

MCP SERVER

NO CODE

CLOUD HOSTED

# Gupshup Messaging MCP

Automate WhatsApp communications with your AI agent.

Gupshup Messaging MCP connects your AI agent directly to WhatsApp Business APIs. You can automate customer communications by sending free-form text, pre-approved template messages, and rich media assets like images or documents. It also lets you send interactive buttons and react to incoming messages with emojis, all without leaving your chat client.

**A+** Quality Score 100/100

whatsapp-api

conversational-messaging

template-messages

interactive-buttons

omnichannel



# 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.

# Gupshup Messaging MCP

6 tools available

Cloud-hosted on Vinkius

Your AI agent manages complex communication workflows on WhatsApp. Instead of manually drafting multiple message types—like a quick text follow-up, an order status update using a template, or attaching troubleshooting documentation—you just tell your agent what needs to happen. It handles the nuances: sending a basic free-form text when needed, but automatically switching to a structured template for formal notifications. You can also enrich the conversation by including interactive buttons, asking users questions directly inside WhatsApp. Because this MCP connects messaging actions with data retrieval and generation capabilities through the Vinkius catalog, your agent doesn't just talk; it executes full business processes on the platform you actually use.

---

## Core Capabilities

### 01 — Send structured templates

The agent sends official WhatsApp template messages for automated alerts and notifications.

### 03 — Share various media types

The agent sends images, videos, audio files, or documents using public URLs.

### 05 — Acknowledge messages visually

The agent reacts to specific incoming messages by adding an emoji reaction, keeping the conversation lively and acknowledged.

### 02 — Handle live text conversation

Your AI client writes quick, free-form text responses within the standard 24-hour chat window.

### 04 — Create interactive user choices

You send message buttons with up to three predefined options to guide the user's next move.

# One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to this MCP and provide your unique Gupshup API Key.
- 02 Connect your preferred AI client (like Cursor or Claude) to Vinkius using the credentials.
- 03 Instruct your agent in natural language: 'Send an order confirmation template for user X' or 'Reply with a photo of the document'.

The bottom line is that once connected, you treat WhatsApp communication like any other API call—you just talk to your AI client.

---

## Built For

This MCP is for Operations Managers who manage high-volume customer support channels. It's for Marketing Leads whose campaigns rely on structured, automated messaging flows. And it's for Developers building robust consumer communication tools that can handle everything from simple chat to complex template sequences.

### Customer Support Agent

They use this MCP to quickly share images or documents with a customer and acknowledge their message with an emoji reaction, keeping the conversation flowing without switching apps.

### Marketing Automation Specialist

They rely on it to trigger structured communications, like sending template messages for abandoned carts or interactive buttons asking users to vote in a poll.

### DevOps Engineer

They integrate the MCP into production pipelines to ensure automated message delivery and media handling is reliable at scale.

## What Changes When You Connect

- 
- 01 Stop relying on basic text messages. Use the `send_template_message` tool to send formal, structured alerts for things like shipping confirmation or password resets. The user gets a professional look every time.

---

  - 02 Keep conversations natural by using `send_text_message`. When you're in that 24-hour window with a customer, your agent sends quick replies that feel like a person wrote them, not a bot.

---

  - 03 Improve engagement immediately by utilizing the `send_interactive_message` tool. Instead of just asking 'Yes/No,' you can give users clickable buttons right inside WhatsApp, making actions simple.

---

  - 04 Handle support requests efficiently. You can use `send_media_message` to instantly share troubleshooting guides or product manuals with a single command. No more linking out to an external site.

---

  - 05 Maintain momentum in conversations by using the `send_reaction` tool. It lets your agent acknowledge a customer's message—even if it's just a photo—with a simple thumbs up emoji.
- 

---

## Real-World Applications

### Handling post-purchase follow-up

A marketing lead needs to check in on users who haven't opened their welcome email. They ask the agent to send an 'Order Confirmation' template using `send_template_message` to a list of emails, ensuring every message has the right order ID and name filled in automatically.

### Live support issue triage

A customer is sending photos of damage. The support agent uses `send_media_message` to immediately share a link to the 'Damage Reporting Guide' PDF, while simultaneously using `send_reaction` to acknowledge the photo they just sent.

### Gathering quick user feedback

A team wants to know if users received their service update. They ask the agent to send an interactive message using ``send_interactive_message``, presenting choices like 'Yes, I got it' or 'No, need help,' and analyzing the responses.

### Automating appointment reminders

The scheduling team uses a natural language prompt to trigger an automated WhatsApp message. The agent calls ``send_template_message`` with the user's name and date, confirming their upcoming appointment without human intervention.

---

## Patterns to Avoid

---

### Treating WhatsApp like email

#### ✗ AVOID

The developer tries to send a full marketing pamphlet attached as a PDF, hoping it looks professional. This fails because the messaging service expects structured content or templates.

#### ✓ INSTEAD

Instead of attaching everything, use ``send_template_message`` for the core message (e.g., 'Welcome!') and then follow up with ``send_media_message`` to provide a single, focused PDF document.

### Ignoring conversation context

#### ✗ AVOID

A support agent pastes a long block of text into WhatsApp when the customer just sent a photo. The message looks messy and unguided.

