MCP SERVER | NO CODE | CLOUD HOSTED

Reflect MCP

Map your knowledge graph with natural conversation.

Reflect lets your AI agent read, write, and explore your private knowledge graph directly through conversation. It turns your notes app into an active research assistant that maps connections between ideas, bookmarks, books, and daily thoughts.

A+ Quality Score 100/100

networked-thought | backlinks | knowledge-graph | personal-wiki | thought-capture | daily-journaling

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/reflect

SCAN TO VIEW

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.

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.

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.

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	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	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.

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.

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.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken 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.

Agent Activity

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

FinOps

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

Tool Health

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

Firewall & DLP

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

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.

Reflect MCP

10 tools available
Cloud-hosted on Vinkius

You connect your Reflect account to your AI client to give it access to your personal, networked thought graph. This MCP lets you treat your entire collection of notes not as separate files, but as one interconnected web of knowledge.

Instead of manually searching through dozens of folders or apps, you simply ask your agent questions about your ideas. It can look up specific notes, find every piece of content that links back to a core concept, and map out the relationships between disparate thoughts. You can also tell it to capture action items from a call and log them directly into today's daily note, or save a URL you just read so it gets indexed with metadata.

This integration means your AI agent becomes a personal knowledge broker that understands your context—the connections between your research, your reading list, and your brainstorming sessions. When you pair this MCP with the Vinkius catalog of tools, you build an assistant capable of synthesizing complex ideas without ever needing to leave your chat window.

Core Capabilities

01 — List all available knowledge workspaces

The agent can show you every separate area or 'graph' where you keep notes.

02 — Retrieve and manage specific notes

You can ask the agent to pull up a note by title, list all notes in a workspace, or retrieve the full content of a single document.

03 — Build new permanent knowledge entries

The agent can create brand-new, structured notes and save them into your graph instantly.

04 — Log daily thoughts and summaries

You can append conversation summaries or quick ideas directly to today's date entry without creating a new note.

05 — Map related concepts

The agent finds all other notes that point back to a specific topic, revealing connections you might have forgotten about.

06 — Capture and organize links and books

It saves web bookmarks or imports highlights from your reading list so they are searchable alongside your text notes.

One Click on Vinkius — From Prompt to Execution

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

01 First, you authorize the Reflect MCP plugin within your active AI client extension.

02 Next, you secure your personal OAuth Access Token from your Reflect developer settings and embed it into the integration setup.

03 After connecting, you chat naturally with your agent, prompting tasks like, 'Find all notes connected to 'AI Strategy'.' or 'Summarize this meeting and log it to my daily note.'

The bottom line is that you tell the AI what to find or write using plain English, and it handles the complex data retrieval from your private graph.

Built For

This MCP is for anyone whose work relies on synthesizing information from multiple sources—the researcher who manages dozens of notes, the writer building an argument over months, or the consultant who needs to recall context from a project that happened last quarter.

Academic Researcher

You use this MCP to track connections between sources and theories. You ask your agent to run `get_backlinks` on key concepts, instantly seeing every paper or note where that idea has been mentioned.

Technical Writer

You rely on the agent to pull together decisions from multiple meetings and articles. You ask it to compile a new document using `create_note` with structured data gathered from your graph.

Consultant/Project Manager

When wrapping up a project, you prompt the agent to review all associated materials by listing related links and documents. You use `append_daily_note` to log key decisions for client reports.

What Changes When You Connect

01 Deep context retrieval: Instead of manually searching through dozens of files, just ask the agent to use `get_backlinks` and instantly see every note that referenced a key idea. It finds connections you might have forgotten about.

02 Structured logging: When you finish a call or meeting, tell your AI client to summarize it and log those decisions using `append_daily_note`. This keeps your main journal clean while ensuring no action item is lost.

03 Idea capture on the fly: Stop wasting time switching apps. You can ask the agent to create permanent notes with `create_note` or save a URL with `create_link`, keeping your brainstorming session uninterrupted.

04 Full knowledge visibility: Get an overview of all your work by first running `list_graphs`. This helps you understand which major workspaces contain related projects and ideas, making your entire system visible to the AI.

05 Reference material access: Your agent can handle external context. It shows you everything in `list_books` or pulls up a specific document using `get_note`, treating all your sources as equally accessible.

Real-World Applications

Synthesizing research for a grant proposal

A researcher has compiled notes on five different theories. They ask their agent to run `list_notes` across the project graph, then use `get_backlinks` on 'Quantum Computing' to pull together every piece of evidence related to that topic, creating a comprehensive draft outline.

Archiving client meeting takeaways

A consultant finishes a video call. Instead of copying bullet points into an email, they tell their agent to summarize the discussion and use `append_daily_note` in the main project graph. This permanently logs the decisions with context.

Reviewing old work for inspiration

A writer is starting a new book chapter. They ask their agent to list all notes related to 'Victorian London' using `list_notes`. The AI then pulls up key details from an archived note with `get_note`, providing immediate context.

Building a project roadmap

A founder needs to track external inspiration. They use the agent to save crucial industry articles via `create_link` and then ask for a summary of all saved links related to 'AI Ethics', ensuring they capture ideas from multiple sources.

Patterns to Avoid

Using notes as simple dumping grounds

✗ AVOID

Pasting huge blocks of text into random documents without linking them, making the information hard to find later.

✓ INSTEAD

Don't just dump data. If it's a core idea, use `create_note` and give it a clear subject. To connect it to existing work, always ask your agent to run `get_backlinks` on that new note.

