

MCP SERVER

NO CODE

CLOUD HOSTED

Vapi MCP

Turn Chat Context into Real-Time Voice Calls

Vapi brings automated voice communication into your existing chat workspace. Use this MCP to make outbound human-like phone calls, run web-based voice demos, and retrieve full call transcripts and metrics—all controlled by your AI agent. It lets you treat your chatbot like a real telephony dispatch center.

A+ Quality Score 100/100

voice-ai

telephony

conversational-ai

webrtc

call-automation

transcription



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.

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 — 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.

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.

Vapi MCP

10 tools available
Cloud-hosted on Vinkius

Your AI client can now handle more than just text; it can talk on the phone. With this MCP, your agent gains the ability to place outbound calls using various numbers and even set up web-based voice connections. You don't have to switch applications or copy call logs into a separate system anymore. Need to know what happened during a client demo? Your agent pulls the full transcript and performance metrics instantly. Want to adjust how your assistant behaves? You can update its core persona, change its underlying language model, or even build entirely new specialized assistants without leaving your chat window. Because Vinkius hosts this MCP, you connect once from any compatible client (Claude, Cursor, etc.) and get full access to advanced voice operations—all guided by natural conversation.

Core Capabilities

01 — Initiate Outbound Calls

Instruct your agent to call a specific phone number or generate a web link for a live voice interaction.

03 — Configure AI Personas

Create new voice assistants or modify existing ones by updating their core instructions, model type, and voice IDs on the fly.

02 — Access Call History and Metrics

Retrieve detailed records, full transcripts, and performance metrics from any past communication session.

04 — Audit Telephony Resources

List all connected phone numbers, available multi-agent teams (squads), and defined tools for advanced call routing.

One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to this MCP and enter your Vapi Private API Key into your AI client.
- 02 Use natural language prompts in your chat agent to initiate actions, such as asking the assistant to place a demo call or list current assistants.
- 03 Your AI client executes the voice operation through Vapi, providing you with real-time transcripts, detailed logs, and actionable data.

The bottom line is, your chat agent becomes the control center for all your automated phone and voice work.

Built For

Anyone dealing with high-volume client interactions who hates switching between CRM, telephony dashboards, and their AI editor. This MCP is for ops teams that need real-time visibility into call outcomes without manual data entry.

Customer Support Agent

Needs to pull the exact voice transcript from a difficult customer call immediately after it ends so they can summarize action items and update tickets.

Sales Engineer

Requires their agent to place an outbound demo call directly to a prospect's number right from their IDE, logging the outcome automatically.

Voice AI Developer

Spends time tweaking assistant personas, updating system prompts, or changing LLM models for new assistants without opening multiple development windows.

What Changes When You Connect

- 01 You instantly get full call transcripts by using 'get_call_details', eliminating the need to manually listen back to hours of recordings or copy/paste text.

-
- 02 Manage your entire voice AI setup—including creating new personas with 'create_assistant' and tweaking existing ones with 'update_assistant'—all from a single prompt in your editor.

 - 03 Streamline demos by using 'list_phone_numbers' to audit available lines, then initiating a live test call via 'create_phone_call', all without leaving your workspace.

 - 04 Quickly summarize complex interactions; after a customer calls, you can use the agent to pull the transcript and generate summary action items based on what happened.

 - 05 Audit entire conversation pathways by listing multi-agent groups with 'list_squads' or checking which tools are available via 'list_agent_tools'.

 - 06 Get immediate visibility into your operations using 'list_calls' to see a log of recent activity, helping you audit usage patterns quickly.
-

Real-World Applications

Post-Call Summarization for Support Tickets

A support agent receives an alert about a complex call. Instead of logging into the separate telephony system and manually transcribing minutes, they ask their agent to run 'get_call_details' on the specific ID. The agent instantly provides a summarized transcript detailing all agreed-upon next steps.

Scaling Voice AI Personalities

A development team needs to launch a new product support bot. Instead of writing boilerplate code, they use 'create_assistant' to define its model and voice parameters. Later, if the script needs tweaking, they just run 'update_assistant'.

Simulating Sales Demos

A sales engineer needs to test a new demo script before calling a client. They use 'list_phone_numbers' to pick an internal line, then prompt the agent to run 'create_web_call'. This lets them practice the flow and record metrics without risking real prospect data.

Troubleshooting Call Routing

An operations manager notices call failures across different departments. They use 'list_squads' to view all possible routing paths and then check which tools are attached using 'list_agent_tools' to find the bottleneck.

Patterns to Avoid

Using Vapi for simple data lookup

X AVOID

Trying to use Vapi just because it handles conversation. If all you need is a static list of users, using 'list_users' from another tool will be overkill and inefficient.

✓ INSTEAD

Keep this MCP reserved for tasks that require real-time voice communication or transcript access. Use tools like 'get_call_details' when the output must include metrics or spoken dialogue.

Manually updating assistant prompts

X AVOID

Writing new instructions into a separate web dashboard every time you change how your bot behaves, which is slow and error-prone.

✓ INSTEAD

Use the 'update_assistant' tool. This lets your agent modify the persona's system prompt or model type directly through a single chat command.

