

MCP SERVER

NO CODE

CLOUD HOSTED

# Messenger MCP

Audit chats, manage personas, and reply instantly.

The Messenger MCP connects your AI client directly to your Facebook Page, giving you full control over conversations without ever touching the Meta dashboard. Audit message history, manage brand personas, and send direct replies instantly from your workflow.

**A+** Quality Score 100/100

instant-messaging

customer-engagement

social-platform

chat-automation

conversation-management

persona-management



# 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.

### 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.

# Messenger MCP

8 tools available

Cloud-hosted on Vinkius

Managing a high-volume Facebook Page used to mean navigating complex dashboards just to check if someone needed help or what messages were being sent out. This MCP changes that. It lets your agent act as a real-time communications assistant right inside your existing workflow. You can list all active conversations and pull up detailed message histories with timestamps—all without logging into Meta. Need to respond? Your agent sends direct text replies instantly, keeping your response rate high. Furthermore, you don't just manage chats; you keep brand consistency by listing and retrieving persona details and ensuring the correct media assets are available for responses. When you connect this MCP through Vinkius, your AI client handles all that messy operational work, turning enterprise messaging into a natural conversation.

---

## Core Capabilities

### 01 — Audit Conversation Threads

List active conversations and retrieve detailed message history for any specific thread.

### 02 — Automate Customer Replies

Send direct text replies to users instantly, ensuring rapid response times.

### 03 — Manage Brand Identity

List and pull information on brand personas and message creatives to maintain consistent messaging.

# One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to this MCP, providing your Facebook Page Access Token and Page ID.
- 02 Connect the credentials to your preferred AI client (like Claude or Cursor).
- 03 Ask your agent to perform tasks like listing conversations or sending a reply; it executes the action via the connected tools.

The bottom line is, you tell your agent what needs doing—auditing chats, updating personas, or replying—and it does it using the data and functions of your Facebook Page.

---

## Built For

This MCP is for operations teams who are tired of manually logging into multiple dashboards just to manage customer interactions. It's built for anyone whose job requires monitoring high-volume, fast-moving social chat threads.

### Social Media Manager

Performs rapid audits of conversation threads and verifies that new brand personas are correctly configured before a campaign launches.

### Customer Success Manager

Monitors user inquiries across all active chats, retrieving message history and sending quick updates straight from their workflow without switching apps.

### Marketing Lead

Verifies that the correct messaging identity is applied by listing brand personas and checking available media assets for automated responses.

---

## What Changes When You Connect

- 01 Stop switching between dashboards. Use `list_conversations` to get a summary of all active threads in one go, allowing you to triage issues immediately without manual navigation.

- 
- 02 Maintain high customer satisfaction by using `send_message` to deliver direct replies the moment an inquiry comes in, keeping conversations flowing even outside business hours.

---

  - 03 Keep your brand voice consistent. By listing and retrieving persona data via `get_persona_info`, you ensure every automated reply matches your established identity.

---

  - 04 Deeply understand user issues. The `get_messages` tool pulls full conversation histories for auditing, letting agents answer complex questions by reviewing the entire thread context.

---

  - 05 Control your page's structure. You can use `get_page_settings` and `get_page_info` to verify that necessary operational settings are maintained correctly.
- 

---

## Real-World Applications

### Handling a sudden PR crisis

A social media manager needs to track public sentiment immediately. They ask their agent to run `list_conversations`, quickly identifying all threads mentioning 'outage' or 'delay'. The agent then uses `get_messages` on the top five threads to pull full context for human review.

### Responding after a campaign launch

A marketing lead needs to send out follow-up materials. They check `list_message_creative` to confirm the latest image is available, and then use `send_message` to push the updated content directly into all relevant threads.

### Onboarding a new team member

A training manager needs to verify which brand voice is active. They ask their agent to execute `list_personas` and then use `get_persona_info` on the correct profile, ensuring all staff know the required messaging guidelines.

### Reviewing old customer complaints

An operations manager needs context on a ticket from six months ago. They tell their agent to use `get_messages` for that specific conversation ID, instantly retrieving the full log so they can update the internal record accurately.

---

# Patterns to Avoid

---

## Trying to manually copy thread details

### ✗ AVOID

A user has to open 15 separate chat windows on the Meta web interface and copy chunks of text, losing context and spending hours doing it.

### ✓ INSTEAD

Instead, ask your agent to `list_conversations` first. Then, for each key interaction, use `get_messages` to pull a clean, structured log directly into your workflow.

---

## Assuming the brand voice is current

### ✗ AVOID

A team member drafts a response using outdated terminology or an old product name because they aren't sure of the latest guidelines.

