

MCP SERVER

NO CODE

CLOUD HOSTED

Memo MCP for AI Agents

Give your agent long-term memory to remember user preferences and past context.

Mem0 gives your AI agent persistent memory. It stores, searches, and recalls facts, preferences, and context across different sessions. Instead of your agent forgetting everything the moment a chat ends, Mem0 lets it build a lasting profile of every user it interacts with.

A+ Quality Score 100/100

long-term-memory

personalization

context-retention

semantic-search

user-profiles

ai-memory



The connectivity layer between AI and the world's software.

Vinkius sits between AI and every application. All communication passes through Vinkius Cloud via the Model Context Protocol (MCP) — with governance, observability, and security at every layer.

Your AI Connections Run Through Vinkius Cloud

The world's largest
managed MCP catalog

Vinkius is the connectivity layer where AI connects to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.

— Architecture principle

Four Pillars of the Vinkius Runtime

01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

AES-256

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

Sandboxed execution

One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT`https://edge.vinkius.com/{token}/mcp`

Claude



Cursor



VS Code



Windsurf



Grok



Gemini

Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot is used outside Vinkius infrastructure, the server is quarantined instantly.

Emergency Kill Switch

EU AI Act Art. 14(1)
Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

Control Plane

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

Incident Log

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

Get started at cloud.vinkius.com — connect your AI agent in under 60 seconds.

Memo MCP

4 tools available

Cloud-hosted on Vinkius

Connect your AI agent to Mem0 to give it a long-term memory layer. Most agents start every conversation with a blank slate, forcing users to repeat their preferences, project details, and past decisions over and over again. This MCP changes that by allowing your agent to build and maintain a persistent knowledge base of user-specific information. When a user shares a detail, the system automatically pulls out the relevant facts and stores them so they can be recalled later. If you need to know what a specific user likes or how they prefer to work, your agent can query these memories to provide a more tailored experience. It handles the heavy lifting of organizing and searching through these data points, so your agent doesn't have to manage a messy database manually. You can find this in the Vinkius catalog to get your agent up and running with a memory that feels natural. It turns a standard chatbot into a personalized assistant that actually learns from every interaction. Instead of a tool that just answers questions, you get a system that understands context. It remembers that a user prefers certain coding styles, has specific deadlines, or even just has a particular sense of humor. This level of persistence is what makes an AI feel like a true partner rather than a temporary utility.

Core Capabilities

01 — Store new facts

Mem0 lets you save specific user preferences or context into a permanent profile.

03 — View user history

Mem0 lets you see every piece of information the agent has gathered for a specific user.

02 — Recall relevant info

Mem0 lets you find specific past details using natural language queries.

04 — Remove outdated facts

Mem0 lets you clear out incorrect or old information to keep the agent's knowledge base clean.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/mcp/mem0 — connect your AI agent in three steps.

- 01 Subscribe to the Mem0 MCP and get your API key.
- 02 Plug the key into your AI client configuration.
- 03 Start chatting and watch your agent save and recall facts automatically.

The bottom line is your agent stops forgetting who it's talking to and starts remembering every detail that matters.

Built For

This is for developers and product teams who are tired of building forgetful bots.
It's for the person building a personalized SaaS experience where the AI needs to remember a user's style across weeks of interaction.

AI Agent Developer

Builds complex agents that need to remember past decisions without constant manual prompts.

Chatbot Builder

Creates high-touch customer service bots that remember a customer's previous issues and preferences.

SaaS Product Manager

Integrates persistent memory into a product so the AI feels like a long-term partner rather than a one-off tool.

What Changes When You Connect

- 01 Stop asking the same questions. Use `add_memory` to save details once so the agent never has to ask for them again. This makes every new session feel like a continuation of the last one.
- 02 Provide instant personalization. Your agent can use `search_memories` to find specific facts, making every interaction feel unique and tailored to the user's specific needs.