Forgetting to audit resources

X AVOID

Attempting to place a call when you don't know which phone numbers are active or if you need multi-step routing.

✓ INSTEAD

Always start by calling 'list_phone_numbers' and then checking the available paths using 'list_squads'. This ensures your agent has all necessary resources.

The Right Fit

Use this MCP if your core workflow involves automated voice communication, call logging, or real-time transcription. If you need to make a phone call programmatically, pull metrics from a completed talk, or dynamically adjust the behavior of an AI persona mid-workflow, this is what you want. Don't use it if your entire job revolves around purely structured data manipulation (like validating JSON schemas) or simple database reads; for that, look at tools focused on pure record management. If you only need to list assets, 'list_phone_numbers' is great, but remember its primary strength is the *action* and *transcript* following the action.

The Pain of Manually Following Up After a Call

Today, if you have an important client call, you end up with several steps: You listen to the recording or read the transcript. Then, you copy key action items into your CRM's notes section. Next, you might have to log back into a separate telephony system just to pull the final metrics, which takes five minutes of clicking through three different tabs.

With this MCP integrated, that entire process collapses into one chat command. Your agent handles the call details retrieval and provides structured transcripts right where you are working. You get actionable summaries instantly.

Get Call Context with the Vapi MCP

You no longer have to rely on remembering what was said or manually piecing together notes from multiple sources. The agent handles retrieving the full transcript and metrics using 'get_call_details', making context retrieval immediate.

It's not just about getting data; it's about turning a messy, multi-system operation into one simple prompt.

Vapi: 10 Tools for Call & Voice Operations

These tools let you programmatically handle every aspect of voice communication, from setting up new assistants to retrieving the final transcript of a completed call.

#	TOOL	DESCRIPTION
01	<code>create_assistant</code>	Builds a brand-new voice AI assistant persona using specified configuration details.
02	<code>create_phone_call</code>	Starts an outbound phone call immediately, requiring the target number and customer information.
03	<code>create_web_call</code>	Generates a dedicated web link for setting up a voice call, useful for internal testing or virtual demos.
04	<code>get_call_details</code>	Retrieves the full details, transcripts, and performance metrics associated with a specific call ID.
05	<code>list_assistants</code>	Lists all currently configured voice AI assistants available in your Vapi account.
06	<code>list_calls</code>	Shows a list of both active and recently completed voice calls managed by the system.
07	<code>list_phone_numbers</code>	Retrieves a comprehensive list of every phone number connected to your Vapi account.
08	<code>list_squads</code>	Lists all defined multi-agent teams or conversational pathways for complex routing.
09	<code>list_agent_tools</code>	Shows a list of every tool that your Vapi assistants are currently allowed to use.
10	<code>update_assistant</code>	Modifies the settings, prompts, or models for an existing voice AI assistant persona.

See It in Action

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

U List all our configured Voice assistants and their IDs.



You currently have 3 conversational Voice AI assistants configured in Vapi: 'Lead Qualification Bot' (ID: ba1-342x), 'Support Intake Matrix' (ID: cd3-99ab), and 'Internal Switchboard' (ID: 9fz-77qp). Do you want to see the system prompts behind any of these?

U Get the details and full transcript for call ID 'vapi1234'.



I've retrieved the call log for 'vapi1234'. This was an outbound call that lasted 2 minutes and 15 seconds.

Summary: The AI Assistant contacted John Doe about his recent warranty claim. John indicated he has already shipped the defective unit. The AI processed this information gracefully and closed the ticket conceptually.

Here is a snippet of the transcript:

User: 'Yeah I sent it out through FedEx yesterday.'

Assistant: 'Perfect, thank you John! I have updated your claim status to incoming...'

U Update assistant 'Bot Support' to change its `model.model` parameter to `gpt-4o-mini`.



Successfully updated the conversational model for 'Bot Support'. It has transitioned in Vapi engine logic and is now utilizing `gpt-4o-mini`. Its next active conversation will run purely on this adjusted backend structure.

Frequently Asked Questions

01 How do I check if Vapi supports my company phone numbers?

You use the 'list_phone_numbers' tool to pull a list of every number connected to your account for review. This ensures you know what resources are available before making a call.

02 Can I change my assistant's voice or model using Vapi MCP?

Yes, you use the 'update_assistant' tool. You can adjust settings like the underlying LLM model or mutate the Voice IDs on the fly without needing to access the main dashboard.

03 What is the difference between list_assistants and list_calls?

The 'list_assistants' tool shows you your conversational personas (the bots themselves). The 'list_calls' tool shows you logs of actual completed or active conversations.

04 How do I start a demo call with Vapi MCP?

You use the 'create_phone_call' tool. You just need to provide the target phone number ID and details about who is calling for a successful outbound connection.

05 Does Vapi MCP help me manage complex call flows?







Yes, you can use 'list_squads' to see all defined multi-agent teams. This helps structure advanced conversational paths and ensure the right bots talk to each other.

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 <code>"vapi": { "url": "..."} </code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

Vapi 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 Vapi. 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	Vapi MCP
Server ID	019d761b-1b9b-739d-a8fa-87ede84a83ae
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/vapi.