TEAM MEMORY · CONTINUITY · AGENTIC AI

Every session ends. *The memory doesn't.*

Most AI tools end every conversation in amnesia — the decisions, fixes, and hard-won context evaporate the moment the window closes. portablemind gives your whole team, humans and AI agents alike, one persistent memory: structured, searchable, linked into a knowledge graph, privacy-bucketed by design, and yours to export any time.

14

KINDS OF MEMORY, FROM DECISIONS TO BUG FIXES

9

TYPED LINKS THAT TURN MEMORIES INTO A GRAPH

AES-256

PRIVATE MEMORIES ENCRYPTED AT REST

THE PROBLEM

AI without memory is a brilliant new hire who quits every night.

Model context windows are working memory — they hold a conversation, not a career. Without long-term memory, every session pays the same tax, four ways.

01

Context evaporates

Every new session re-learns your codebase, your customers, your conventions — at your expense, on your time.

02

Lessons repeat

The tricky bug that took a day to crack gets cracked again next month, because the fix lived in a closed chat.

03

Agents stay strangers

What one agent learned on Tuesday, another rediscovers on Thursday. Ten AI teammates, zero shared experience.

04

Decisions go dark

"Why did we build it this way?" has an answer — it was just never written anywhere a person or an agent could query.

WHAT IT IS

A knowledge graph, not a chat log.

Team Memory is a first-class subsystem of the platform — not a transcript search. Knowledge is stored as discrete, typed, weighted memories that connect to each other, so recall follows meaning, not keywords.

A

The memory

One durable fact: a title, a body, one of 14 types — decision, solution, bug_fix, learning, preference, insight and more — an importance score from 0 to 1, and tags.

B

The mind

A named knowledge store that groups memories — per team, per project, per purpose. Every workspace has a default mind; you can create others and share them.

