MCP SERVER | NO CODE | CLOUD HOSTED

Trengo MCP

Unify all customer conversations into one inbox.

Trengo MCP connects your customer service team to WhatsApp, email, chat, and social channels through one shared inbox. It lets your agent manage ticket lifecycles, check contact details, and send messages across every channel without switching apps.

A+ Quality Score 100/100

omnichannel-inbox | helpdesk-ticketing | shared-inbox | customer-service | live-chat | whatsapp-integration

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

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.

Trengo MCP

12 tools available
Cloud-hosted on Vinkius

Managing customer conversations used to mean logging into five different dashboards just to see a single issue's full history. This MCP changes that by giving you a unified view of all incoming communication—whether it arrived via email, WhatsApp, or live chat. You can use your agent to quickly list every ticket and pull up the entire message history for any contact in one place. Need to update something? Your agent handles status changes and internal notes automatically. It even lets you query team member lists so you know who's available right now. By connecting this MCP through Vinkius, you get a single point of control over your whole support operation, letting your AI client handle the heavy lifting for coordination, ticketing, and messaging.

Core Capabilities

01 — Manage all tickets

You can list every open or closed ticket and create new ones with a single command.

02 — Handle conversations

Fetch the full message history for any specific ticket, including internal notes from your team.

03 — Send messages

Reply to customers or add private notes to a conversation thread across different channels.

04 — Coordinate the team

List all available team members and check their current workload status.

05 — Manage contacts

Retrieve your entire customer database or verify that communication channels are set up correctly.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide the Trengo API Token, which you find in your account settings.

02 Connect your preferred AI client (like Cursor or Claude) to Vinkius. Your agent now has access to all the communication tools.

03 Ask your agent a natural language question, like 'List all open billing tickets' or 'Send a follow-up message to John Doe'.

The bottom line is that you talk to your AI client how you normally talk to a coworker; it does the rest of the work in Trengo.

Built For

Support Managers and Operations Leads who are tired of context switching between email, chat apps, and ticketing systems. This MCP is for people whose job depends on having a complete, real-time view of customer interactions across every possible channel.

Customer Success Manager

Responding to inquiries by pulling up ticket history from WhatsApp or email and updating the status as 'RESOLVED' without leaving the chat window.

Support Operations Lead

Coordinating communication channels, running reports on team availability using list_team_members, and setting up webhook events to track system activity.

Help Desk Manager

Quickly reviewing ticket histories across all platforms to monitor overall team performance and identify bottlenecks in the support process.

What Changes When You Connect

01 Stop context switching. Instead of jumping between email, WhatsApp web, and your ticketing system, you use a single chat interface to manage everything.

02 Instantly access full history using list_messages. When an agent takes over a ticket, they don't waste time searching; the entire conversation is immediately available.

03 Maintain accuracy when closing tickets. Use update_ticket to change statuses and add internal notes, ensuring compliance and a clear audit trail for every interaction.

04 Coordinate faster with your team. list_team_members lets you check who is available or who owns a specific account before assigning the ticket.

05 Never miss a communication event. The webhook tools allow you to monitor conversations in real-time, letting your agent trigger actions as soon as something happens.

Real-World Applications

Needing full context for an escalated issue

A customer reports a major outage via social media. Instead of manually searching three different systems, the agent asks their AI client to list_contacts and then use list_messages on that contact ID. The AI pulls up the last 5 messages from email, chat, and WhatsApp instantly, giving the team lead all the context needed for an accurate response.

Handling a backlog of old tickets

The Ops Lead needs to clean up old support requests. They ask their agent to list_tickets, filter by 'OPEN', and then use update_ticket on the results, setting them all to 'PENDING' status with a note explaining the follow-up required.

Sending a proactive follow-up

A ticket was resolved last week but nothing happened. The agent asks their AI client to send_message to that customer ID, reminding them of the resolution and asking if they need anything else. This is all done without needing the customer's direct email address.

Onboarding a new team member

A new support rep needs to know who is available right now. They ask their agent to list_team_members, and the AI client provides an immediate status report showing who is online and how many tickets they currently own.

Patterns to Avoid

Copy/pasting history manually

✗ AVOID

A user sees a conversation in email, copies the text, switches to chat, pastes it, then opens the ticketing system and pastes it there too. The result is lost context and wasted time.

✓ INSTEAD

Use your agent to list_messages on the ticket ID. This pulls all history—email, chat, social—into one view automatically. Then use send_message to reply directly from that single pane of glass.

Forgetting the status update

✗ AVOID

A conversation is resolved, but the user forgets to manually change the ticket status in the system, leaving it stuck as 'OPEN' and blocking reporting.