### ✓ INSTEAD

First, run `list_personas` to see all available identities. Then, use `get_persona_info` to confirm which persona rules must guide your `send_message` reply.

---

## Not knowing what settings are required

### ✗ AVOID

A new manager assumes the Page is ready for a sale because they just see messages coming in, but forgets that crucial tracking or opt-in settings need to be enabled.

### ✓ INSTEAD

Run `get_page_settings` first. This checks all necessary operational details and lets you verify that your brand presence adheres to internal standards.

---

## The Right Fit

Use this MCP if your primary bottleneck is accessing, auditing, or acting upon conversational data housed in Facebook Messenger. If you need to answer questions like 'What did the customer say three days ago?' (use `get_messages`), or 'How do I send a reply right now?' (use `send_message`). You also need it if maintaining brand consistency across multiple agents is critical; check `list_personas` and `get_persona_info`. However, don't use this if you only need to pull raw analytics—like total unique users over the last quarter or general demographic data that isn't tied to a specific chat thread. For broad, non-conversational data reports, look for dedicated business intelligence connectors instead.

---

---

## The time sink of logging into Meta to check chats

Today, managing customer engagement means opening the Facebook Page dashboard and wading through endless threads. You're clicking between conversation lists, searching for specific IDs, and then having to manually open each thread just to copy the message history you need for a report or a follow-up.

With this MCP, your agent handles the grunt work. Instead of manual clicks and tab switching, you simply instruct your AI client to `list_conversations`, giving you an immediate overview. You get clean data streams and instant actions without ever touching the original dashboard.

---

## Get Persona Oversight with Messenger MCP

Before every automated message, teams currently have to stop what they're doing and check a separate wiki or document to confirm the correct brand tone, available assets, and persona rules. This causes delays and leads to inconsistent messaging.

Now, you just ask your agent about it. You run `list_personas` to see all options, then use `get_persona_info` to verify the guidelines right before sending a message. Your bot speaks with authority, every time.

---

# Messenger MCP with 8 Tools

Use these eight tools to perform every operational task needed for managing a professional Facebook Page through conversational AI.

#	TOOL	DESCRIPTION
01	<code>get_messages</code>	Pulls the full message history for a single conversation thread.
02	<code>get_page_info</code>	Retrieves basic information about your connected Facebook Page.
03	<code>get_page_settings</code>	Gets specific operational settings for the Facebook Page itself.
04	<code>get_persona_info</code>	Fetches detailed information about a specific brand persona or bot identity.
05	<code>list_conversations</code>	Lists all recent, active conversations currently happening on the Page.
06	<code>list_message_creative</code>	Retrieves a list of media assets and creatives available for automated messaging.
07	<code>list_personas</code>	Lists every registered brand persona associated with the Page account.
08	<code>send_message</code>	Sends a direct text message reply to any user or recipient.

---

## See It in Action

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

**U** List all active Messenger conversations for my Page.



I've retrieved your conversations. You have 5 recent threads, including interactions with 'John Doe' and 'Jane Smith'. Which conversation ID would you like to audit for history?

**U** Send 'Thank you for contacting us!' to recipient ID 12345678.



Message sent! I've successfully delivered your reply to recipient 12345678. The response is now visible in the Messenger thread.

**U** Show me the message history for conversation t\_xxxx.



I've retrieved the history for t\_xxxx. There are 10 messages in this thread. The latest was 'Hello, I have a question about my order' from the user. Would you like the full log?

---

## Frequently Asked Questions

### 01 How do I audit old chat history using Messenger MCP?

You can retrieve detailed message history for any specific conversation ID by calling the `get_messages` tool. This pulls a full log so you know exactly what was said and when.

### 02 Does Messenger MCP help me update brand personas?

It helps you manage and verify them, not update their core definition. You can use `list_personas` to see all available profiles, and `get_persona_info` to check the current details.

---

**03 Can I send replies using Messenger MCP from my IDE?**

Yes, your agent uses the `send_message` tool to deliver direct text replies. This keeps your response rapid and conversational, even when you're working inside VS Code or Cursor.

---

**04 What if I need to check page operational settings? Use Messenger MCP.**

You can use `get_page_settings` to monitor the Page's core configuration. This is useful for operations managers who need to ensure compliance and proper setup.

---

**05 Is `list_conversations` limited to only recent chats?**

It lists active conversations, giving you a clear snapshot of current activity without needing to dig through massive archives manually.







---

# 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>"messenger": { "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

# Messenger 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 Messenger. 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	Messenger MCP
Server ID	019d8458-3523-7080-9f2c-7d0b909b9ecb
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/messenger](https://vinkius.com/mcp/messenger).