

MCP SERVER

NO CODE

CLOUD HOSTED

2Chat MCP for AI Agents

Automating high-volume group message sending and verification

2Chat connects AI agents directly to WhatsApp's messaging infrastructure. It gives your agent the ability to send messages, manage groups, and verify phone numbers across multiple accounts using a single gateway. This MCP lets you automate complex communication workflows, running campaigns or coordinating internal team alerts entirely through natural language prompts.

A+ Quality Score 100/100

whatsapp-automation

multi-device

group-management

phone-verification

messaging-gateway

customer-service



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 — connect your AI agent in under 60 seconds.

2Chat MCP

5 tools available

Cloud-hosted on Vinkius

Using 2Chat, your AI client gains full control over WhatsApp messaging without needing manual intervention. You can write simple instructions—like 'Send the quarterly report to Project Alpha'—and have your agent handle the entire process: verifying the recipients, creating a new group if needed, and sending multi-media messages instantly. This eliminates logging into separate apps or manually copying numbers dozens of times.

It moves beyond basic messaging. You can build complex operational workflows that monitor incoming chat status, manage participation lists, and even handle communications across multiple linked WhatsApp accounts simultaneously. Because this MCP is hosted on Vinkius, you connect your preferred AI client once to access 2Chat alongside thousands of other tools for specialized tasks. It gives your agent the 'voice' it needs for modern communication, whether you're running large-scale marketing campaigns or just coordinating a quick internal team alert.

Core Capabilities

01 — Send structured messages

The AI can send text, images, PDFs, and voice notes to single numbers or entire groups.

03 — Verify number existence

Before sending anything, the agent checks if a given phone number is actually registered on WhatsApp to prevent failed deliveries.

05 — Monitor message status

It provides real-time updates on messages, letting you track if a message was sent, delivered, or read.

02 — Manage recipient lists and accounts

You can list all connected WhatsApp phone numbers to determine which account should be used for sending messages.

04 — Group creation and management

The MCP can create new groups with specific participants or list all existing groups linked to your accounts.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/mcp/2chat — connect your AI agent in three steps.

- 01 Subscribe to the 2Chat MCP and input your unique API key into your AI client.
- 02 Your agent uses natural language prompts (e.g., 'Send this announcement to the Dev Team group').
- 03 The MCP executes the necessary sequence of actions, handles verification, and sends the message via WhatsApp.

The bottom line is that your AI client can execute sophisticated communication tasks on WhatsApp using simple text commands, without needing a human in the loop for every step.

Built For

This MCP is for anyone whose job involves mass communication or complex group coordination via WhatsApp. Think community managers running large groups or development teams building automated notification systems. If you spend time copying and pasting messages across multiple channels, this tool saves you hours.

Community Manager

You use 2Chat to manage announcements for thousands of people in large WhatsApp groups, ensuring every message reaches the right audience without manual effort.

Marketing Specialist

You run automated campaigns by having your agent check numbers and send tailored messages or media files to segmented customer lists.

Operations Manager

Your team uses 2Chat to automate internal alerts, like shift changes or system outages, sending instant notifications to dedicated WhatsApp group channels.

What Changes When You Connect

- 01 Stops failed sends. The `check_number` tool verifies recipients before any message goes out, dramatically improving delivery success rates.

-
- 02 Manages scale effortlessly. You can use 2Chat to create new groups or list existing ones, allowing your agent to manage communication across dozens of teams.

 - 03 Full media support. Sending isn't limited to text. The `send_message` tool handles images and PDFs, making rich content campaigns easier than ever.

 - 04 Centralized control. By listing all connected numbers using `list_numbers`, you ensure your agent always sends messages from the correct source account.

 - 05 Group coordination simplified. Creating new groups or adding members via these tools means you never have to manually manage participant lists again.
-

Real-World Applications

Rolling out internal policy updates

The Ops Manager needs to notify 15 different department leads. They prompt their agent: 'Send the new security protocol PDF to all relevant groups.' The agent first uses `list_groups` to find the right channels, then sends the secure file via `send_message`, ensuring everyone gets the required update.

Onboarding a project team

The Community Manager starts a new initiative. They tell their agent: 'Create a group called Project Phoenix and add these six members.' The agent uses the `create_group` tool, immediately setting up the communication hub for the whole team.

Running a large-scale marketing funnel

A Marketing Specialist wants to hit 50 new leads. They prompt their agent: 'Check these numbers and send a personalized intro message.' The agent uses `check_number` on each lead first, then executes the targeted message via `send_message`, skipping any invalid contacts.

Consolidating departmental chats

The Development Lead needs to ensure all project stakeholders are in one place. They ask their agent to list groups and then use `create_group` to build a centralized, permanent communication channel for the entire development team.

Patterns to Avoid

Ignoring number validation

✗ AVOID

The agent attempts to send an announcement using only hardcoded numbers in a script. If even one number is inactive or incorrectly formatted, the whole batch fails and requires manual debugging.

✓ INSTEAD

Always run `check_number` on your target audience list before sending anything. This guarantees delivery success and keeps your communication workflows running smoothly.

