MCP SERVER | NO CODE | CLOUD HOSTED

Front MCP

Manage every customer message in one place.

Front MCP connects your AI agent directly to your unified support inbox, managing conversations across email, SMS, chat, and social media. Instead of switching between platforms, your agent handles everything from triaging new tickets to updating contact details, giving your team one source of truth for every customer interaction.

A+ Quality Score 100/100

shared-inbox | team-collaboration | email-management | customer-operations | conversation-tracking | unified-inbox

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.

Front MCP

12 tools available
Cloud-hosted on Vinkius

Your agent uses this MCP to take over the complexity of modern customer support. It pulls data from all your communication channels—email, chat, SMS, and social media—and treats it like a single shared inbox. You can ask your AI client to find every open ticket for 'Acme Corp' across three different platforms, or tell it to update a contact's phone number based on the last message received.

This means you don't have to jump between separate tools just to get the full story. Your agent organizes conversations, changes their status (open, archived, spam), and even sends replies, all from your preferred client. This centralized control makes it look like having a dedicated support coordinator sitting right next to you. By connecting through Vinkius, you give your AI agent access to this entire suite of team collaboration tools.

Core Capabilities

01 — View and filter all active channels

You can see every communication channel—email, chat, SMS, or social—in one place without manually checking each tab.

02 — Find specific customer conversations

Search across thousands of messages to quickly pinpoint threads using keywords or conversation IDs.

03 — Update and assign tickets

Change the status of a conversation, marking it open, archived, or spam. You can also reassign ownership within your team.

04 — Get customer contact information

Retrieve full details about any person involved in a chat, like their name and phone number.

05 — Read complete message history

Pull the entire transcript for any conversation, including messages from all parties and channels.

06 — Send replies to conversations

Draft and send a response to a customer directly through your agent without ever leaving your workflow.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP on Vinkius.

02 Next, you'll retrieve your API token from Front (Settings > Developers > API Tokens).

03 Finally, connect the token and start managing your unified inbox using your AI client.

The bottom line is that once connected, your agent automatically acts as your dedicated support coordinator, reading and writing to all your shared inboxes.

Built For

This MCP is for Ops Leads and Support Managers who are tired of juggling multiple dashboards—one for email, one for chat, and another for social. If you spend half your day just aggregating context from different platforms before replying to a customer, this is for you.

Support Team Lead

Uses the MCP to instantly triage high-volume inboxes, assigning urgent conversations and updating statuses using natural language commands.

Customer Success Manager

Retrieves full interaction histories for a customer across all channels and updates their contact details without ever leaving their primary workspace.

Operations Analyst

Monitors channel activity, manages team assignments across shared mailboxes, and gets a comprehensive overview of active support streams.

What Changes When You Connect

01 Instead of manually checking email, SMS, and social media tabs, your agent uses `list_active_channels` to monitor all streams from a single interface. You get real-time visibility into every incoming customer request.

02 You never lose context again. Use `get_conversation_details` or `list_conversation_messages` to pull the full history for any ticket, regardless of how many channels it passed through. This gives your team total operational visibility.

03 Triage high-volume support using natural language. You can tell your agent to find all open tickets related to 'billing' and then use `update_conversation_status` to mark them as assigned or resolved instantly.

04 Keep your customer records clean by first calling `get_contact_info`. This lets you ensure you have the most up-to-date details before replying, saving time on verification steps.

05 Replying is simple. Once the conversation is ready, simply ask your agent to use `reply_to_conversation`, and it sends the message directly without forcing you to switch tabs or copy/paste anything.

Real-World Applications

A customer asks about a feature on Twitter, but needs billing help.

The Ops Analyst tells their agent: 'Find all conversations mentioning 'billing' and 'feature request'.' The agent uses `search_conversations_by_query` to pull the thread from Twitter and combines it with any recent emails, allowing the analyst to reply accurately without missing context.

A high-priority support ticket comes in via SMS.

The Support Lead asks: 'List all open conversations that are marked as urgent.' The agent uses `list_conversations` and checks the channel, immediately notifying the correct team member who can then use `update_conversation_status` to assign it.

A customer calls back after receiving an initial email reply.

The CS Manager asks: 'Get all messages for conversation ID cnv_123.' The agent uses `list_conversation_messages`, retrieving both the original email thread and the new SMS transcript, letting the manager craft a single response.

A team needs to confirm if a contact is correct before sending data.

The Operations Analyst asks: 'What are the details for John Doe?' The agent runs `get_contact_info` and returns the verified phone number, ensuring the subsequent communication is accurate.

Patterns to Avoid

Trying to manually check multiple inboxes

✗ AVOID

Logging into your email client, then switching over to the chat platform dashboard, and finally checking the social media inbox just to gather context on one issue.

✓ INSTEAD

Just ask your agent to `list_active_channels` first. Then tell it: 'Find all conversations about X.' It handles the multi-platform aggregation for you.

