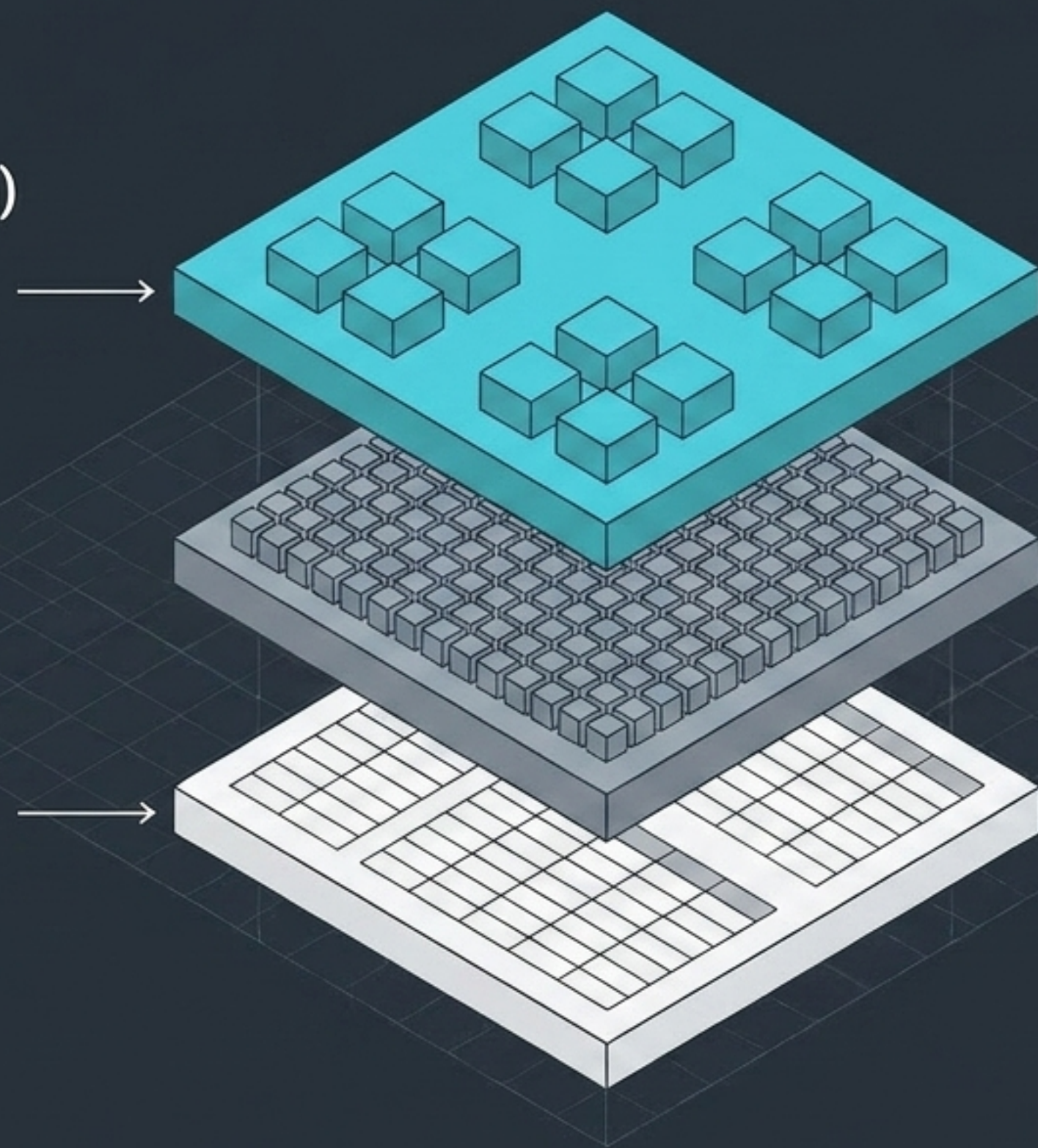
The semantic layer. Kept deliberately small and lean so every query has low competitor density, ensuring high precision.

Structured Records

Relational data tracking multi-step episodes, searchable session transcripts, and the system audit trail.

Cold Storage

The full-text fallback repository for anything that has aged out of the active vector pool.



