

MCP SERVER

NO CODE

CLOUD HOSTED

# Telnyx MCP

## Manage Messaging, Numbers, and Billing From Your Agent

Telnyx MCP lets your AI agent manage all aspects of your telecommunications infrastructure. Send and track SMS/MMS messages, check current account credits, list phone numbers, or configure messaging profiles—all from a single chat window.

**A+** Quality Score 100/100

sms-api

mms-messaging

voice-ai

carrier-infrastructure

message-delivery

telecom-automation



# 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 — 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](https://cloud.vinkius.com) — connect your AI agent in under 60 seconds.

# Telnyx MCP

10 tools available

Cloud-hosted on Vinkius

Need to handle high-volume communications without leaving your IDE? This MCP connects directly to your Telnyx account, letting you manage everything related to texting and calling using natural conversation. Instead of logging into a dashboard, you just ask your agent what you need done. You can instantly check if you have enough credits before sending out a major campaign. Need to send an MMS with a specific image? Just tell the agent who it goes to and where the media is hosted. Beyond messaging, the MCP helps you manage phone numbers by listing all owned E.164 numbers or creating new application-specific messaging profiles. Because this connection lives on Vinkius, your agent can access these powerful telecom tools alongside thousands of others, keeping your entire workflow consolidated and simple.

---

## Core Capabilities

### 01 — Send Text Messages

Dispatch standard SMS texts to any number using a specified phone number.

### 02 — Send Multimedia Messages (MMS)

Send rich MMS messages that include external media URLs, like images or videos.

### 03 — Check Account Status

Retrieve your exact Telnyx account credit balance to verify spending capacity.

### 04 — Inspect Message Delivery

Look up specific message details and confirm whether a text was delivered, failed, or rejected.

### 05 — Manage Phone Numbers

List all active E.164 phone numbers owned by your account, along with their unique IDs and configuration settings.

# One Click on Vinkius — From Prompt to Execution

Available at [vinkius.com/mcp/telnyx](https://vinkius.com/mcp/telnyx) — connect your AI agent in three steps.

- 01** Subscribe to this MCP on Vinkius and provide your private Telnyx API Key.
- 02** Your agent accesses the credentials, allowing it to query live account data like balances and number listings.
- 03** You issue a command or ask a question (e.g., 'Send an MMS to +1234...') and the MCP executes the required action.

The bottom line is you control complex telecom operations by talking to your AI client, instead of navigating multiple web portals.

---

## Built For

This connector targets communications engineers, marketing automation managers, and support staff who spend too much time switching between dashboards or running manual API calls. If you need reliable message delivery verification or managing large pools of phone numbers without leaving your coding environment, this is for you.

### Development Engineer

Debugging messaging webhooks or provisioning new application-specific messaging profiles directly within the IDE.

### Marketing Operations Manager

Verifying the account balance before triggering a massive, time-sensitive SMS broadcast campaign.

### Technical Support Specialist

Investigating specific delivery failures by retrieving precise message status details immediately upon receiving a ticket.

## What Changes When You Connect

- 01 Stop checking multiple dashboards. You can check your account balance using `get_account_balance` and immediately know if you have the credits needed for a big campaign.
- 02 Need to debug comms? Use `list_phone_numbers` to get all your owned numbers and then use `get_number_settings` to inspect any specific configuration details, all in one conversation.
- 03 Sending content is simple. You can dispatch standard SMS texts with `send_text_message` or send complex MMS messages using `send_media_message`, specifying the media URL.
- 04 The MCP lets you manage infrastructure setup by allowing you to create and delete messaging profiles via `create_messaging_profile` and `delete_messaging_profile` without needing a dedicated UI login.
- 05 Support teams save time investigating failures. They can use `get_message_details` to instantly pull up the delivery status of any message, confirming if it was rejected or delivered.

---

## Real-World Applications

### Verifying SMS Campaign Readiness

A Marketing Manager needs to send 50,000 texts. They first prompt the agent with 'What is my current account balance?' and receive confirmation of credits via `get_account_balance`. Once verified, they tell the agent to run a test batch using `send_text_message`, ensuring funds are sufficient before launching the full campaign.

### Onboarding a New Application Channel

A Development Engineer needs a dedicated messaging channel for a new app. They ask the agent to 'Create a messaging profile named X,' triggering `create_messaging_profile`. After creation, they use `get_messaging_profile_details` to confirm all parameters are set correctly.

### Investigating Lost Messages

A Support Tech receives a ticket about a message that never arrived. They ask the agent for details on the UUID, triggering `get_message_details`. The tool immediately confirms whether the text was rejected or if it delivered successfully.

### Auditing Number Assets

An Operations Lead needs to confirm every number owned by the company. They instruct the agent to 'List all E.164 numbers,' using `list_phone_numbers` to get a full, auditable inventory of IDs and active lines.

---

## Patterns to Avoid

---

### Assuming Message Status

#### X AVOID

A user sends an SMS via their dashboard but later assumes the message went through. They waste time checking other logs, unsure if it actually reached the recipient.

#### ✓ INSTEAD

Instead of guessing, ask your agent to use `get_message_details` with the message UUID immediately after sending a text or MMS. This guarantees you get the definitive delivery status.

### Mixing Up Profile Types

#### X AVOID

A developer tries to send an MMS but uses parameters designed for standard SMS, leading to failure because the media URL isn't recognized.

#### ✓ INSTEAD

Always use `send_media_message` when sending images or videos. If you need to change the profile used for sending, first run `list_messaging_profiles` and then update it using `get_messaging_profile_details`.

### Deleting a Profile by Guess

#### X AVOID

A user wants to clean up old profiles but manually deletes one without knowing its exact name or ID, potentially breaking live systems.

