MCP SERVER | NO CODE | CLOUD HOSTED

WSLA (WhatsApp) MCP

Automate conversations and notifications via WhatsApp.

WSLA (WhatsApp) MCP connects your AI agent directly to the WhatsApp Business Platform using Meta Cloud API. Send instant text messages, trigger pre-approved business notifications, and even let your agent react to incoming customer media. Use this when you need reliable, scalable communication automation for support or marketing efforts.

A+ Quality Score 100/100

whatsapp-api | conversational-ai | business-messaging | customer-engagement | notifications

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/wsla-whatsapp

SCAN TO VIEW

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.

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.

WSLA (WhatsApp) MCP

5 tools available
Cloud-hosted on Vinkius

This connection lets your AI client handle customer conversations directly inside WhatsApp. You can send standard text replies during active support windows, but you also have access to pre-approved templates for things like appointment reminders or shipping alerts. If a user sends a picture, your agent doesn't just see it; it gets the technical details of that media so your workflow can understand what it is. Want your bot to feel more natural? You can even program reactions using emojis when a customer message comes in. Setting this up means you get reliable, scaled messaging powered by Meta’s infrastructure, accessible through Vinkius's MCP catalog and ready for any AI agent.

Core Capabilities

01 — Send text replies

The agent can send immediate, freeform text messages to a customer within the active support window.

02 — Send notification templates

It allows sending proactive, structured updates using message templates for things like order status or billing alerts.

03 — React to messages

Your agent can automatically send an emoji reaction to a customer's incoming message, making the conversation feel more human.

04 — List available templates

The system retrieves and lists all pre-approved message templates associated with your business account for review.

05 — Analyze media content

When a customer sends images or files, the agent gets structured details about the media type and contents.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to this MCP and provide your Meta System User Access Token and WhatsApp Phone Number ID.

02 Next, connect this MCP to your preferred AI client (Claude, Cursor, etc.) within Vinkius's ecosystem.

03 Finally, prompt your agent to perform a communication action—for example, 'Send the shipping update template to user X.' — and the workflow executes the message.

The bottom line is that you plug in your credentials once, and your AI client handles all the complex API calls for messaging on your behalf.

Built For

This MCP is built for teams whose daily work revolves around communicating with customers via WhatsApp. Think of marketing ops staff who need to send high-volume, templated updates; or support managers tired of manually copying and pasting status updates into chat windows.

Customer Support Specialist

They use the agent to automatically reply with standard text messages or react to messages, improving response speed during peak hours.

Marketing Operations Manager

They automate outreach campaigns by triggering template-based notifications (e.g., for abandoned carts) only to opted-in users.

Backend Developer

They integrate real-time WhatsApp messaging into complex, multi-step workflows where the success of one step depends on a message sent from another tool.

What Changes When You Connect

01 Send immediate replies: Your agent handles common customer queries using the `send_whatsapp_text` tool, keeping response time low during active support hours.

02 Proactive alerts: Use `send_whatsapp_template` to send pre-approved updates for shipping or appointments, ensuring customers get critical information reliably without manual intervention.

03 Natural engagement: By triggering a reaction with `send_whatsapp_reaction`, your agent's responses feel less robotic and more helpful when responding to customer messages.

04 Smarter context: When the user sends media, `get_whatsapp_media_details` pulls the necessary data. Your workflow can then react intelligently based on whether it's a photo or an invoice.

05 Visibility into options: The `list_whatsapp_templates` tool lets you programmatically verify which official message types are available before trying to send them.

Real-World Applications

Handling Post-Purchase Follow-up

A customer buys a product and needs tracking. Instead of waiting for manual emails, the agent triggers `send_whatsapp_template` with the package ID, proactively notifying the customer the moment the shipment leaves the warehouse.

Triage Incoming Support Media

A user messages support and sends a screenshot of an error. Your agent uses `get_whatsapp_media_details` to confirm it's a PNG file, analyzes the content type, and then replies with troubleshooting steps using `send_whatsapp_text`.

Automating Appointment Confirmation

A booking system needs confirmation. The agent calls `list_whatsapp_templates` to verify the correct template name, then executes `send_whatsapp_template` at the exact time needed, confirming the appointment details.

Improving Chat Feel

A support bot is responding to a user's lengthy query. To make the exchange feel less transactional, the agent uses `send_whatsapp_reaction` immediately after receiving the message, showing it read and acknowledged the input.

Patterns to Avoid

Assuming freeform messaging is always safe

✗ AVOID

An agent tries to send a complex marketing announcement using `send_whatsapp_text`. This fails because WhatsApp requires pre-approved templates for bulk communication.