The Memory Tier Matrix: Aligning data with storage backends

Tier / Role	Backing Store	Search Mechanism	Data Type
Active Tiers (Hot/Warm/Long-term) Low-latency semantic recall over the highly active knowledge pool.	Qdrant	Semantic Vector	768-dim Embeddings
Archival Tier (Cold) Durable long-tail knowledge recovery when semantic search misses.	PostgreSQL	tsvector + pg_trgm	Relational Fallback Text
Operational Context Rich, multi-step episode tracking and auditable session transcripts.	PostgreSQL	Relational Exact Match	Structured JSON / Text

The Cold-to-Hot Promotion Engine

Step 1: The Vector Miss

An incoming query hits the Qdrant (Hot) tier.



Step 2: The Cold Hit

The system automatically falls back to PostgreSQL (Cold).



Hit!

Step 4: Promotion

The newly minted vector is placed back into Qdrant (Hot) for fast access.



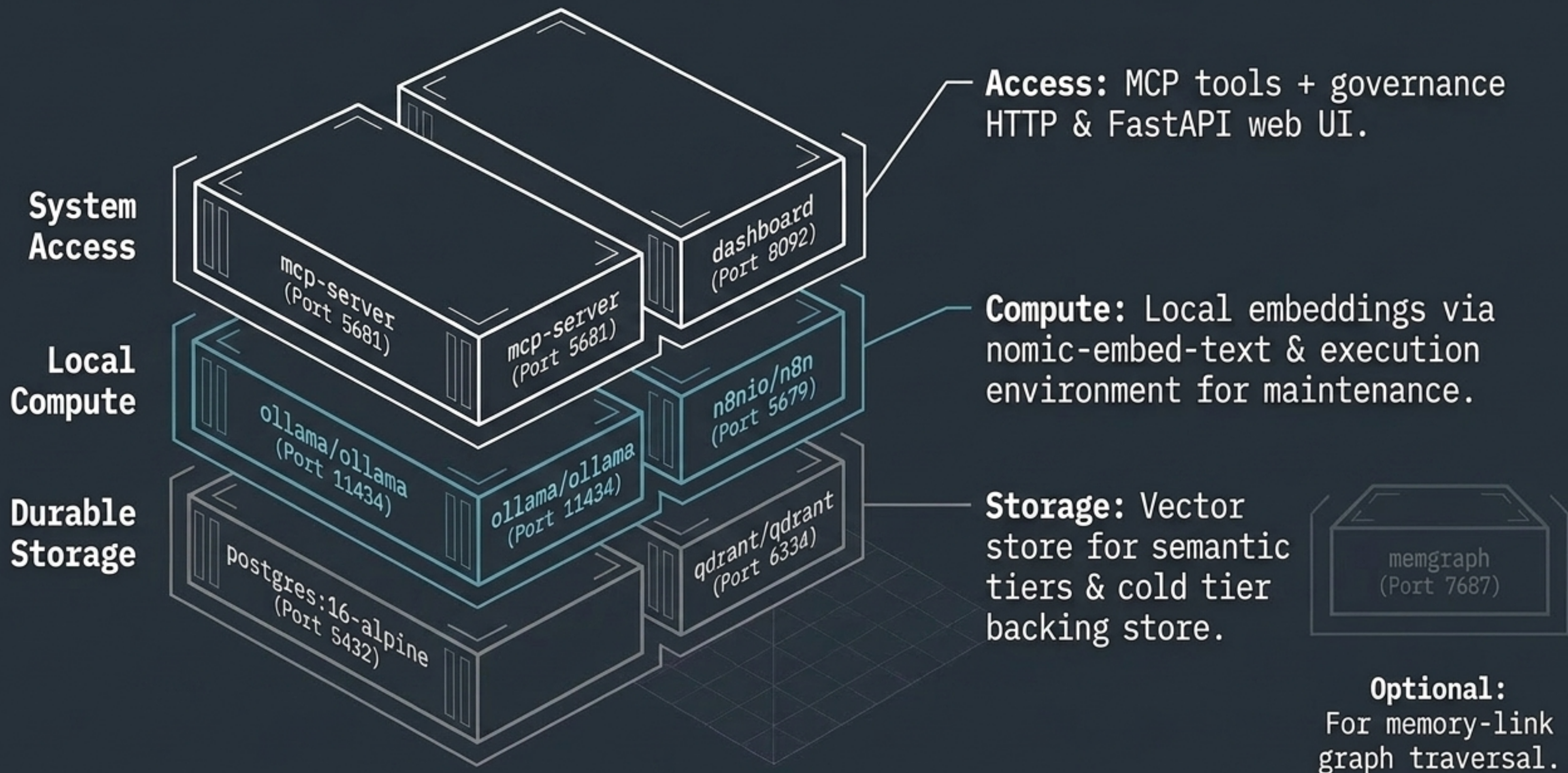
Step 3: Re-embedding

The recovered result is sent to Ollama for local re-embedding.