-
- 03** Keep knowledge bases clean. Use `delete_memory` to prune old data and ensure your agent doesn't get confused by outdated info or incorrect facts from previous sessions.
-
- 04** Build complete user profiles. Use `get_memories` to see exactly what the agent knows about a person at any given time, which is great for auditing or building deeper context.
-
- 05** Improve retention across sessions. Your agent can recall context from weeks ago, making the experience feel like a continuous conversation rather than a series of disconnected chats.
-

Real-World Applications

User prefers dark mode

A user tells an agent they like dark mode. The agent uses `add_memory` to save this, then applies it to every future UI suggestion without being asked again.

Coding preferences

A developer mentions they use TypeScript. The agent uses `search_memories` later to provide the correct syntax for their specific stack automatically.

Customer Support

A customer complains about a recurring bug. The agent saves this via `add_memory` so the next agent who helps knows the history immediately.

Personalized Coaching

A fitness client shares their goals. The agent recalls these via `search_memories` to tailor every workout plan it generates for them.

Patterns to Avoid

Saving every single word

X AVOID

Trying to save every single sentence from a chat as a distinct memory.

✓ INSTEAD

Use `add_memory` to only store key facts, preferences, and significant context points rather than raw transcripts.

Forgetting to prune

X AVOID

Letting the memory get cluttered with old data that is no longer true.

✓ INSTEAD

Periodically use `delete_memory` to remove outdated facts so the agent doesn't get confused by old information.

Searching without context

X AVOID

Asking the agent to find memories without specifying which user you are talking to.

✓ INSTEAD

Always provide a specific user ID when using `search_memories` to keep data private and contextually accurate.

The Right Fit

Use this if you need your AI agent to have a 'brain' that persists across different chat sessions. It's perfect for building personalized assistants, long-term coaching bots, or SaaS tools where the AI needs to 'know' the user. Don't use this if you just need a simple RAG system for a static document; a standard vector database is better for that. Also, don't use this if you don't have a specific user ID system, as Mem0 relies on associating memories with specific users to keep context organized. If you need a way to handle massive amounts of unstructured data for a general knowledge base, a dedicated search tool is a better fit than a memory layer.

Memo MCP for Persistent Agent Memory and Personalization

Most AI interactions today feel like a game of telephone. You tell an agent your preferences, your project goals, and your specific constraints, only to have it forget everything the second you start a new thread. You end up copy-pasting the same instructions every single morning just to get the agent on the right track.

Mem0 changes that by giving your agent a permanent place to store what it learns. When a user shares a preference, the agent saves it to a long-term profile. The next time you open the app, the agent already knows who you are and how you like to work. You get a tool that actually builds a relationship with the user over time.

Memo MCP for Building User Profiles and Context Retention

Manually managing user context usually involves messy databases or trying to stuff everything into a prompt window that's quickly running out of space. You're constantly balancing what the agent needs to know right now versus what it learned three weeks ago.

This MCP handles that balance for you. It extracts the important bits and stores them efficiently. Instead of managing rows of data, you just let the agent interact naturally. The result is a much smoother experience where the AI feels like a teammate who's been on the team for months, not a stranger you have to re-brief every day.

Memo MCP: 4 Tools for Persistent Agent Memory

Give your agent the ability to store, search, and recall user facts and preferences automatically.

#	TOOL	DESCRIPTION
01	delete_memory	Mem0 uses delete_memory to remove a specific memory by its ID when it's no longer accurate or needed. This helps keep your agent's knowledge base clean and relevant.
02	get_memories	Mem0 uses get_memories to list every stored memory for a specific user to build a full profile of their history. Use this to see exactly what your agent knows about a person.
03	search_memories	Mem0 uses search_memories to find the most relevant memories based on a natural language query from the user. This allows your agent to recall past facts and preferences instantly.
04	add_memory	Mem0 uses add_memory to save new facts and preferences from a conversation into the user's persistent memory profile. The system automatically extracts key details for you.

