MCP SERVER | NO CODE | CLOUD HOSTED

Mattermost MCP

Audit team history and manage communication flows.

Mattermost (Secure Team Collaboration) MCP gives your AI agent full control over mission-critical team communication. Use it to search across every channel, audit user roles, manage complex message threads, and ensure compliance by programmatically inspecting or modifying chat history.

A+ Quality Score 100/100

team-messaging | channel-management | collaboration | compliance-auditing | workflow-automation | user-management

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/mattermost-secure-team-collaboration

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

Mattermost (Secure Team Collaboration) MCP

10 tools available
Cloud-hosted on Vinkius

Need to keep tabs on what's happening in a highly regulated team environment? This MCP connects your Mattermost instance directly to your AI agent, letting you manage the whole communication lifecycle without ever clicking through the UI. You can send formatted messages that include specific user mentions, or use fuzzy search to track down hidden channels across your entire infrastructure. Beyond simple chat, you can enumerate every active member and check their roles for compliance audits. If a historical conversation is key, this MCP lets you retrieve exact message timelines from any channel. It's all managed through natural conversation via Vinkius—you just tell your agent what data points you need, and it pulls them directly into your workflow.

Core Capabilities

01 — Find hidden channels

Scan the entire Mattermost workspace to locate public or private channels by name.

02 — Retrieve message history

Get a precise chronological list of all posts from a specific channel at any time.

03 — Manage team members and roles

List active users, verify their system roles, or get the necessary IDs to route mentions correctly.

04 — Send structured posts

Dispatch a formatted message into any channel, including specific user tags.

05 — Audit and modify messages

Update existing chat records or delete them entirely, all while retaining crucial audit timestamps for compliance.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/mattermost-secure-team-collaboration** — connect your AI agent in three steps.

01 Subscribe to this MCP on Vinkius.

02 Provide your Mattermost Host URL and a personal access token.

03 Your AI agent manages the workspace, allowing you to query team data or send messages through natural conversation.

The bottom line is you manage secure team collaboration by directing your agent with conversational commands.

Built For

This MCP is for the security analyst who can't trust manual log reviews, or the engineering lead drowning in Slack threads. If your job requires knowing who said what, when, and why—you need this.

Security Analyst

Audits channel memberships and message timelines to investigate potential security breaches or ensure communication policy adherence.

IT Operations Engineer

Monitors server activities, retrieves team structure details, and sends automated system alerts directly from the chat interface.

Engineering Lead

Coordinates cross-functional updates by searching for historical technical discussions across dozens of different Mattermost channels efficiently.

What Changes When You Connect

01 Track down hidden discussions. Use the `search_channels` tool to find public or private channels by name, no matter how deep they are nested in your infrastructure.

02 Maintain a clear record of events. The `get_channel_posts` tool retrieves exact message graphs, giving you full visibility into project status and historical conversations.

03 Manage the team structure easily. You can use `get_teams` to list all global workspaces, which is necessary for routing complex organizational queries.

04 Ensure compliance always. Use `update_post` to edit chat contents while guaranteeing that audit timestamps are visibly preserved in the record.

05 Know exactly who you're talking to. The `get_all_users` tool maps every active human and bot ID, so mentions always route correctly without guessing usernames.

Real-World Applications

Investigating a data leak

A security analyst needs to know if a sensitive client name was mentioned in a private chat from three weeks ago. They use the `search_channels` tool, pinpointing the correct team, and then call `get_channel_posts` to pull the exact messages needed for their report.

Onboarding new global teams

An IT Ops engineer needs to build a system alert that notifies three different department leads. They use `get_all_users` to get every required user ID, then use the `create_post` tool to dispatch a message with all necessary mentions.

Revising historical technical specs

An engineering lead realizes they need to correct a misstated metric from an old thread. They retrieve the original post using `get_channel_posts`, then use the `update_post` tool to fix the number while keeping the audit trail intact.

Auditing team access rights

A compliance officer needs proof of who is authorized to communicate with a specific department. They call `get_team_members` to list everyone and verify their current user roles against company policy.

Patterns to Avoid

Manually searching for old data

✗ AVOID

A team member spends 20 minutes clicking through dozens of channels, copy-pasting snippets into a spreadsheet just to build a picture of project status.

✓ INSTEAD

Instead, ask your agent to use `get_channel_posts` and give you the exact message graph from the required channel. You get the data instantly, structured and ready for analysis.

Forgetting user IDs

✗ AVOID

A developer tries to tag a specific person but uses their common name instead of their unique Mattermost ID, so the mention fails and nobody is notified.

✓ INSTEAD

First, run `get_all_users` to map every active identity. Then, use that authoritative list to ensure your mentions are correctly routed when you call `create_post`.

Assuming a message was deleted

✗ AVOID

A manager thinks a conversation is gone, so they ask for it. They waste time asking about chat history that might not exist or is restricted.

