

MCP SERVER

NO CODE

CLOUD HOSTED

Front MCP

Manage Shared Inboxes and Customer Threads

Front MCP manages shared team communication across email, SMS, and chat. It lets your AI agent track active conversations, list all inboxes, fetch full message histories, and send replies directly—all without leaving your client interface.

A+ Quality Score 100/100

shared-inbox

email-management

team-collaboration

customer-operations

thread-tracking

sms-integration



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.

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.

Front MCP

12 tools available
Cloud-hosted on Vinkius

Managing customer communications used to mean juggling tabs: checking the support inbox, finding the right thread history, then drafting a reply while keeping track of who owns it. Now, you can handle that whole flow inside your AI agent. This MCP connects your Front account directly, giving your agent the power to act like a full-time operations employee. You'll get real-time visibility into every shared inbox and conversation status, so you never miss an assignment or lose context on a reply. Need to update a ticket from 'open' to 'resolved'? Your agent handles that instantly. It even lets you find all your team members or client contacts before replying. Getting this connected through Vinkius means you only sign up once for the entire catalog, giving you deep control over every communication point.

Core Capabilities

01 – View and organize shared inboxes

List all team mailboxes and retrieve specific conversations routed to them.

03 – Review full message history

Pull the complete thread history for a specific interaction so you know the whole story before replying.

05 – Update ticket status automatically

Change a conversation's status (like archiving or reopening) programmatically after resolving an issue.

02 – Search for customer interactions

Find any conversation across your account, checking its current status or who it's assigned to.

04 – Send replies on behalf of the team

Draft and send responses to active conversations directly from your agent chat window.

One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to this MCP through Vinkius and provide your Front API Token, found in your developer settings.
- 02 Connect your AI client—whether it's Cursor or Windsurf—using the credentials you just provided.
- 03 Tell your agent what you need. For example, 'Find all open conversations in the Support inbox that mention billing issues.' The agent executes the command and brings back the data.

The bottom line is your AI client uses this MCP to speak the language of Front, turning complex backend operations into simple chat commands.

Built For

This connector is for customer support managers and ops engineers tired of switching between dozens of tabs just to get a full picture of what's happening in the queue. If you manage client communications, this saves your team hours every week.

Customer Support Agent

Needs to quickly check open conversations and pull message history without leaving their agent interface while drafting a reply.

Operations Manager

Automates the archiving of resolved issues or verifies team member assignments across multiple shared inboxes.

Account Manager

Gets a real-time overview of client communications and sends quick, context-aware follow-ups via simple AI commands.

What Changes When You Connect

- 01 Never lose context on a reply again. Use the `list_conversation_messages` tool to fetch the entire thread history instantly, so your response is accurate.

-
- 02** Keep your team focused. You can use `list_shared_inboxes` to see all operational mailboxes in one place and find exactly where an issue landed.
-
- 03** Automate cleanup tasks. After resolving a ticket, use `update_conversation_status` to archive the conversation right away, keeping the open queue clean.
-
- 04** Collaborate without leaving your agent. You can send replies using `send_inbox_reply`, making it look like you're typing directly into Front.
-
- 05** Know who owns what. Use `search_conversations` and `get_conversation_details` to quickly identify which team member is responsible for a given client issue.
-

Real-World Applications

Handling complex billing inquiries

A user needs to respond to a payment question. They first use `search_conversations` to find all open billing issues, then call `list_conversation_messages` on the top result to understand the full history before drafting a response with `send_inbox_reply`.

Onboarding new team members

A manager needs to know who can work in support. They use `list_inbox_teammates` to get a roster of available agents, ensuring accurate assignment before checking if the inbox is already assigned via `get_conversation_details`.

Tidying up closed tickets

An ops manager finishes reviewing a week's worth of resolved support issues. They call `list_inbox_threads` to see what was handled, and then run `update_conversation_status` on all completed threads to clear the active queue.

Getting a client overview

An account manager needs quick visibility on a specific client. They use `list_address_book` to pull contact details, then run `search_conversations` using the client's name to see all recent interactions.

Patterns to Avoid

Assuming status changes are manual

X AVOID

A user reads a ticket is done and manually tells their agent, 'It should be closed.' The agent has no way to confirm the official change.

✓ INSTEAD

You must use `update_conversation_status` to programmatically change the conversation's status. This ensures your changes are recorded correctly in Front.

Only looking at the subject line

X AVOID

The agent only reads the email subject and replies, assuming the full context is clear.

