MCP SERVER | NO CODE | CLOUD HOSTED

ManyChat MCP

Manage Tags, Flows, and Subscriber Data Instantly

ManyChat MCP automates messenger marketing directly from your AI agent. It lets you manage subscriber profiles, apply tags for segmentation, and trigger entire conversation flows without leaving your client environment. Find users by name or custom fields, update their status, and control every touchpoint in your chat automation.

A+ Quality Score 100/100

messenger-marketing | chatbot | subscriber-segmentation | flow-automation | lead-nurturing | omnichannel-messaging

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

SCAN TO VIEW

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.

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.

ManyChat MCP

11 tools available
Cloud-hosted on Vinkius

Managing a large list of contacts across different messaging channels usually means switching between five tabs and copying data back and forth. This MCP changes that. Connect it to any AI agent, and you gain full command over your messenger marketing flows through natural conversation.

You can instantly find specific subscribers—whether by their name or a custom field value. Once located, your agent handles the rest. You can read all their existing tags, add new ones (like 'High Value Lead'), remove old ones, and even set up custom fields if needed. Need to send them a welcome sequence? You just tell your agent which flow to use; it sends the message immediately.

This level of control means you don't have to leave your workspace. Instead, you manage subscriber data, tags, and flows right from where you are working, making complex segmentation tasks simple conversational commands. Because this MCP is part of Vinkius, you get access to a massive catalog of tools, ensuring that if the automation task exists, your agent can find it.

Core Capabilities

01 — Discovering Users

Locate specific subscribers by searching their name or querying unique values from custom fields.

02 — Segmenting Audiences

Add, remove, or check all existing tags on a subscriber to accurately group them for targeted campaigns.

03 — Updating Profiles

Set specific values in custom fields on a user's profile after gathering new data points.

04 — Controlling Conversational Flows

Send pre-built automated conversation flows to single subscribers or list which sequences are available.

05 — Retrieving User Data

Fetch detailed information, all associated tags, and existing flow assignments for any given subscriber ID.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to the ManyChat MCP on Vinkius and provide your API Token.

02 Next, your AI client calls a specific tool function (like finding users or adding tags) using natural language prompts.

03 The MCP executes the command against your ManyChat account and sends the resulting data back to your agent for immediate use.

The bottom line is that you control complex marketing automation steps entirely through conversation, without ever needing to manually navigate the ManyChat dashboard.

Built For

Marketing Operations managers who spend too much time switching between chat platforms and spreadsheets. It's for digital marketers and lead nurturing specialists whose day involves constant segmentation, flow adjustments, and profile updates across multiple channels.

Digital Marketing Manager

Uses the MCP to identify specific groups of users by querying custom fields and then triggers a targeted sequence using `send_flow`.

Marketing Operations Specialist

Manages large-scale segmentation tasks, such as running `add_tag` or `remove_tag` across hundreds of contacts in bulk via AI prompts.

Customer Success Manager

Looks up detailed user history and status using `get_subscriber_info`, ensuring the support team knows exactly what conversations a user has had previously.

What Changes When You Connect

01 Instead of manually searching for users in the ManyChat dashboard, you simply ask your agent to `find_subscriber_by_name` or by a custom field value. The data comes back instantly for immediate action.

02 Segmenting becomes trivial. You can tell your agent to `add_tag` (e.g., 'Webinar Attendee') or `remove_tag` based on conversation history, keeping your campaigns highly accurate.

03 When a lead qualifies, you don't copy/paste details. Your agent can execute the sequence: `set_custom_field` first, then run `add_tag`, and finally send the appropriate welcome flow using `send_flow`.

04 Never wonder what a user is connected to again. Use `get_subscriber_tags` or `get_subscriber_flows` to pull all necessary context into your conversation history in seconds.

05 The MCP lets you run complex logic—like checking if a user exists (`get_subscriber_info`) before trying to update them, eliminating manual double-checks and API errors.

Real-World Applications

A lead needs follow-up after an event.

The sales team knows a specific contact attended the October webinar. Instead of finding them manually, they prompt their agent: 'Find users who have the custom field 'Webinar Attended' set to TRUE.' The agent uses `find_subscriber_by_custom_field` and then runs `add_tag` on all results, followed by sending a specific follow-up flow.

A user profile needs immediate correction.

Customer support gets a call from 'John Smith' who was miscategorized. The agent uses `find_subscriber_by_name` to locate the ID, then executes `set_custom_field` to fix their status and runs `remove_tag` for the old incorrect tag.

Running a complex qualification sequence.

A user interacts with a bot. The agent first checks if the user exists using `get_subscriber_info`. If they do, it executes `add_tag` and then runs `send_flow` to kick off the next stage of their nurturing journey.

Need a list of all current campaign groups.

The marketing manager needs to audit what tags are available. They simply ask the agent to 'List all usable tags.' The tool responds by running `list_tags`, giving them the full inventory at once.

Patterns to Avoid

Trying to manage segmentation outside of chat.

✗ AVOID

Opening ManyChat, going into the 'Tags' section, clicking a user, and manually selecting tags or flows. This is slow, repetitive, and requires constant context switching.