✓ INSTEAD

After resolving the issue, tell your agent to execute update_ticket. This automatically changes the status (e.g., to CLOSED) and ensures proper record-keeping.

Using multiple tokens/API keys

✗ AVOID

Trying to connect Trengo via one client for chat data, and then connecting a separate service just for email data.

✓ INSTEAD

Connect this MCP through Vinkius. It centralizes all your communication tools under one roof, letting your agent talk to every channel simultaneously.

The Right Fit

Use this MCP if the core problem is fragmented customer data. If you handle support across WhatsApp, email, and chat, but your current process requires switching between different dashboards or copying/pasting conversations, this is for you. You need a single source of truth. Don't use this if your only goal is pure internal knowledge management; in that case, an internal documentation tool might be better. Also, don't rely on it to *write* the responses—it manages the data and status updates. For drafting perfect copy, you

still need a dedicated writing assistant, but for coordinating the message delivery itself, this MCP is essential.

The headache of juggling multiple support channels

Right now, when an issue comes in through WhatsApp, your team has to switch apps. Then they go to the email client to see if the customer replied there too. They might open a dedicated chat widget just for live issues. This constant app jumping means lost context, slow response times, and manual data entry across five different tabs.

With this MCP, all those channels feed into one unified view. Your AI agent sees everything—WhatsApp messages, emails, chats—as one continuous conversation thread. The result is that your team stops chasing down threads and starts solving problems.

Manage conversations with the Trengo MCP

The tedious steps of checking contacts in a separate tab, manually logging into another service just to check status, or having to copy 50 lines of chat history are gone. Your agent handles that background coordination.

You simply tell your AI client what needs doing—'Close this ticket and notify the team lead.' It executes all the required steps automatically using update_ticket, list_team_members, and sending messages in a single flow.

Trengo MCP with 12 Tools

Use these tools to manage everything from listing all open support issues to sending direct messages and coordinating your entire team.

#	TOOL	DESCRIPTION
01	`create_ticket`	Opens a brand new support ticket within Trengo.
02	`create_webhook`	Sets up a new webhook to track specific conversation events in real time.
03	`get_account_profile`	Retrieves information about the current user's profile within Trengo.
04	`get_ticket`	Pulls up specific details for one existing support ticket.
05	`list_channels`	Lists all communication channels that are currently configured and active.
06	`list_contacts`	Retrieves a comprehensive list of all customers in your database.
07	`list_messages`	Shows the entire message history associated with a specific ticket.
08	`list_tickets`	Lists all current support tickets, allowing you to see which ones are open or closed.
09	`list_team_members`	Gets a list of all available team users and their current status.
10	`list_webhooks`	Lists the webhooks you have previously set up for event tracking.
11	`send_message`	Sends a direct message to a customer via any connected channel.
12	`update_ticket`	Changes the status of an existing ticket, like moving it from 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 all currently open support tickets.

I've retrieved your open tickets. You have 4 active conversations including 'Billing issue' from John Doe, 'Login error' from Smith, and 2 others. Which one would you like to see the messages for?

U Show me the last 3 messages for ticket #88231.

I've fetched the history. The last 3 messages include: a customer query about shipping, your team's response, and an internal note regarding the tracking number. Shall I reply to the customer?

U Close ticket #10293 as 'CLOSED' and add a note 'Resolved via AI'.

Ticket updated! I've set the status of ticket 10293 to 'CLOSED' and added your internal note. The conversation has been archived successfully.

Frequently Asked Questions

01 **How does Trengo MCP handle conversations?**

It gives you one unified inbox to view all incoming communications from WhatsApp, email, chat, and social media. This means your agent sees every message in the same place, regardless of where it originated.

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

Yes, you can use the update_ticket tool. Your agent changes the status (e.g., OPEN to CLOSED) and adds internal notes, keeping your records clean and accurate.

03 **What if I need to check who on my team is available?**

You can use list_team_members. Your agent queries this tool to give you a real-time report on which users are online or currently handling tickets, speeding up assignment.

04 **Does Trengo MCP let me send messages outside of a ticket?**

Absolutely. The send_message tool lets your agent communicate with customers directly across various configured channels, even if there isn't an active support ticket running.

05 **How do I get the full history for a complex issue using Trengo MCP?**

Use list_messages. This tool pulls the entire conversation transcript—including customer queries and internal team notes—for one specific ticket, giving you total context immediately.

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 `"trengo": { "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

Trengo 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 Trengo. 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	Trengo MCP
Server ID	`019dd179-8267-737c-a3d0-c2a4052b6672`
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/trengo.

© 2026 Vinkius · vinkius.com · support@vinkius.com