Not knowing which conversation is active

✗ AVOID

Wasting minutes scrolling through old threads because you aren't sure if a ticket was resolved or just put on hold.

✓ INSTEAD

Use `list_conversations` to see the current status of every thread, and then use `update_conversation_status` when your team has finished working on it.

Replying without checking contact details

✗ AVOID

Assuming you have the right phone number or department name for a customer because you saw it once weeks ago.

✓ INSTEAD

Always start by asking the agent to `get_contact_info` and confirm all necessary data points before drafting any replies.

The Right Fit

Use this MCP if your primary bottleneck is *context*—the time spent gathering information from disparate communication systems (email, chat, SMS). If you need an AI agent to act as a unified support desk that can see the full customer journey across all channels, this is the right tool. You must be managing multiple shared inboxes and require programmatic access to conversation status updates.

Don't use this if your goal is purely internal knowledge management (e.g., searching only company documents). For that, a document indexing MCP would work better. Also, don't use it just because you need to send mass marketing emails; this focuses on 1:1 customer support conversations. If your team struggles with finding the right conversation or updating its status manually, this is what you need.

The Headache of Context Switching

Today, handling one customer interaction means logging into at least three different platforms. You check your email for the initial complaint; then you switch to the chat dashboard to see if they followed up there; and finally, you jump to the social media tab just to confirm their account handle. Every click is a moment where critical context can be lost or forgotten.

With this MCP, your agent pulls all these channels into one view. You ask it to find a customer's history, and it aggregates every piece of data—the email thread, the chat logs, the SMS texts. The result? Your AI client gives you the single source of truth immediately.

Front MCP: Full Conversation Control

The manual steps that disappear are the tab switching, the copy-pasting of IDs, and the status update clicks. You no longer need to log in five times a day just to get an overview.

Now, you tell your agent what to do—'Send a follow-up on cnv_123' or 'Check their phone number.' The system executes it across all channels automatically. It’s not just better organization; it changes the speed of service.

Front: 12 Tools for Communication Operations

These tools let you programmatically list conversations, manage team contacts, search message history, and send replies across all your connected communication channels.

#	TOOL	DESCRIPTION
01	`reply_to_conversation`	Sends a direct message to a customer within an existing conversation thread.
02	`get_contact_info`	Retrieves and shows you the contact details for any person involved in a chat.
03	`get_conversation_details`	Gathers specific information about an entire conversation thread, like who was involved or when it started.
04	`get_api_status`	Checks the connection status to ensure your AI agent is still linked up correctly.
05	`get_message_content`	Reads and displays the full text content of a single message within a chat.
06	`list_active_channels`	Lists all communication channels currently connected to your team's shared inboxes.
07	`list_team_contacts`	Shows a list of contacts and user profiles available within your organization.
08	`list_conversations`	Retrieves a list of all active customer conversations in the shared inbox.
09	`list_shared_inboxes`	Lists all the dedicated team inboxes your organization manages.
10	`list_conversation_messages`	Gets a full, detailed history of messages for a specific conversation thread.
11	`search_conversations_by_query`	Finds and lists conversations using search terms or specific keywords across the entire system.
12	`update_conversation_status`	Changes the status of a conversation, such as marking it as resolved, open, or spam.

See It in Action

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

U List all open conversations in my shared inbox.

I've retrieved 5 open conversations. Notable threads include 'Issue with pricing' (ID: cnv_123) and 'New feature request' (ID: cnv_456). Which one should I provide details for?

U Show me the message history for conversation 'cnv_123'.

Scanning cnv_123... I found 4 messages. The latest was an email from 'John Doe' asking about Enterprise discounts. Would you like the full content of the last message?

U Reply to conversation 'cnv_123' saying 'I will check that for you right now'.

Done! I've sent your reply to conversation cnv_123. The customer will see your message in their inbox immediately.

Frequently Asked Questions

01 **How does Front MCP help with cross-channel communications?**

The Front MCP treats email, SMS, chat, and social media as one continuous conversation stream. It uses tools like `list_active_channels` to unify them into a single view for your AI agent.

02 **Can I use Front MCP to update ticket statuses?**

Yes. You can tell the agent to change the state of any conversation using the `update_conversation_status` tool, marking tickets as open, closed, or spam immediately.

03 What if a customer mentions something in two different channels?

Your agent uses `search_conversations_by_query` to find all related threads across platforms. This ensures you capture every mention and keep the full context for your team.

04 Does Front MCP help with contact data?

Absolutely. Use the `get_contact_info` tool to retrieve verified contact details, preventing errors when sending replies or updating records.

05 Can I reply using the Front MCP?

Yes. The agent can send a direct message using `reply_to_conversation`, allowing you to communicate with the customer without leaving your AI client's window.

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-alternative": { "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 →

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	`019dd0f5-6e71-73b3-bc32-d74f16eac860`
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/front-alternative.