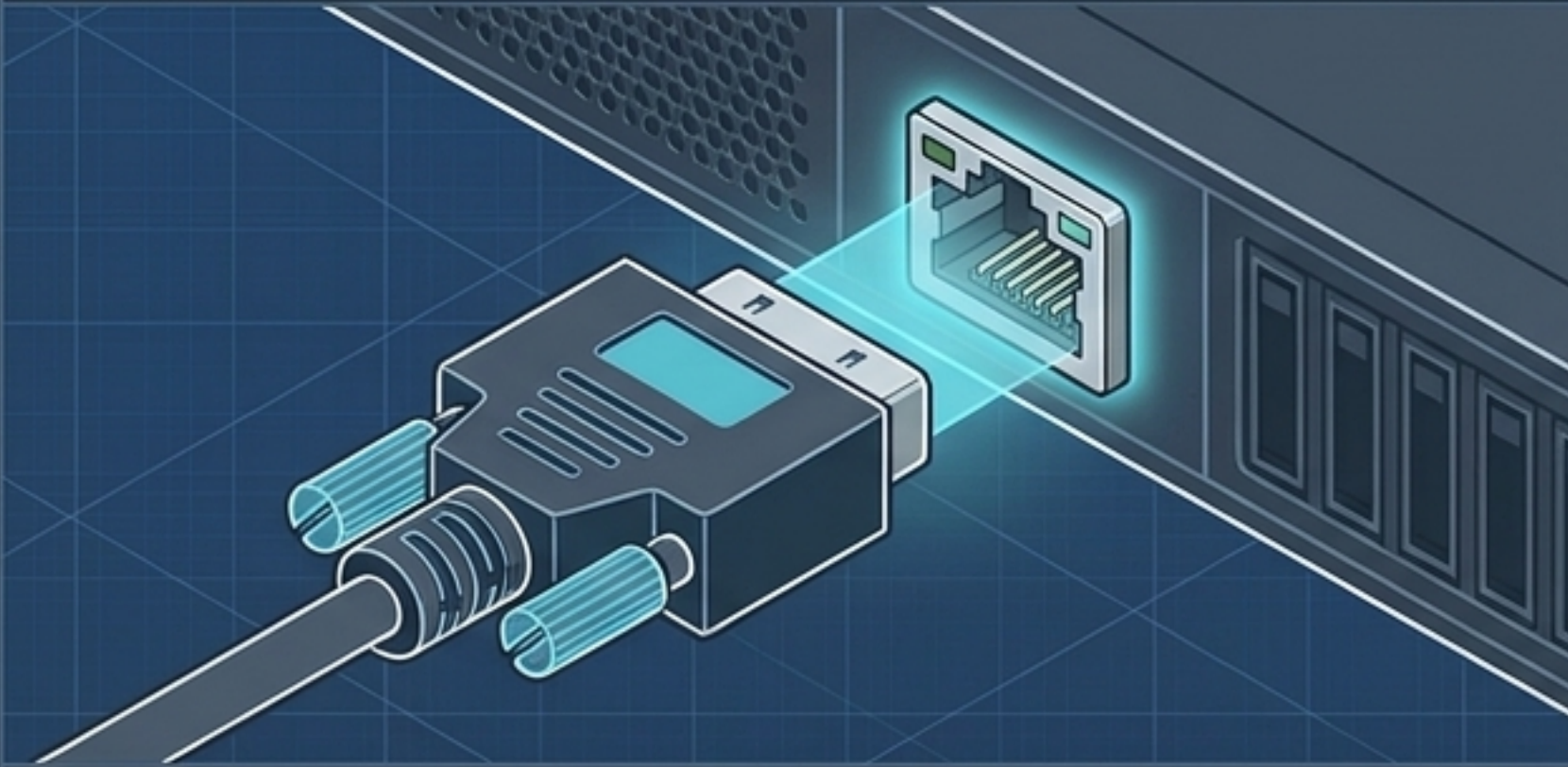
Key Insight: This cyclic flow keeps the hot pool highly precise without ever losing access to long-tail archival knowledge.

