MCP SERVER | NO CODE | CLOUD HOSTED

Groove MCP

Manage support tickets and customer data with conversation.

Groove MCP connects your entire helpdesk to any AI agent, letting you manage customer support tickets through natural conversation. You can list all open issues, summarize long conversations, draft replies, update ticket statuses, and pull up full customer records—all without opening the main Groove dashboard.

A+ Quality Score 100/100

shared-inbox | ticket-management | customer-service | support-automation | helpdesk-api | conversation-tracking

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

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.

Groove MCP

12 tools available
Cloud-hosted on Vinkius

You don't have to jump between multiple tabs or remember complex workflows just to handle a customer inquiry. With this MCP, you use your AI client to talk directly to your helpdesk data like it's an extension of your own brain. Need to know what happened in ticket #405 last week? Just ask. Want to update the status from 'Pending' to 'Closed'? Tell your agent. It handles the API call for you. You manage everything—from creating a new ticket to getting details on who the customer is and checking if an agent was active—all within your chat window. This level of control means less time clicking buttons and more time actually helping customers. If you use Vinkius, this MCP adds Groove's full power into your existing AI workflow, making complex support operations feel as simple as sending a message.

Core Capabilities

01 — Reviewing and Tracking Tickets

List all tickets across the account or fetch the complete conversation history for any specific ticket number.

02 — Writing Replies and Notes

Draft and send official replies to customers, or add internal notes directly into a support thread.

03 — Managing Ticket Statuses

Change the status of any ticket—opening it, putting it on hold, or marking it as resolved.

04 — Retrieving Customer and Agent Data

Look up specific customer profiles by email, list all active agents, or view details for a particular support agent.

05 — Coordinating Mailboxes

List and inspect every configured mailbox ID to keep track of your multi-channel support setup.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP on Vinkius and paste your Groove API Access Token into the connection settings.

02 Next, tell your AI client exactly what you need—for example, 'List all open tickets for billing issues,' or 'Get customer details for jane.doe@example.com.'

03 Your agent sends a clean command to Groove, and the results appear immediately in the chat window.

The bottom line is that you treat your entire helpdesk as an interactive conversation, not a dashboard full of forms.

Built For

This MCP is for support leads and customer success managers who spend too much time juggling dashboards. It helps the small business owner who needs to close tickets fast without logging into the main application.

Support Lead

Needs to quickly summarize long ticket threads, audit agent responses, and monitor overall team capacity using natural language commands.

Customer Success Manager (CSM)

Handles high-priority cases by checking customer profiles and updating details or closing tickets directly from the chat interface.

Small Business Owner

Manages daily customer inquiries, ensuring resolved tickets are properly closed without needing to navigate complex back-end menus.

What Changes When You Connect

01 Instead of manually opening the main dashboard, you use your agent to list all available tickets instantly. You just ask for 'open billing issues' and get a clean summary list without any clicks.

02 You maintain full visibility into customer history by using the get_customer tool. This lets you cross-reference contact records with ticket data right in your chat window, making context retrieval instant.

03 Sending replies is simple; instead of finding the right ticket thread and composing a message, you tell your agent to 'Draft a reply on #402.' The create_message function handles it.

04 Need to keep track of who's doing what? You can use list_agents to see all team members online or check individual productivity details without leaving the chat flow.

05 Managing ticket status is streamlined. Use update_ticket_state to change a ticket from 'Pending' to 'Closed,' and you get confirmation immediately, keeping your records accurate.

Real-World Applications

Handling an Urgent Customer Inquiry

A CSM gets a chat request for customer Jane Doe. Instead of searching the dashboard, they ask their agent to run 'get_customer' on her email. They see all past issues and then use 'list_messages' on ticket #405 to find the specific error code needed to write a perfect reply using create_message.

End-of-Day Report Generation

A Support Lead needs to know how many tickets were opened today and who handled them. They ask their agent to 'list_tickets' for the last 24 hours, getting a clean count and list of ticket numbers that they can then use to update_ticket_state for closure.

Onboarding New Team Members

A manager wants to train a new agent on team capabilities. They ask their agent to 'list_agents' and also check the status of all configured channels by running 'list_mailboxes,' ensuring nothing is missed.

Handling Multiple Inquiry Types

A small business owner receives a mix of billing inquiries and login issues. They first run 'list_tickets' to see the queue, then use get_ticket on a specific number, summarizing the details before using create_message to send out a resolution update.

Patterns to Avoid

Trying to find data across multiple systems

✗ AVOID

Copying customer emails from a separate CRM into Groove, then manually creating tickets one by one because the tool seems too complex.

✓ INSTEAD

Don't copy-paste. Use your agent to 'get_customer' first, pulling all necessary details and history in one go. Then, use create_ticket if needed.

