

MCP SERVER

NO CODE

CLOUD HOSTED

# Telegram Bot MCP

Manage all messaging and media from your AI agent.

Telegram Bot MCP lets you automate everything that happens inside your Telegram chats and groups. Use natural language commands from any AI client to broadcast announcements, manage media, monitor group membership, or route customer service messages—all programmatically.

**A+** Quality Score 100/100

bot-api

instant-messaging

notifications

chat-automation

webhook

media-delivery



# 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.

# Telegram Bot MCP

13 tools available  
Cloud-hosted on Vinkius

Need to control a Telegram bot without opening the app? This MCP connects your existing bot directly to your agent, letting you automate complex communication tasks through simple conversation. You can send rich content—photos, documents, videos, and audio files—or just plain text formatted with Markdown or HTML. It also handles chat management; check who's in a group, find out how many members are online, or see the list of administrators. If you're building an AI workflow that needs to interact with people via Telegram, this MCP is what you need. Since Vinkius hosts thousands of connectors, integrating your bot into any agent flow—whether running on Claude, Cursor, or Windsurf—is straightforward and reliable.

---

## Core Capabilities

### 01 — Broadcast and Format Content

Send text messages with rich formatting (Markdown/HTML), photos, documents, videos, and audio to any chat or channel.

### 03 — Process Message History

Fetch recent updates from users or programmatically forward and delete specific messages within any group.

### 02 — Monitor Chat Status

Retrieve detailed information about a chat, including the total member count or a list of who the admins are.

# One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to this MCP on Vinkius and provide your unique Telegram Bot Token, which you get from BotFather.
- 02 Connect your preferred AI client—like Claude or Cursor—to the Vinkius catalog.
- 03 Use natural language commands in your agent. Your agent executes the necessary API calls to control your bot.

The bottom line is that once connected, you treat your Telegram bot like any other service tool, controlling it entirely through conversation.

---

## Built For

This MCP is for the Community Manager who dreads manually posting announcements across 10 different groups. It's also for Support Engineers who need to route customer chat histories between departments automatically, or DevOps teams building automated notification pipelines.

### Community Manager

Broadcast a single announcement message to dozens of channels and groups instantly, without logging into the Telegram app.

### Support Engineer

Automatically pull chat history details or forward critical customer messages from one department's group to another for review.

### DevOps Automation Specialist

Build alert systems that send detailed notifications, including attached documents and photos, when a specific system event occurs.

---

## What Changes When You Connect

- 01 Stop manual broadcasting. You can send announcements to entire channels and groups using the `send_message` tool, eliminating one-by-one posting.

- 
- 02** Get real-time insights into community size. Use `get_chat_members_count` to know exactly how many people are in a group without checking the app.
- 
- 03** Manage support conversations efficiently. You can retrieve chat history using `get_updates`, allowing your agent to build context before responding.
- 
- 04** Control message flow. Need to clean up a thread? Use `delete_message` or `forward_message` to manage content programmatically.
- 
- 05** Deliver complex media packages. Send everything from simple text with formatting via `send_message` all the way up to large documents using `send_document`.
- 

---

## Real-World Applications

### Onboarding New Users

A Community Manager needs to welcome new subscribers. They ask their agent, 'Send a warm welcome message and attach the PDF guide.' The agent then uses `send_document` and `send_message` to deliver the full kit automatically.

### Support Escalation

A support team member needs to hand off a chat. They ask the agent: 'Forward this entire conversation thread, including all media, to the Billing department.' The agent uses `forward_message` to keep the context intact.

### Incident Alerting

A DevOps engineer needs to alert a team about a system failure. Instead of writing multiple messages, they instruct their agent: 'Send an urgent alert photo and a diagnostic report document.' The agent uses `send_photo` and `send_document`.

### Auditing Group Membership

A moderator needs to check if certain users have been promoted. They ask their agent for 'the list of admin usernames in this group.' The agent executes `get_chat_admins` and provides the list.

---

# Patterns to Avoid

---

## Treating it like a simple broadcast tool

### X AVOID

Assuming you can only send plain text updates. If you try to attach an image or PDF, your agent will fail because the basic messaging tool isn't enough.

### ✓ INSTEAD

Always use specialized tools for media. Use ``send_photo`` when sending images, and use ``send_document`` whenever you need to transmit files like PDFs.

---

## Trying to view chat history via the API

### X AVOID

Asking the agent to 'show me everything that was said yesterday.' The bot only gets updates when they happen; it doesn't store a historical database.