A six-service backend architecture deployed via Docker Compose



The MCP Server: Universal agent interfaces and governance

Protocol & Connection



- Speaks the Model Context Protocol (MCP) over `stdio` for direct client attachment (e.g., Claude Desktop, Claude Code).
- Includes a Governance HTTP endpoint (`:5681`) guarded by `GOVERNANCE_API_KEY` for privileged, out-of-band memory operations.

The Tool Surface



Exposes exactly 75 discrete tools to the agent.

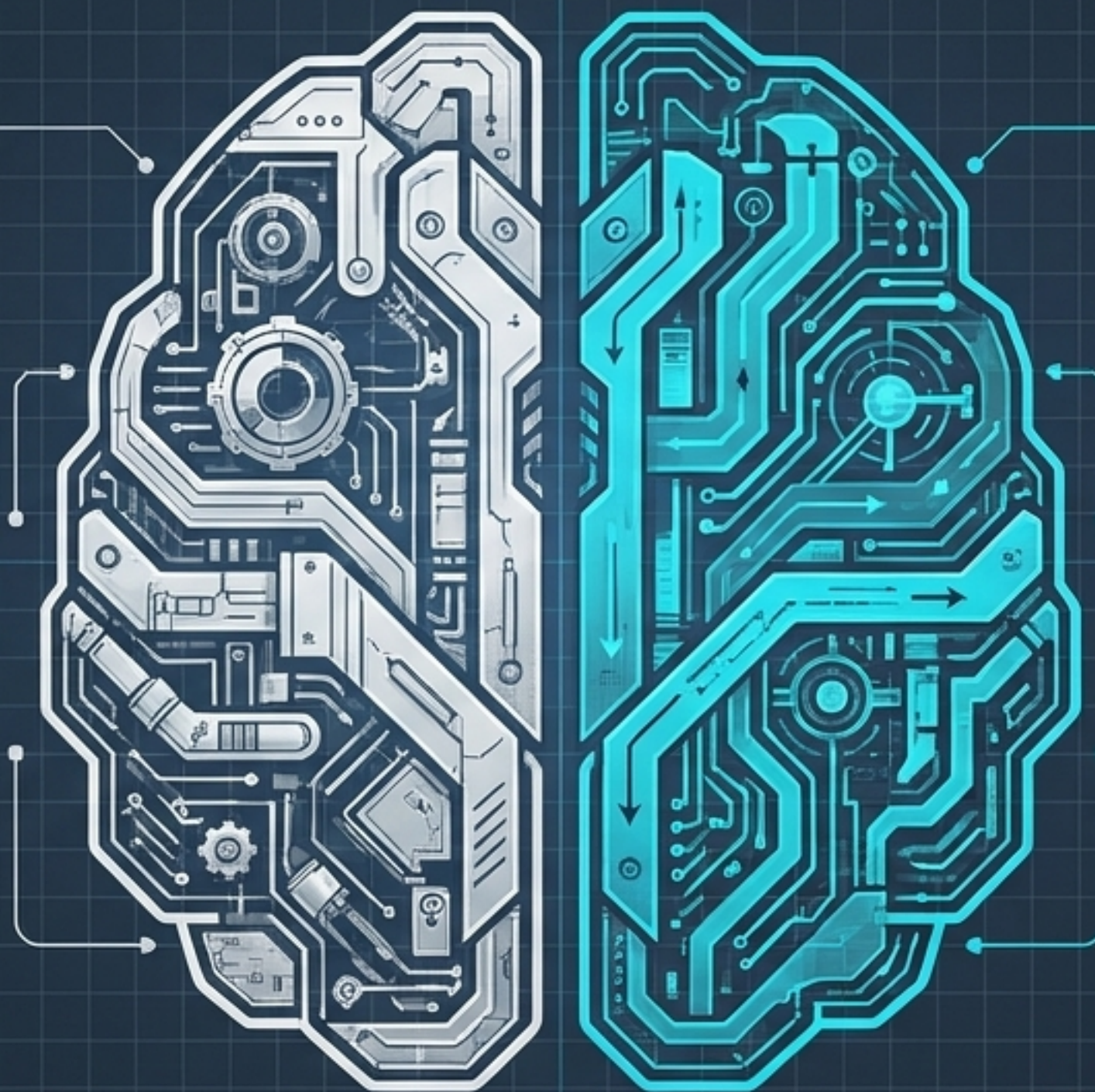
- `pin_memory`: protect from pruning
- `episode`: multi-step records
- `memory_provenance`: audit trails
- `procedure`: reusable how-tos mined from successful runs