Forgetting to update the ticket status

✗ AVOID

Replying to a customer problem but forgetting to mark the ticket as resolved because you have to navigate away from the chat interface.

✓ INSTEAD

After sending your reply using create_message, immediately prompt your agent with 'update_ticket_state' and tell it to close or change the status.

Asking for data without context

✗ AVOID

Just asking 'Show me ticket information' without providing a number. The system fails because it doesn't know which record you mean.

✓ INSTEAD

Be specific. Always provide the ticket number and ask your agent to 'get_ticket' by its unique ID.

The Right Fit

Use this MCP if your primary bottleneck is context switching—if handling a single customer requires opening three different dashboards (the main ticket view, the user profile, and the internal notes log). It lets you consolidate all that data retrieval into conversation. However, don't use it if your goal is pure content generation or advanced scripting logic; for those tasks, an API-first development tool might be better suited. If you just need to build a simple web form, avoid this MCP and look at dedicated database connectors instead.

Support teams spend hours clicking through tabs they should never have to open.

Right now, handling one customer usually means opening the main ticket dashboard. Then you click on 'Customer Profile' in a sidebar tab to see their history. Next, you might switch to another app just to check if an agent was assigned. You then copy that data and paste it into a draft reply. It’s clicking, switching, and copying, every single time.

With this MCP, all of that happens conversationally. You tell your agent what needs doing—like 'summarize the last three interactions for customer X'—and you get the full context right where you are. Your AI client does the heavy lifting across those different systems so you just read the answer.

Groove MCP gives you total control over ticket creation and status updates.

You no longer have to manually navigate deep into the dashboard settings. You can initiate a new record using create_ticket, or if an issue is resolved, use update_ticket_state to mark it closed without ever touching the 'Status' dropdown menu.

This capability means your workflow moves at conversational speed. The system knows exactly what needs changing and executes it instantly, eliminating manual administrative steps entirely.

Groove MCP: 12 Tools for Customer Support Management

These twelve tools let you handle every aspect of helpdesk operations, from listing all tickets to updating their final status, through conversational commands.

#	TOOL	DESCRIPTION
01	`create_message`	Adds a new reply or internal note directly into an existing ticket conversation.
02	`create_ticket`	Generates a brand-new helpdesk ticket record in the system.
03	`get_agent`	Retrieves specific details about an agent using their email address.
04	`get_customer`	Pulls up all records and contact history for a customer based on their email.
05	`get_mailbox`	Retrieves details for one of your configured support mailboxes using its unique ID.
06	`get_ticket`	Fetches all information about a specific ticket by using its assigned number.
07	`list_agents`	Provides a list of every agent currently set up in the Groove account.
08	`list_customers`	Returns a comprehensive list of all customers recorded in your account.
09	`list_mailboxes`	Lists every mailbox ID configured for multi-channel support management.
10	`list_messages`	Pulls a complete list of all messages and comments associated with a ticket.
11	`list_tickets`	Retrieves a comprehensive, filterable list of every support ticket in the system.
12	`update_ticket_state`	Changes the official status of an existing helpdesk ticket (e.g., open to closed).

See It in Action

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

U List my open tickets in Groove

Retrieving your open tickets... I found 5 tickets requiring attention, including #402: 'Billing Inquiry' and #405: 'Login Issue'. Would you like me to summarize the latest message for the login issue?

U Show the messages for ticket #405

Fetching conversation history... Ticket #405 contains 3 messages. The customer reported they cannot reset their password. There is also an internal note from Agent Sarah mentioning a potential database sync delay.

U Reply to ticket #402: 'Your refund has been issued and should appear in 3-5 days'

Sending reply... Success! I've posted the message to Ticket #402 informing the customer about their refund. The ticket state remains 'pending'. Should I mark it as closed?

Frequently Asked Questions

01 **How do I find all open tickets using the Groove MCP?**

You use list_tickets to get a comprehensive view of your helpdesk. This function lets you see every ticket that needs attention across your entire account.

02 **Can I add internal notes without replying to the customer? (Groove MCP)**

Yes, you can use create_message and specify it as an 'internal note.' This lets team members communicate about a ticket's status or findings without alerting the customer.

03 **What if I need to see all my configured support channels? (Groove MCP)**

You use list_mailboxes. This function pulls up every mailbox ID you have set up, ensuring no channel is missed when managing multi-channel support.

04 **Does the Groove MCP let me create new customer records?**

No, it doesn't create customers from scratch. However, you can use get_customer to pull existing details and full contact history for any known email address.

05 **Is list_messages better than get_ticket? (Groove MCP)**

They serve different purposes. Use get_ticket to retrieve the main metadata about the issue itself, but use list_messages when you need a full feed of every comment and communication exchange.

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 `"groove": { "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

Groove 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 Groove. 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	Groove MCP
Server ID	`019d75ab-dda7-7292-97a9-f741519666cb`
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/groove.