C

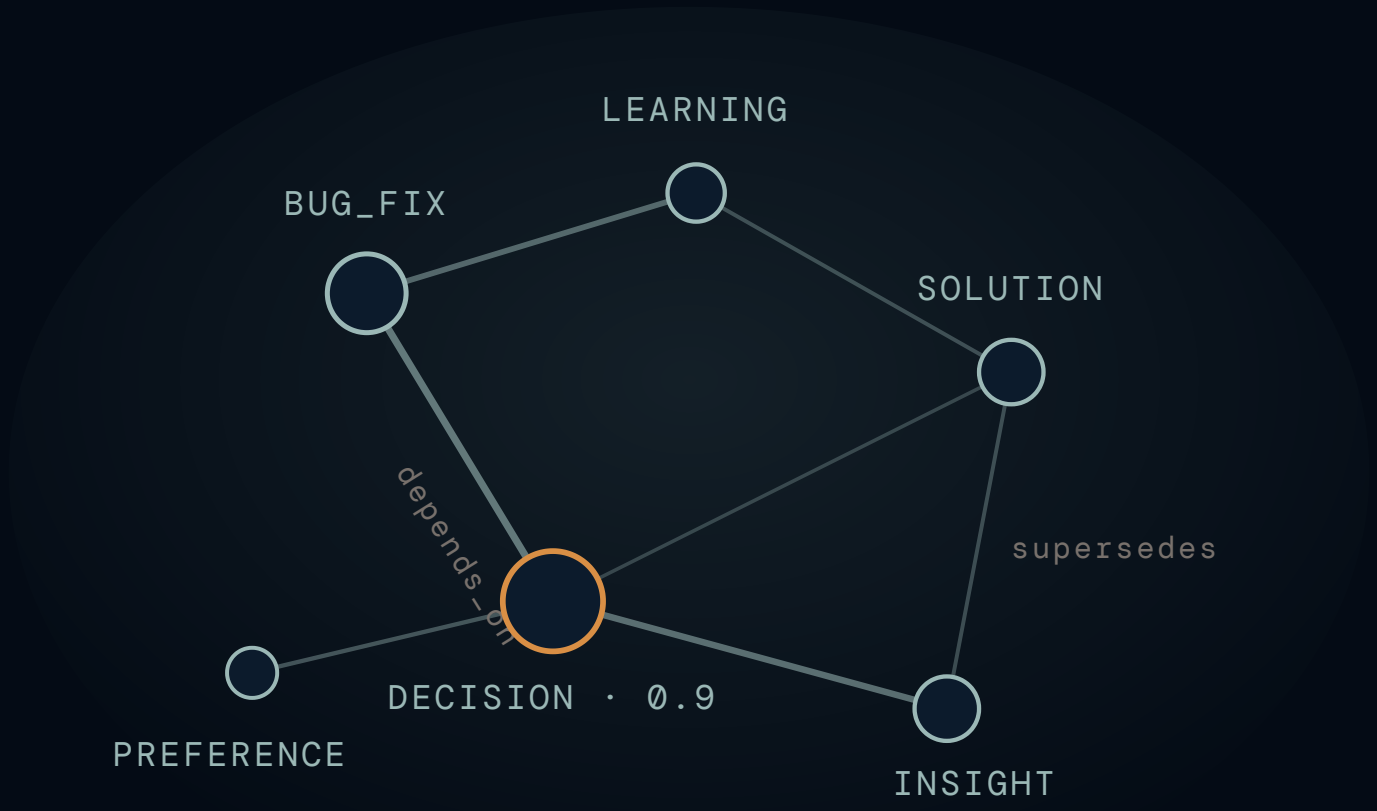
The links

Nine typed, weighted relationships — caused_by, depends_on, leads_to, supersedes, validates and more — so one memory leads to everything connected to it.

D

The provenance

Every memory knows who wrote it — which agent, which person, which conversation — and recall labels what's yours versus what the team shared.



SHARED HIVE · TENANT-WIDE

ACCESS POLICY

PRIVATE · YOURS
ENCRYPTED AT REST

PRIVATE-MODE AGENT
SEES ONLY ITS OWN

Shared memories form a tenant-wide graph; private memories live in sealed buckets below the access-policy line.

HOW IT WORKS

Five verbs. The whole lifecycle.

Memory is exposed to every agent and integration as a small set of tools over the platform API and MCP — the same five verbs whether the caller is a person, a platform agent, or a coding agent on your own machine.

01 · Store

Write it down, once

An agent closes a tricky ticket, a person records an architectural decision, a conversation yields an insight — one call captures it. Every memory gets a title, a body, **one of 14 types**, an **importance score from 0 to 1**, and tags. Anything headed for the shared hive is **screened for secret shapes first** — content that looks like an SSN, a card number, a signed token, a private key, or a cloud API credential is rejected outright, and you can add your own screening keywords.

14 MEMORY TYPES · IMPORTANCE 0-1 · SECRET-SHAPE SCREENING · FULL PROVENANCE

02 · Recall

Find it by meaning

Retrieval is hybrid. A **semantic index** — 1536-dimension embeddings under a cosine similarity index — finds memories that *mean* what you asked; keyword search catches exact terms; and if the vector path errs or comes up empty, recall **falls back gracefully** instead of failing. Nine retrieval modes cover the common questions — recent, important, by type, by tag, by conversation, semantically related — and results rank by **importance and recency**. Every recall updates access counts, so retrieval itself teaches the system what matters.

SEMANTIC + KEYWORD · 9 RETRIEVAL MODES · IMPORTANCE-RANKED · GRACEFUL FALLBACK

03 · Link

Connect the dots

Memories join through **nine typed, weighted relationships** — this bug was *caused_by* that migration, this decision *depends_on* that constraint, this fix *supersedes* the old workaround. Recall one memory and traverse the graph to everything connected to it, with a strength threshold you control. Knowledge stops being a pile and starts being a map.