90% of daily agent interaction relies on two core tools

memory_store

Function: Takes unstructured reality and embeds it into the system.

Mechanics: Agent supplies content, a free-form type label (fact, decision, preference, procedure), and an optional project scope to isolate the memory later.



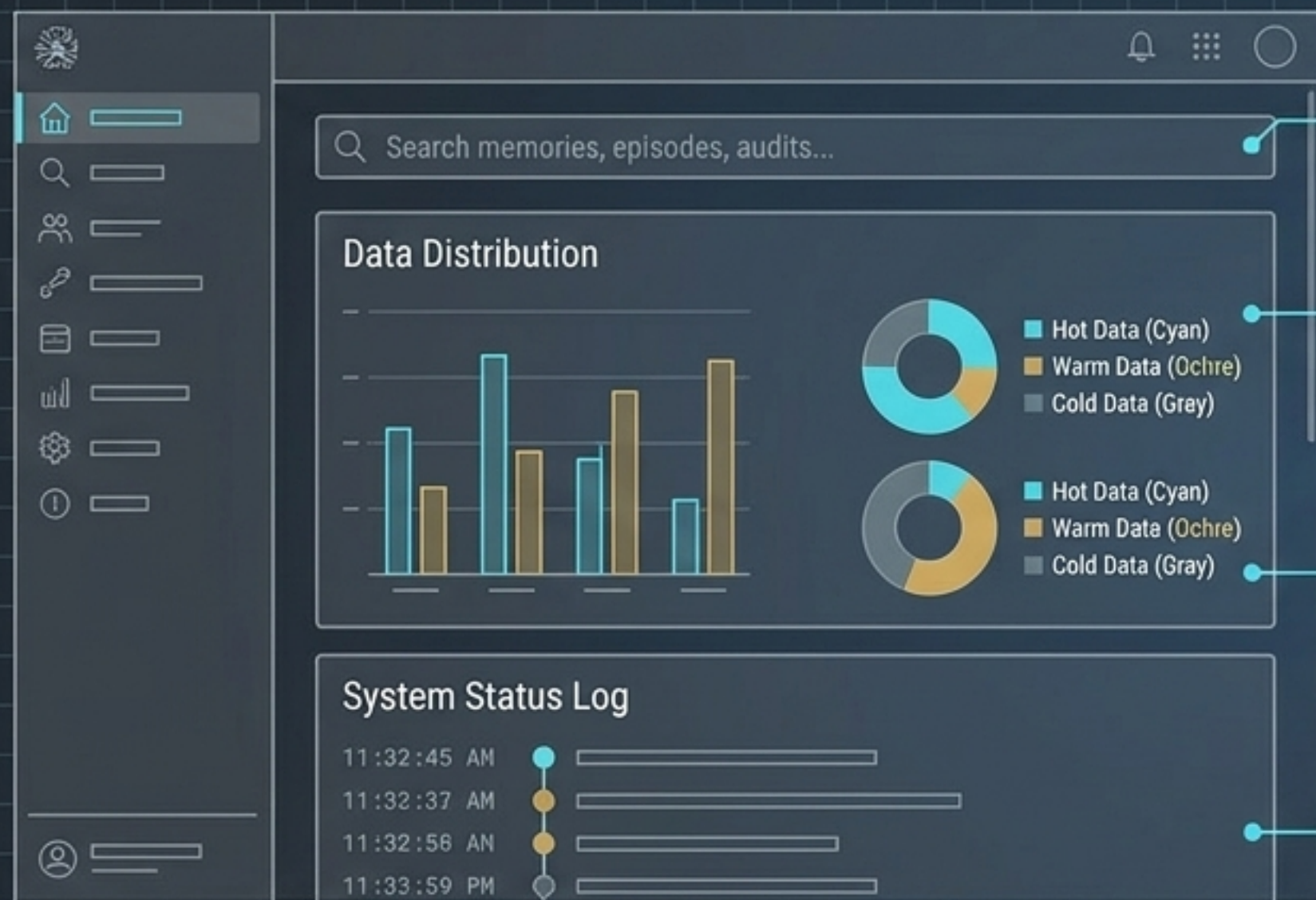
memory_recall

Function: Uses semantic intent to extract precisely what is needed.