See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

U Remember that I prefer dark mode, use VS Code, and my favorite language is TypeScript.



Stored 3 new memories for your profile:

- **User prefers dark mode for UI**
- **User's primary IDE is VS Code**
- **User's favorite programming language is TypeScript**

These will be automatically recalled in future conversations.

U What do you remember about my coding preferences?



Found 3 relevant memories (ranked by relevance):

- **User's favorite programming language is TypeScript** (relevance: 0.95)
- **User's primary IDE is VS Code** (relevance: 0.89)
- **User prefers dark mode for UI** (relevance: 0.72)

U Show me all the memories you have stored for my user profile.



Found 5 memories for user_123:

- 'User prefers dark mode'
- 'User's primary IDE is VS Code'
- 'Favorite language is TypeScript'
- 'Works best in the morning'
- 'Prefers functional programming patterns'

Created between 2026-01-15 and 2026-04-01.

Frequently Asked Questions

01 What is Mem0 MCP for AI agents?

Mem0 MCP for AI agents provides a persistent memory layer. It lets your agent remember facts, preferences, and context from previous conversations so it can provide a personalized experience every time you talk to it.

02 How does Mem0 help with personalization?

It stores specific user details automatically. When you tell your agent something about your workflow or preferences, it saves that info and recalls it during future sessions to make the AI feel more tailored to you.

03 Can my AI agent remember things across different sessions?

Yes, that's the main goal. While most agents start fresh every time, this MCP allows your agent to pull from a permanent memory bank to keep track of your history and preferences indefinitely.

04 How do I keep the agent's memory from getting cluttered?

You can manage the stored information easily. The system allows for the removal of specific memories, so you can delete outdated facts or incorrect information to keep the agent's context sharp.

05 Is Mem0 MCP for AI agents good for building user profiles?

It's perfect for that. It allows you to view all stored memories for a specific user, making it easy to see exactly what the agent has learned about them and how it's building their unique profile.

06 Does Mem0 MCP for AI agents require a complex setup?

No, it's designed to be a straightforward memory layer. Once you connect your API key to your AI client, the agent starts handling the memory extraction and storage automatically.

07 Is Mem0 free to use?

Yes! Mem0 offers a free Hobby tier with 10,000 memories and 1,000 search calls per month — no credit card required. Paid plans start at \$19/month for higher limits. An open-source version (Apache 2.0) is also available for self-hosting.

08 How does Mem0 extract and store memories?

When you send content to Mem0, its AI automatically extracts key facts and structured information. For example, if you send 'I prefer Python over JavaScript and work best in the morning', Mem0 creates two separate memories: one about language preference and one about work schedule. These are stored in a hybrid architecture (key-value + vector + graph) for fast semantic retrieval.

09 Can I organize memories by user or agent?

Yes! Every memory operation supports scoping by `user_id`, `agent_id`, or `run_id`. This means you can maintain separate memory banks for different users, different agents, or even different conversation runs — keeping context perfectly isolated.

Go Live in 60 Seconds

Get your connection token from cloud.vinkius.com, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

`https://edge.vinkius.com/[TOKEN]/mcp`

CLIENT	WHERE TO CONFIGURE
 Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
 Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
 VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add <code>"mem0": { "url": "..." }</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → Add endpoint URL
 ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
 Gemini	Extensions → Add MCP Server → Paste endpoint URL

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

-  Ask ChatGPT 
-  Ask Claude 
-  Ask Perplexity 
-  Ask Gemini 
-  Ask Grok 

READY TO CONNECT

Memo is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

Start at cloud.vinkius.com →

vinkius.com · support@vinkius.com

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by Mem0. All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

DOCUMENT INFORMATION

Generated	July 2026
MCP Server	Mem0 MCP
Server ID	019d75d2-180a-72ef-9f22-8542429f72a5
Platform	Vinkius Cloud for AI Agents
Endpoint	<code>https://edge.vinkius.com/{token}/mcp</code>

LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/mem0.