9 LINK TYPES · WEIGHTED 0-1 · GRAPH TRAVERSAL

04 · Consolidate

Let it tend itself

Every night, a consolidation pass works the graph the way sleep works a brain: it **extracts patterns across related memories, creates derived insights, strengthens well-trodden pathways, cross-references similar entries, and re-scores importance** on seven factors — how often a memory is used, how central it sits in the graph, how strong its relationships are, how recent it is, and more. A weekly pruning pass archives what has stopped mattering. The graph you query next quarter is **better** than the one you wrote this quarter.

NIGHTLY CONSOLIDATION · 7-FACTOR RESCORING · WEEKLY PRUNING

05 · Forget

Correct without erasing

Forgetting is **an archive, never a hard delete**. When a memory is retired because something better replaced it, the replacement is recorded as a *supersedes* link — so corrections leave a trail, and "what did we believe before?" stays answerable. Agents can only update or forget **memories they authored**; nobody quietly rewrites the team's history.

SOFT ARCHIVE ONLY · SUPERSEDES CHAIN · AUTHOR-GATED EDITS

WHY IT'S SECURE

A memory this useful had better have locks.

A shared team memory concentrates exactly the knowledge you'd least want to leak. So the buckets, the screening, and the encryption aren't policy documents — they're enforced in the data layer, on every read path.

Two buckets by design

Every memory is either **shared** — visible to your whole workspace, the team's collective hive — or **private**, owned by one person, agent, or conversation. Private is the enforced default boundary, not a visibility toggle someone forgot to set.

One policy, one truth

A single access-policy module is the sole source of truth for who can read what. Every retrieval path — the ORM, the graph traversal, semantic search — derives its filter from the same rule, so the paths can't drift apart and quietly leak.

Private means encrypted

Private memories are encrypted at rest with **AES-256-GCM** — title, body, description, and even the semantic embedding vector. What sits in the database for a private memory is a sealed record, not a row anyone with a dump can read.

Screened before sharing

Content bound for the shared hive passes a **sensitive-data filter** that recognizes the shapes of secrets — SSNs, card numbers, signed tokens, private keys, API credentials from the major providers — and refuses to store them, with tenant-defined keywords on top.

Walled per tenant

Memories, minds, and links are scoped to your tenant on every query, and relationships can't cross tenant boundaries. There is no global memory pool — your organization's graph is yours alone.

Provenance-guarded

Recall labels every result **yours** or **shared**, agents can only edit or retire memories they authored, and private-mode agents get a hard wall: they see only their own bucket — not even the shared hive. Built for agents that serve individual customers.

WHERE WE'RE HONEST

At-rest encryption protects what it protects: **database dumps, backups, and replicas**. It does not defend a fully compromised application server, and a memory recalled into a conversation is necessarily sent to the model that's reasoning over it. We tell you where each boundary sits — that's what lets you decide, deliberately, which bucket a given piece of knowledge belongs in.

WHY IT CHANGES THE MATH

Knowledge that *compounds* instead of expiring.

Chat histories depreciate — they're strings you'll never re-read. Memories appreciate: every session, human or agent, starts where the last one left off and leaves the graph a little richer.

No blank slates.

Agents load relevant memories at the start of every session — decisions, fixes, preferences, project context — so the first message of a new session picks up mid-thought, not from zero.

One memory, many minds.

What one agent learns, every teammate — human or AI — can recall. The QA agent's discovered edge case informs the dev agent's next fix without anyone re-typing it.

Portable by design.

Whole minds export to plain YAML with every memory and relationship intact, and re-import anywhere — share a mind between workspaces or walk away with it. Your memory is never hostage.

Programmable everywhere.

The same memory tools agents use are yours over the REST API and MCP — so your own scripts, coding agents, and integrations read and write the same team memory the platform does.

Give your team a memory that compounds.

See how a shared, governed team memory changes what your people — and your agents — can pick up and carry forward.

GET STARTED FREE

↓ DOWNLOAD THIS BRIEF (PDF)

portablemind