### ✓ INSTEAD

Use ``get_updates`` to see what has changed since your last check, or rely on the chat client itself for viewing long-term message history.

---

## Manually checking group size

### X AVOID

Going into Telegram and clicking 'Members' just to know the count. This is slow, especially if you have many groups.

### ✓ INSTEAD

Use ``get_chat_members_count`` directly in your agent workflow for instant, reliable counts.

---

# The Right Fit

You should use this MCP if your core job function involves communicating or managing content *within* the Telegram platform itself. This includes broadcasting announcements, handling customer service chats, or archiving group history. However, don't use it if you need to interact with external systems; for example, if you need to update a user's record in a CRM database (like Salesforce) or pull inventory data from an eCommerce backend. For those tasks, look for MCPs designed specifically for those platforms, not messaging APIs.

---

## Managing chat communications is tedious and manual.

Today, sending a single announcement requires multiple steps: opening Telegram, navigating to the first group, pasting the message, hitting send. Then you repeat that entire process for every other channel or group in your roster until they're all done.

With this MCP, tell your agent what needs to be said once. Your AI client handles the connection and executes the broadcast across multiple channels automatically. You just get confirmation that the message was delivered.

---

## Send everything in one place with Telegram Bot MCP.

You don't have to switch between different tools or interfaces to manage media. Instead of separately handling text, then uploading a photo, and then attaching a document, you instruct your agent to 'send the full status update.'

Now, managing complex communications is one simple request. The system handles the sequencing, ensuring the message, photos, documents, and videos are all delivered as part of one cohesive action.

---

# Telegram Bot MCP with 13 Tools

These tools let you perform specific actions inside Telegram, such as sending rich media, retrieving member counts, or deleting old messages.

#	TOOL	DESCRIPTION
01	<code>delete_message</code>	Deletes a message from any Telegram chat.
02	<code>forward_message</code>	Forwards a specific message within a Telegram chat.
03	<code>get_chat_admins</code>	Retrieves the current list of administrators in a given chat.
04	<code>get_chat_members_count</code>	Returns the total number of members currently in a Telegram group or channel.
05	<code>get_chat_info</code>	Retrieves general details about a specific Telegram chat, including its type and ID.
06	<code>get_bot_info</code>	Fetches configuration information about your connected bot itself.
07	<code>get_updates</code>	Retrieves the latest activity or updates that have happened within the Telegram bot's scope.
08	<code>get_user_profile_photos</code>	Returns a list of available photo sizes for a user's profile pictures.
09	<code>send_audio</code>	Sends an audio file directly to a specified Telegram chat.
10	<code>send_document</code>	Transmits a document, like a PDF or ZIP file, into a Telegram chat.
11	<code>send_message</code>	Sends formatted text messages to a Telegram chat, supporting Markdown and HTML styling.
12	<code>send_photo</code>	Delivers an image photo to a Telegram chat; you can optionally include a caption with it.
13	<code>send_video</code>	Transmits a video file into a specified Telegram chat.

---

## See It in Action

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

### U Get information about my Telegram bot.



Your bot 'MyAssistantBot' (@my\_assistant\_bot) is active. Bot ID: 123456789. It's currently configured to handle messages and commands.

### U Send a message to my channel @mychannel saying 'Hello everyone!'



Message sent successfully to @mychannel. The message ID is 456 and was delivered at 2024-01-01 10:00:00 UTC.

### U How many members are in my group -1001234567890?



Your group 'Tech Community' currently has 1,247 members. It's a supergroup with public visibility.

---

## Frequently Asked Questions

### 01 How do I send formatted text using Telegram Bot MCP?

You use the `send_message` tool. This function supports both Markdown and HTML formatting, so you can make your announcements look professional right out of the gate.

### 02 Can I track who is in a chat using `get_chat_members_count`?

Yes, `get_chat_members_count` provides an accurate count of members. It's useful for knowing if your broadcast reached the expected audience size.

---

**03 What is the difference between `send_photo` and `send_document`?**

Use `send_photo` when you are delivering a visual image, which might include an optional caption. Use `send_document` for files like PDFs or ZIPs that need to be downloaded by the recipient.

---

**04 Does Telegram Bot MCP handle message replies?**

You can manage message history using tools like `forward_message`. This allows you to pass along context from one user to another, which is essential for support roles.







---

# 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>"telegram-bot": { "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

# Telegram Bot 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 Telegram Bot. 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	Telegram Bot MCP
Server ID	019d8488-bf86-71f6-82a2-5961a1749fd3
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/telegram-bot](https://vinkius.com/mcp/telegram-bot).