Treating the graph like a simple file cabinet

✗ AVOID

Assuming that just listing notes (`list_notes`) is enough; failing to use the connections between them.

✓ INSTEAD

Don't stop at the list. After running `list_notes`, tell your agent, 'Now find all the backlinks for the top three items.' This reveals the true context of your knowledge.

Losing track of quick insights

✗ AVOID

Having brilliant ideas during a meeting but having to manually copy them into an external document later.

✓ INSTEAD

Keep it in Reflect. Immediately use `append_daily_note` or `create_note` while the idea is fresh. This guarantees the context and date are logged automatically.

The Right Fit

Use this MCP if your core job involves synthesis, relationship mapping, or recall from a large, interconnected body of work. You need to know *how* ideas connect—not just that they exist. The value here is in the 'backlinks' capability; it solves the problem of knowing what you already know about a topic. Don't use this if your primary task is simple data logging (like a CRM or calendar) or structured

transaction processing, as those tools manage discrete records better. If all you need to do is save a single list of names and emails, a dedicated contact manager works fine. But if you need the AI to look at 10 different notes and tell you what three common themes they share, this MCP is essential.

The struggle of keeping track of your own brain

Right now, knowledge lives in silos. You have a research note here, a highlight list there, meeting minutes in another app, and key ideas scattered across multiple documents. To build one cohesive argument, you're constantly switching tabs, copying bullet points into a master document, and cross-referencing data manually. It’s exhausting, slow, and always feels like you're forgetting that critical link between two seemingly unrelated topics.

With this MCP, the process becomes conversational. You simply ask your agent, 'What connections exist between my notes on AI Ethics and my bookmarks from last month?' The system pulls together information using tools like `get_backlinks` and presents a synthesized answer, not just a list of documents. It turns fragmented data into immediate insight.

Reflect Notes: Structured Knowledge Capture

Manual logging means you either forget to log the decision entirely, or you write it down in a generic text file that has no connection to the project. You lose the metadata—the context of *when* and *why* that decision was made.

Now, by using tools like `append_daily_note` or `create_note`, your knowledge is automatically anchored to its date, its graph, and its subject matter. Your captured insights become part of a living record, not just ephemeral text.

Reflect Notes: 10 Available Tools

These tools allow your AI agent granular control over every aspect of your personal knowledge base—from listing graphs to appending daily thoughts.

#	TOOL	DESCRIPTION
01	`append_daily_note`	Adds formatted text content directly into today's date entry in your journal.
02	`create_link`	Saves a new web link or bookmark to one of your defined knowledge graphs, automatically pulling metadata.
03	`create_note`	Generates an entirely new note within a graph, allowing you to specify the subject and content upfront.
04	`get_backlinks`	Retrieves a list of all other notes that contain links pointing back to a specific piece of content.
05	`get_current_user`	Fetches basic profile details for the Reflect user who is logged into your AI client.
06	`get_note`	Pulls up the full text and metadata for a specific, existing note in your graph.
07	`list_books`	Displays all books you have saved or imported into the Reflect system.
08	`list_graphs`	Shows a list of every separate workspace or 'graph' you currently use in Reflect Notes.
09	`list_links`	Lists all the web bookmarks you have saved within a specific knowledge graph.
10	`list_notes`	Generates a list of every note contained within one designated Reflect graph.

See It in Action

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

U List all available graphs in my Reflect account.

Using `list_graphs`, I found 2 distinct workspaces: "Personal Brain" (ID: 6f...2c) and "Project Insights" (ID: 9a...11).

U Create a permanent note titled 'Meeting 2024 Strategy' inside my 'Personal Brain' graph with summary bullet points.

Executing `create_note` on Graph ID '6f...2c'. I have successfully created 'Meeting 2024 Strategy' and piped the Markdown list of bullets into it.

U Find notes linked by backlinks that point to my note 'React Learnings'.

I called `list_notes` to verify 'React Learnings', acquiring its note_id 'xx82', then ran `get_backlinks`. You have 3 other notes connecting to it: 'Frontend Architecture', 'Day 2 Journal', and 'Tech Stack Overview'.

Frequently Asked Questions

01 **How does Reflect MCP help me find connections between my notes?**

It uses the `get_backlinks` tool to scan your entire graph and list every single note that points back to a specific concept. This reveals context you might have otherwise missed.

02 **Can Reflect MCP write new content for me?**

Yes, it can create entirely new notes using `create_note` or append structured updates to your journal via `append_daily_note`. You just give the agent the material and tell it where to put it.

03 Do I need a specific client app to use Reflect MCP?

No, as long as your AI client supports the Model Context Protocol (MCP) standard, you can connect this integration. This means compatibility across most modern agents and coding environments.

04 What is the difference between `list_notes` and `get_note` in Reflect MCP?

`list_notes` gives you a directory of every note in a graph, showing titles. `get_note` pulls up the actual content and metadata for one specific note when you know its title or ID.

05 How do I save external information using Reflect MCP?

You use the `create_link` tool to bookmark URLs. The agent handles extracting relevant metadata automatically, so your bookmarks are searchable alongside everything else in your graph.

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 `"reflect": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → 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

Reflect 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 Reflect. 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	June 2026
MCP Server	Reflect MCP
Server ID	`019d75fe-08aa-71cd-bd2f-ed8415f79544`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

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/reflect.