✓ INSTEAD

Use `get_channel_details` to inspect the channel's internal properties first. This tells your agent if there are structural limitations before you attempt any data retrieval.

The Right Fit

Use this MCP if your job requires controlling, auditing, or extracting structured information from team communications—anything that goes beyond simple conversation flow. You need it when the integrity of the record (like checking user roles via `get_team_members` or preserving audit logs with `update_post`) is more important than speed. Don't use this if you just want to quickly send a single, one-off announcement; your local client can do that. However, if you only need to know the name of a channel and not its contents, then `search_channels` or `get_channel_details` are sufficient. If you absolutely must delete messages for compliance reasons, rely on `delete_post`, but understand this MCP is built for deep operational control, not simple messaging.

The pain of chasing conversations in chat platforms

Think about the time sink: you need to know what was discussed last week regarding Project X. You open Mattermost and start clicking—Channel A, Channel B, then you remember it might have moved to a private thread in Team Z. You end up jumping between tabs, manually scrolling through threads, copying key dates, and pasting them into your notes just to build a timeline.

With this MCP, the process changes entirely. Instead of clicking, you tell your agent: 'Show me the full timeline for Project X.' Your agent runs `get_channel_posts` and instantly delivers the structured message history. You get immediate context without ever leaving your workflow.

Control chat data with Mattermost MCP

Manual control means relying on human memory or the platform's basic search, which often misses hidden channels. You can't programmatically list every team structure, nor can you verify if a user has changed roles since last quarter.

This MCP gives your agent deep system access. It lets you run `list_team_channels` to map out the entire network and use `get_team_members` to audit who belongs where. You move from guessing what's in the chat to knowing exactly what data points exist.

Mattermost (Secure Team Collaboration) MCP - 10 Tools

This collection of tools lets you interact with every aspect of your Mattermost workspace—from finding hidden channels to updating posts and auditing user roles.

#	TOOL	DESCRIPTION
01	`create_post`	Sends a formatted message payload directly into any specified channel.
02	`delete_post`	Removes an explicit text post from the Mattermost chat history.
03	`get_channel_details`	Retrieves deep internal information about a specific chat channel.
04	`list_team_channels`	Finds and extracts the list of public channels within an entire team structure.
05	`get_channel_posts`	Gathers the full, precise timeline matrix of messages from a given channel.
06	`get_team_members`	Lists all currently attached user accounts and their capabilities within a team.
07	`get_teams`	Identifies the main global workspaces or parent teams in your Mattermost setup.
08	`get_all_users`	Provides a full map of every active user and bot ID on the server, ensuring correct mention routing.
09	`search_channels`	Scans the entire database to discover any hidden or public chat channel by name.
10	`update_post`	Changes the content of an existing message while automatically preserving audit timestamps.

See It in Action

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

U List all teams available in my Mattermost instance

I've retrieved 2 teams: 'Engineering' (ID: team-123) and 'Security-Ops' (ID: team-456). Which one would you like to list the channels for?

U Search for a channel called 'product-alerts' in the Engineering team

Searching... I've identified the channel '#product-alerts' (ID: chan-987). It is a public channel used for system notifications. Would you like me to fetch the last 10 posts from this channel?

U Send a post to channel 'chan-987': 'Backend migration complete. @alex please verify metrics.'

Post sent successfully to #product-alerts. Your message including the mention for @alex has been dispatched. New Post ID: 'post-13579'. I can monitor for any replies if you'd like.

Frequently Asked Questions

01 **How does Mattermost MCP handle compliance auditing?**

It maintains a robust audit trail even when messages are changed. The `update_post` tool allows you to edit content while automatically preserving the original audit timestamps, ensuring traceability.

02 **Can I find channels that aren't visible in my team?**

Yes. You can use the `search_channels` tool to aggressively scan the database for any hidden or public channel name across your entire Mattermost instance, regardless of visibility settings.

03 **Does Mattermost MCP let me delete posts completely?**

The `delete_post` tool allows you to remove explicit text posts from the chat. This action is irreversible and designed for programmatically clearing records when needed.

04 **How do I make sure mentions work correctly?**

You should use the `get_all_users` tool first. It provides a complete map of all active human and bot identities, guaranteeing your agent routes mentions accurately via their unique ID.

05 **What is the difference between get_teams and list_team_channels?**

Use `get_teams` to identify the global parent workspaces. Then, use `list_team_channels` on those teams to find all the specific public channels within them.

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 `"mattermost-secure-team-collaboration": { "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

Mattermost (Secure Team Collaboration) 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 Mattermost (Secure Team Collaboration). 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	Mattermost (Secure Team Collaboration) MCP
Server ID	`019d75d0-64b5-72f2-8b43-b18bb9cc9e25`
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/mattermost-secure-team-collaboration.