Over-relying on single accounts

✗ AVOID

The team only uses one WhatsApp number for all communications, leading to account saturation or inability to segment messages by department.

✓ INSTEAD

Use `list_numbers` first. This lets you confirm which multiple numbers are connected, allowing your agent to send specific alerts from the correct departmental source.

Forgetting group context

✗ AVOID

The agent sends a general announcement without knowing which groups the user is actually part of.

✓ INSTEAD

Start by using `list_groups`. This pulls up all available channels, giving you visibility into every community or team your account can reach.

The Right Fit

Use 2Chat if your core problem involves automated, multi-platform messaging on WhatsApp. You need this MCP when the process requires sequenced actions: first verifying a list of contacts using `check_number`, then creating necessary groups via `create_group` or listing them with `list_groups`, and finally sending rich messages using `send_message`. Don't use it if you only need to read chat history—you'll want an MCP designed for data retrieval. If your goal is simple one-off messaging without group management, a basic messenger API might suffice, but 2Chat gives you the full orchestration layer that modern operations require.

Automating WhatsApp Group Announcements with 2Chat

Currently, running company-wide announcements on WhatsApp means manually checking every contact list. You copy a message into a new group chat, then repeat that process for the 'Sales' group, the 'Ops' group, and so on. It's tedious clicking, lots of chance to miss a group, and you lose time verifying if recipients are still active.

With 2Chat, your agent handles all this complexity. You simply tell it: 'Send the annual meeting schedule PDF to all department groups.' The MCP uses `list_groups` to find every relevant channel, then sends the file via `send_message`, automatically managing the whole rollout.

Managing WhatsApp Contact Lists with 2Chat

Before connecting this MCP, verifying a list of contacts was a nightmare. You'd have to manually test numbers one by one in the app just to see if they were active or if someone had changed their number. If you got hundreds of leads, this process stopped dead.

Now, 2Chat fixes that upfront. Your agent runs `check_number` across your entire list instantly. You get a clean, verified roster ready for any campaign —no guesswork and no failed sends.

5 Tools for WhatsApp Messaging & Group Management

Use these five functions to programmatically list contacts, create groups, send messages, and check number validity within your automated workflows.

#	TOOL	DESCRIPTION
01	<code>list_numbers</code>	Retrieves a list of all WhatsApp phone numbers currently connected and authorized under your 2Chat account.
02	<code>send_message</code>	Sends text or media content to specific individuals or designated groups within WhatsApp.
03	<code>check_number</code>	Verifies if a given phone number is active and registered on the WhatsApp platform, preventing delivery failures.
04	<code>create_group</code>	Creates an entirely new WhatsApp group and allows you to add specified participants at the same time.
05	<code>list_groups</code>	Fetches a list of all existing WhatsApp groups that your connected numbers are members of.

See It in Action

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

U I need to send the Q3 results PDF announcement to all team leads.



Success. The announcement has been sent successfully across 4 groups: 'Leadership Team', 'Finance Leads', and 'Ops Directors'.

Details:

- File attached: Q3_Results.pdf (12MB)
- Status: Delivered to all 4 groups.
- Next steps: I've also logged the group UUIDs for easy reference later.

U Check if the main sales numbers are connected and list all groups I belong to.



I've checked your connections. You currently have 3 active WhatsApp accounts linked.

Connected Numbers:

- +1 (555) 123-4567 (Primary)
- +1 (555) 987-6543 (Secondary)

Groups You Are In:

- Project Alpha Core Team
- Marketing Updates Q4
- Executive Leadership

U I'm starting a new initiative. Create a group called 'Project Phoenix' and add +1234567890.



Group created successfully! 🎉

New Group Details:

- Name: Project Phoenix
- Members added: 1 (+1234567890)
- Status: Ready for communication.

You can now send messages to this group using the 'send message' function.

Frequently Asked Questions

01 How can I use 2Chat to automate sending WhatsApp announcements?

You tell your agent the message content and the target group name. The MCP handles everything: it verifies the groups, sends the announcement via `send_message`, and manages the delivery status for you.

02 Does 2Chat help me manage multiple WhatsApp accounts?

Yes, it's built for multi-device support. You connect several numbers, and your agent can use all of them to send messages or run campaigns, preventing reliance on just one account.

03 What if I have a list of new contacts? Can 2Chat handle verification?

Absolutely. Before you send anything, the MCP uses `check_number` to verify every single number in your list. This ensures that failed deliveries due to invalid contacts are eliminated before they happen.

04 Can I set up a new team chat group using 2Chat?

Yes, you can use the `create_group` tool. You tell your agent the desired name and all the participants' numbers, and it builds the group instantly.

05 Is 2Chat better than just sending messages manually through WhatsApp?

It is far more powerful because you don't have to click or copy anything. You write a command in natural language (e.g., 'Send the report'), and your agent executes complex, multi-step actions across groups.

06 Does 2Chat support sending different types of media?







Yes, it handles more than just text. You can send images, PDFs, and voice messages through the system using a single command.

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>"2chat": { "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

2Chat 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 2Chat. 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	2Chat MCP
Server ID	019d7544-cb06-712f-ad1e-568dc4391d54
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/2chat.