✓ INSTEAD

Always call `list_conversation_messages` first. This pulls the entire message thread history, giving you the complete story before you write a single word.

Sending messages without knowing who to CC

X AVOID

The agent sends a follow-up reply but forgets to loop in the account manager or correct team member.

✓ INSTEAD

Use `list_inbox_teachmates` first. This gives you the full roster of available agents, ensuring your replies are routed to everyone who needs to see them.

The Right Fit

You use this MCP if your core pain point is managing communication volume and context across shared mailboxes. You need tools that let your agent read history (`list_conversation_messages`), find the right conversations (`search_conversations`), and act on them (replying or updating status). Don't use this if you are primarily trying to manage internal team documents, HR records, or billing systems outside of Front. If your goal is just data retrieval without action, a basic read-only connection might suffice. But since you need to *do* things—like marking tickets resolved with `update_conversation_status`—this MCP gives you the full operational control layer.

The endless cycle of context switching and manual data fetching.

Today, when a customer emails in, you have to jump between tabs: check the shared inbox for the latest message, click into the conversation view to read the whole history, then copy key details, open your drafting tool, write the response, and finally switch back to Front to manually update the ticket status. It's slow, error-prone, and exhausting.

With this MCP integrated through Vinkius, all those steps vanish. Your agent handles it in one chat session. You ask for a resolution; the agent finds the history using `list_conversation_messages`, drafts the reply, sends it with `send_inbox_reply`, and marks the issue resolved via `update_conversation_status`. It's seamless.

Use Front MCP to handle every communication step.

You stop manually listing teammates or searching for conversation IDs. The agent manages that behind the scenes, allowing you to focus entirely on solving the customer's problem instead of managing the workflow itself.

Your work shifts from clicking through dashboards and copying data points to simply commanding your AI client what needs to happen. It's a massive upgrade in efficiency.

Front with 12 Tools

These tools give your agent direct access to every function of your Front account, from listing contacts to updating conversation status.

#	TOOL	DESCRIPTION
01	<code>verify_api_status</code>	Checks your connection to Front to make sure everything is working correctly.
02	<code>get_conversation_details</code>	Retrieves basic metadata about a specific customer conversation, like who it's assigned to.
03	<code>get_inbox_details</code>	Fetches general information and status details for an entire shared inbox.
04	<code>list_address_book</code>	Pulls a list of all customer contacts stored in your Front account.
05	<code>list_all_conversations</code>	Gets a comprehensive list of every conversation existing in your account.
06	<code>list_inbox_threads</code>	Lists all the active conversations within a specific shared inbox.
07	<code>list_shared_inboxes</code>	Retrieves a list of every shared mailbox your team uses.
08	<code>list_conversation_messages</code>	Gets the full, chronological message history for any given conversation thread.
09	<code>list_inbox_teachmates</code>	Retrieves a list of all team members who have access to your shared inboxes.
10	<code>search_conversations</code>	Searches the entire platform for conversations based on criteria like status or keyword.
11	<code>send_inbox_reply</code>	Sends a reply to an active conversation thread from your agent interface.
12	<code>update_conversation_status</code>	Changes the status of a conversation, such as marking it resolved or assigning it to someone else.

See It in Action

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

U List all shared inboxes in my Front account.



Retrieving inboxes... I found 3 shared inboxes: 'Support' (ID: inb_123), 'Sales' (ID: inb_456), and 'General Info' (ID: inb_789).

U Search for open conversations in the Support inbox.



Searching conversations... I found 4 open conversations matching your query. Two are currently assigned to 'John', and two remain unassigned.

U Archive conversation 'cnv_987'.



Conversation updated! I have successfully changed the status of conversation cnv_987 to 'archived'. It will no longer appear in the open queue.

Frequently Asked Questions

01 How do I get an API Token for Front?

Log in to your Front account, click the gear icon (Settings), go to 'Developers' > 'API Tokens', and create a new token with 'Shared Resources' scope.

02 Can I search for specific types of conversations?

Yes, use the 'search_conversations' tool. It accepts standard Front search syntax, such as 'is:open' or 'inbox:support'.

03 Is it possible to reply to an email via the agent?

Yes! Use the 'send_inbox_reply' tool. You must provide the Conversation ID, the message body, and the Teammate ID of the sender.

04 How do I archive a conversation?

Use the 'update_conversation_status' tool and set the status parameter to 'archived'. This updates the global status for the shared inbox.

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 `"front": { "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

Front 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](https://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 Front. 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	Front MCP
Server ID	019d75a1-59a0-713a-ac52-51bece568fd8
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/front.