#### ✓ INSTEAD

Before deleting anything, always run `list_messaging_profiles` and review the output. Then, use `get_messaging_profile_details` on the specific profile before running `delete_messaging_profile`.

## The Right Fit

Use this MCP if your primary need is managing communication flow (SMS/MMS) or maintaining an inventory of phone numbers. You'll use it when you need to programmatically check account balance, send media-rich messages, or validate delivery status. Don't use this if you are only dealing with voice call routing (this MCP focuses

on text). Similarly, don't rely on this for complex billing reconciliation outside of checking the current credit balance; it's a messaging tool, not an accounting ledger. If your goal is simply to chat with contacts, use a dedicated CRM connector instead.

---

## Manually Checking Telecom Credentials Is a Time Sink.

Right now, every time you need to send a bulk message or check if an old text actually went through, you're doing clicks. You open the dashboard, navigate to 'Billing,' check the balance; then switch tabs to 'Numbers' to list IDs; and finally, you have to submit the message manually in the sending module. It's slow, error-prone, and involves jumping between 3 or 4 different web pages just to execute one simple task.

With this MCP, those manual steps vanish. You simply tell your agent what needs doing—'Send an MMS about Project X.' The agent handles the balance check, the number retrieval, and the message dispatch using a single conversation thread. It's instant confirmation, zero clicks.

---

## Sending Communications Gets Precise with Telnyx MCP

You no longer have to copy-paste numbers or manually verify credentials before running a campaign. The agent handles the number listing via `list_phone_numbers` and validates your funds using `get_account_balance` before you even hit send.

This means high confidence in every message dispatched. You get confirmation that the text was accepted for delivery, not just submitted—a huge difference when reliable communication matters.

---

# Telnyx MCP – 10 Tools Available

These tools allow you to manage your entire telecom stack, from listing phone number assets to sending complex multimedia messages.

#	TOOL	DESCRIPTION
01	<code>create_messaging_profile</code>	Creates and registers a new messaging profile within your Telnyx account for specific applications.
02	<code>delete_messaging_profile</code>	Permanently removes an existing messaging profile; this action cannot be undone.
03	<code>get_account_balance</code>	Retrieves the current, real-time credit balance remaining on your Telnyx account.
04	<code>get_message_details</code>	Looks up a specific message's history and verifies its final delivery status.
05	<code>get_number_settings</code>	Retrieves detailed configuration and capability information for one of your owned phone numbers.
06	<code>get_messaging_profile_details</code>	Fetches specific data related to a particular messaging profile you have created.
07	<code>list_phone_numbers</code>	Lists every E.164 phone number owned by the account and provides their unique IDs.
08	<code>list_messaging_profiles</code>	Retrieves a complete list of all messaging profiles currently configured in your account.
09	<code>send_text_message</code>	Sends an outbound SMS text message, requiring both the source and destination phone numbers.
10	<code>send_media_message</code>	Transmits an MMS message that includes a media URL (JPEG/PNG) along with the required send/receive numbers.

---

## See It in Action

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

### U What is my current Telnyx account balance?



Your current Telnyx account balance is \$245.50. You possess sufficient credits to continue your standard outbound messaging and voice operations.

### U List all the phone numbers I currently own and their IDs.



I retrieved 2 active numbers from your account:

- +14155552671 (ID: 15b14eb1-XXXX)

- +14155559982 (ID: 39a48be9-XXXX)

Both are fully functional. Would you like to check the detailed settings for one of them?

### U Send a text message from +14155552671 to +12125550999 saying that the server is down.



The SMS has been successfully submitted to Telnyx for delivery.

From: +14155552671

To: +12125550999

Message: 'The server is down.'

Message UUID: 8e5f22e8-xxxx. Would you like me to check its real-time delivery status?

---

## Frequently Asked Questions

### 01 How do I check my account balance using Telnyx MCP?

You prompt the agent to get your current credit balance by calling ``get_account_balance``. This instantly tells you how much budget you have left for messaging operations.

---

**02 Can Telnyx MCP send images? What tool do I use?**

Yes, it sends rich MMS messages. You must use the ``send_media_message`` tool and provide a publicly accessible URL pointing to your JPEG or PNG image.

---

**03 What if I need to check delivery status after sending a message?**

You can use ``get_message_details``. Just give it the unique message identifier, and it will tell you if the text was delivered, rejected, or is still pending.

---

**04 Does Telnyx MCP manage phone numbers only for sending messages?**

No. You can use ``list_phone_numbers`` to see all owned assets and ``get_number_settings`` to inspect detailed configuration settings for each number.

---

**05 How do I create a new messaging profile with Telnyx MCP?**

You initiate the process using ``create_messaging_profile``. This allows you to set up dedicated communication channels for your applications without logging into the Telnyx console.







---

# Go Live in 60 Seconds

Get your connection token from [cloud.vinkius.com](https://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
 <b>Claude AI</b>	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
 <b>Cursor</b>	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
 <b>VS Code</b>	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add <code>"teInyx": { "url": "..." }</code>
 <b>Windsurf</b>	MCP Settings → <code>mcp_settings.json</code> → Add endpoint URL
 <b>ChatGPT</b>	Settings → Tools & plugins → Add MCP server → Paste endpoint
 <b>Gemini</b>	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

# Telnyx 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](https://vinkius.com) · [support@vinkius.com](mailto: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 Telnyx. 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	Telnyx MCP
Server ID	019d7611-3c4b-72fd-a592-69266b3d1178
Platform	Vinkius Cloud for AI Agents
Endpoint	<a href="https://edge.vinkius.com/{token}/mcp">https://edge.vinkius.com/{token}/mcp</a>

### 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/telnyx](https://vinkius.com/mcp/telnyx).