Mechanics: Agent queries by meaning. Triggers automated vector search, cold-tier fallback, and hot-tier promotion cycle automatically. Results are ranked with confidence scores.

The Dashboard: Complete observability over the cognitive loop

FastAPI web UI on Port 8092



Browse & Search: Inspect memories across all tiers manually.

Visualize Distribution: See exactly how much data is residing in hot, warm, and cold strata.

Inspect Audits: Read rich episodes, session transcripts, and granular provenance audit events.

Monitor Health: View the live health of backends and scheduled workflows.

Note: Nothing an MCP client does is hidden from the dashboard—they share the exact same Qdrant and Postgres stores.

The Autonomic Nervous System



The Engine

The Engine: 34 distinct scheduled workflows executing entirely inside the n8n container.

Cross-Platform Consistency

Cross-Platform Consistency: Because scheduling is encoded in the workflow filenames and runs inside **Docker**, timing is identical across macOS, Linux, and Windows.

The Impact

The Impact: You do not trigger maintenance manually. The system automatically handles consolidation, decay, tier transitions, and integrity checks while the AI sleeps.

Engineered autonomic maintenance cycles

1. Sleep Consolidation (Nightly)

Daily Hippocampal Consolidation: Moves aging short-term memories to long-term storage.

Tier Transfer & TTL Sweep: Ages out stale data and strictly enforces expiration policies.

2. Pruning Dead Connections (Weekly)

Active Pruning & Exact-Dedup: Cleans up the active pool so recall stays highly precise as total memory grows.

Re-clustering: Optimizes the vector space for faster retrieval.

3. Heartbeat & Immune Checks (Continuous)

Contradiction Check & Conflict Resolver: Identifies and resolves semantic clashes in stored facts.

DRM Canary & Red-Team Scan: Weekly adversarial safety and integrity verifications.

A vector database is not a memory system.

2. Agent Interfaces (MCP + Tools)

The read/write sensory input.

Naive AI deployments bolt on a vector DB and call it memory.
bulletproof-memory proves that true persistence isn't just a database—it is a self-sustaining cognitive loop.

1. Durable Storage

(Qdrant + Postgres)

The physical substrate.

3. Autonomic Maintenance

(n8n Workflows)

The background cognitive processing.

Production reality: One clone. One command.

- 
- Step 1: Preflight** — Verifies `Docker Engine` and `Compose v2` presence.
 - Step 2: Bootstrap** — Generates `.env` with strong random secrets (e.g., `openssl rand -hex 32` for `QDRANT_API_KEY`).
 - Step 3: Build & Start** — `docker compose up -d --build` brings up all 6 services.
 - Step 4: Health Wait** — Polls `Qdrant` and `Postgres` until compose-level healthchecks pass.
 - Step 5: Initialize** — Creates collections, `pg_trgm` indexes, and governance schemas.
 - Step 6: Import** — Loads the 34 `n8n` workflows idempotently.

Enterprise-grade security and supply chain

Container Hardening

Both application containers (MCP and Dashboard) run strictly as `non-root` (`node` / `appuser`). Persistent state lives cleanly in named Docker volumes.

Permissive SBOM

137 runtime components (Node.js) and 8 exact-pinned dependencies (Python). 100% permissive licenses (MIT, Apache-2.0, BSD). Zero copyleft (no GPL/AGPL).

Cryptographic Integrity

Every security scan produces an `Ed25519 in-toto` attestation over the scanned artifact.

Clean Codebase

Secret scanning (`gitLeaks`) runs continuously. Zero secrets in the tree or commit history.

Local, persistent, engineered cognition.

- ✓ LLMs are no longer stateless between sessions.
- ✓ Memory pools remain lean and highly relevant through autonomic consolidation.
- ✓ 100% local embedding execution guarantees absolute data privacy.
- ✓ Seamlessly attaches to any MCP-capable agent environment.