#### ✓ INSTEAD

First, use ``send_reaction`` to acknowledge the customer's picture. Then, if you need more info, use ``send_interactive_message`` to guide them with specific questions instead of a wall of text.

### Forgetting template parameters

#### ✗ AVOID

The agent tries to send an 'Invoice Alert' but forgets to pass the actual invoice number or amount in the prompt, resulting in a generic, unusable message.

#### ✓ INSTEAD

Always check ``get_template_info`` first. When prompting, ensure you provide all necessary variables (like phone number, template name, and array of parameters) for ``send_template_message``.

---

## The Right Fit

Use this MCP if your core communication channel is WhatsApp and your goal is to automate the *content* or *flow* of that conversation. You need a system that can switch between free-text replies, structured alerts (templates), and rich assets (media). Don't use this

if you simply need to send mass SMS messages; look for an SMS gateway MCP instead. Also, don't use it if your main goal is managing contacts—you still need a separate CRM integration for that. However, if your pain point is the manual process of composing varied WhatsApp responses (e.g., 'If this, then send template X with media Y'), then the combination of these tools makes sense.

---

## The constant cycle of copy-pasting status updates and guides.

Today, sending a customer an update means opening WhatsApp, remembering which guide they need, manually uploading the correct document or image link, then composing the text around it. You copy the template details from one spreadsheet, paste them into another system, and finally, you hit send—repeating this tedious sequence dozens of times a day.

With this MCP, your agent handles the whole stack. Tell it: 'Send the latest troubleshooting guide for model Z.' It executes `send_template_message` for the core alert, then uses `send_media_message` to attach the correct document automatically. The result? A complete, professional communication sent in a single prompt.

---

## Gupshup Messaging MCP delivers automated messaging structure.

The manual steps that vanish are the context switching and the content assembly. You don't have to manually determine if the customer needs a quick text reply or a formal template; your agent decides it based on the input, using `send_text_message` when necessary and calling `send_template_message` when required.

What changes is that you stop thinking about *how* to send the message and start focusing only on *what* needs to be communicated. The platform handles the complexity.

---

# Gupshup Messaging: 6 Tools Available

These tools give your AI client the specific functions needed to send everything from simple text replies to rich media packages across WhatsApp.

#	TOOL	DESCRIPTION
01	<code>get_template_info</code>	Retrieves details about a specific WhatsApp template before you send it.
02	<code>send_template_message</code>	Sends an official WhatsApp template, filling in necessary parameters like order numbers or names.
03	<code>send_text_message</code>	Writes a simple, conversational text message within the 24-hour chat window.
04	<code>send_interactive_message</code>	Sends a message that includes up to three clickable buttons for the user's response.
05	<code>send_media_message</code>	Sends content such as an image, video, document, or audio file using a public link.
06	<code>send_reaction</code>	Adds a single emoji reaction to a specific message in the chat history.

---

## See It in Action

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

**U** Send a template 'order\_confirmation' in en\_US to 5511999999999 with params ['12345'].



Template message sent successfully! Status: 200 OK.

**U** Send an interactive message to 5511999999999 asking 'Did you receive your order?' with buttons 'Yes' and 'No'.



Interactive message sent successfully! Status: 200 OK.

**U** React with a thumbs up emoji to message ID msg\_123 for user 5511999999999.



Reaction sent successfully! Status: 200 OK.

---

## Frequently Asked Questions

### 01 How do I use Gupshup Messaging MCP for order confirmations?

You use `send\_template\_message` by providing the template name, recipient number, and filling in parameters like the order ID and confirmation date. This ensures a consistent, branded message every time.

### 02 Can I send images with Gupshup Messaging MCP?

Yes, you use `send\_media\_message`. You just need to provide a public URL pointing directly to the image, video, or document file for your agent to include it.

---

**03 Does Gupshup Messaging MCP support interactive polls?**

It does. Use ``send_interactive_message`` to send a prompt that includes up to three clickable buttons, allowing users to make immediate choices directly within the WhatsApp chat.

---

**04 What is the difference between ``send_text_message`` and ``send_template_message``?**

``send_text_message`` writes quick, natural conversation replies in the 24-hour window. ``send_template_message``, however, sends pre-approved, highly structured alerts for formal business use.

---

**05 How do I make my agent acknowledge a user's message?**

You call the ``send_reaction`` tool and specify which message ID and what emoji you want to react with. This gives immediate, visible acknowledgment in the chat thread.

---

**06 How do I get my Gupshup API Key?**

Log in to your Gupshup Console, navigate to Apps > Your App > Settings, and copy your API Key.

---

**07 Can I send free text messages?**

Yes, but only within the 24-hour session window after the user messages you first.

---

**08 Do I need to pre-approve templates?**

Yes, template messages must be approved by Meta before they can be sent via the `send_template_message` tool.







---

# 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>"gupshup-messaging": { "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

# Gupshup Messaging 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 Gupshup Messaging. 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	Gupshup Messaging MCP
Server ID	019d75ac-ddf3-7219-9a74-d1819af5d573
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/gupshup-messaging](https://vinkius.com/mcp/gupshup-messaging).