✓ INSTEAD

Keep the process within your agent. Use `find_subscriber_by_name` to locate the user ID, then run the commands: `add_tag` followed immediately by `send_flow`. It's all one conversation.

Assuming a user needs different data types.

✗ AVOID

Trying to manually update a field when you aren't sure of the available options, leading to API errors or blank records.

✓ INSTEAD

First run `list_custom_fields` to see exactly what fields are available. Then use `set_custom_field` with the correct key and value.

Re-running flow assignments unnecessarily.

✗ AVOID

Sending a welcome message when you aren't sure if the user needs it, or sending multiple messages back-to-back manually. This looks spammy.

✓ INSTEAD

Always use `get_subscriber_flows` first to confirm which flows are assigned, then use `send_flow` only when necessary for a specific campaign trigger.

The Right Fit

Use this MCP if your marketing process relies on highly contextual, real-time updates to user profiles and immediate flow triggers. If you need to segment users based on their name, activity, or any piece of data stored in custom fields, this is for you. It's best when the decision point—'Does this person qualify for Flow X?'—needs to happen immediately after a human interaction.

Don't use this if your core need is complex billing management,

detailed analytics dashboards, or syncing with external financial systems outside of messaging context. For those needs, you'll need specialized accounting or data warehouse connectors (like those found in the broader Vinkius catalog). If you just need to *read* large datasets but never write back changes, a general database query tool might be better than using `get_subscriber_info` repeatedly.

The Problem with Context Switching

Today's workflow forces you into a cycle of tabs. You receive a lead, so you jump to the messaging platform dashboard. You search for them by name or custom field value. Then, you check their tags and history. Next, you open a separate tab just to manually add a tag and then another one to send the correct flow sequence. It’s slow, error-prone, and breaks your focus.

With this MCP connected through Vinkius, that entire multi-step process collapses into simple conversation. You ask your agent to find the user by name or custom field value, check their tags, update them with a new custom field value, and finally send the appropriate flow—all in one seamless prompt.

Controlling Messenger Marketing with ManyChat

The manual steps that vanish include searching for user data across multiple tabs, clicking through tag lists to verify segmentation rules, and manually copying IDs or flow names into different interfaces. These micro-tasks kill momentum.

Now, you simply state the action: 'Add the VIP tag and send the premium onboarding sequence.' The MCP handles the lookups, updates, and message triggers automatically. You get back high-speed, verifiable automation.

ManyChat MCP with 11 Tools

These tools allow you to control every aspect of messenger marketing automation: finding users, updating profiles, segmenting audiences, and sending conversations.

#	TOOL	DESCRIPTION
01	add_tag	Applies a specific label to an existing subscriber profile.
02	find_subscriber_by_custom_field	Searches for subscribers whose profiles match a specified custom field value.
03	find_subscriber_by_name	Retrieves the profile of a subscriber by knowing their full name.
04	get_subscriber_flows	Lists all automated conversation flows that are assigned to a particular user.
05	get_subscriber_info	Retrieves the core data and status details for any subscriber using their unique ID.
06	get_subscriber_tags	Pulls a list of all tags currently assigned to a specific user's account.
07	list_custom_fields	Shows every custom data field that can be added or queried on your messaging page.
08	list_tags	Provides a complete list of all tags currently in use across the entire account.
09	remove_tag	Removes an existing label from a subscriber's profile when they change status or segment.
10	send_flow	Immediately initiates and sends a specific, pre-written conversation flow to the target user.
11	set_custom_field	Updates or creates a custom data field value for a subscriber's record.

See It in Action

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

U Find subscriber info for ID 12345678.

I've retrieved the info for subscriber 12345678. Name: John Doe, Gender: male, Status: active.

U Add the 'VIP' tag to subscriber 12345678.

The 'VIP' tag (ID: 987) has been successfully added to subscriber 12345678.

U List all tags on my ManyChat page.

Here are the tags for your page: 'VIP', 'New Lead', 'Follow-up'.

Frequently Asked Questions

01 **How do I find a user using ManyChat with this MCP?**

You can use `find_subscriber_by_name` if you know the person's name, or `find_subscriber_by_custom_field` if they have specific data like an account ID.

02 **Can ManyChat MCP update a user's status?**

Yes. You can use `set_custom_field` to change custom field values, which often dictates the user's active status or qualification level.

03 Is sending a flow permanent after I run it?

The MCP allows you to initiate the message using `send_flow`. The conversation history and the assigned flows are always visible via tools like `get_subscriber_flows` for auditing.

04 What is the difference between getting info and listing tags?

`get_subscriber_info` pulls all core data (name, status, etc.) by ID. Meanwhile, `get_subscriber_tags` only gives you a list of every tag attached to that user.

05 Does ManyChat MCP handle bulk tagging?

Yes, if your agent can retrieve multiple subscriber IDs (for example, by listing tags first), it can then execute `add_tag` on those groups of users sequentially.

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 `"manychat": { "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

ManyChat 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 ManyChat. 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	ManyChat MCP
Server ID	`019d75ce-a9d5-720a-aefe-40fb645682fe`
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/manychat.