✓ INSTEAD

Always verify your message type first. Use the `list_whatsapp_templates` tool to check which official templates are available, and then use `send_whatsapp_template` for reliable mass messaging.

Ignoring incoming media context

✗ AVOID

The agent receives a picture of damaged goods but replies with generic text. It fails because it never processed the image content.

✓ INSTEAD

First, run `get_whatsapp_media_details` on any incoming media to determine its type. Then, structure your response using that context before sending anything.

Trying to manually format messages

✗ AVOID

A developer tries to build a notification message with custom formatting (bolding, etc.) via `send_whatsapp_text`. This breaks because WhatsApp limits text structure.

✓ INSTEAD

Stick to the established channels. Use `send_whatsapp_template` and let Meta handle the structured formatting for reliable notifications.

The Right Fit

Use this MCP if your primary communication channel *is* WhatsApp, and you need your agent to perform actions that involve sending or interpreting messages. Specifically, use it when you must send templated alerts (requiring `send_whatsapp_template`) or handle

incoming media files (requiring `get_whatsapp_media_details`). Don't use this if your goal is simply internal team communication; those platforms are better suited for that. Also, don't try to scrape data from WhatsApp—this MCP only sends and interprets structured messages. If you need to read raw chat history or manage user profiles outside of messaging actions, look at a dedicated CRM or database connection in the Vinkius catalog instead.

The Grind: Manually Managing Customer Notifications

Right now, when your product has an update—a shipment status change or an appointment confirmation—you're probably stuck clicking into a spreadsheet, copying the relevant ID, opening a separate messaging dashboard, and manually triggering the notification. It's repetitive, error-prone, and slows down both you and the customer.

With this MCP, that manual process disappears entirely. You let your agent detect the trigger (like an order status change) and automatically execute `send_whatsapp_template`. The result is instant, reliable communication right where the user expects it.

Sending WhatsApp Messages with WSLA (WhatsApp)

The manual steps that go away include checking which message type is valid, manually formatting the content placeholders, and ensuring the notification was actually sent to the correct phone number. All of those checks become automated.

Now, your agent manages this entire lifecycle. You just tell it *what* needs to happen (e.g., 'Send the invoice reminder'), and it handles the technical details—the template selection, the data insertion, and the final transmission.

WSLA (WhatsApp): 5 Messaging & Media Tools

Use these five tools to send messages, manage templates, read incoming media data, or react to conversations directly through your AI agent.

#	TOOL	DESCRIPTION
01	`send_whatsapp_reaction`	This sends an emoji reaction to any received WhatsApp message, acknowledging customer input instantly.
02	`get_whatsapp_media_details`	It pulls structured data about incoming media, letting the agent know if a file is an image, video, or document.
03	`list_whatsapp_templates`	The tool searches and retrieves all pre-approved message templates available for your WhatsApp business account.
04	`send_whatsapp_template`	It sends a structured, predefined message using one of the approved message templates.
05	`send_whatsapp_text`	This sends an immediate, freeform text message to a specific WhatsApp number.

See It in Action

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

U Send a WhatsApp message 'Welcome to our service!' to +1234567890.

Message sent successfully! I've triggered the welcome notification to the recipient via WhatsApp Cloud API.

U List all approved templates for my business account.

I've retrieved your templates. You have 10 approved message templates, including 'shipping_update', 'appointment_reminder', and 'welcome_pack'.

Frequently Asked Questions

01 **Can I use WSLA (WhatsApp) MCP for general chat support?**

Yes. You can send immediate responses using `send_whatsapp_text` when a customer is actively chatting with your service, which handles the conversational flow.

02 **How do I handle media files with WSLA (WhatsApp) MCP?**

You use the `get_whatsapp_media_details` tool. This allows your agent to get structured information about any image or file sent by a customer, letting it react intelligently.

03 **Is there a limit on how many templates I can send with WSLA (WhatsApp) MCP?**

The limits are managed by the WhatsApp Business Platform itself. You first use `list_whatsapp_templates` to see what you've pre-approved, and then your agent sends them via `send_whatsapp_template`.

04 **Does WSLA (WhatsApp) MCP work with multiple languages?**

The core messaging capabilities handle text in various supported languages. However, ensure the templates you manage through `list_whatsapp_templates` are fully translated and approved by Meta.

05 **What if I need to send an ad-hoc message outside of a conversation?**

You can use `send_whatsapp_template` for proactive, notification-style messages. For anything non-templated, you must wait until the user initiates contact first.

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 `"wsla-whatsapp": { "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

WSLA (WhatsApp) 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 WSLA (WhatsApp). 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	WSLA (WhatsApp) MCP
Server ID	`019dd18b-e67d-7149-80e2-92594b27e377`
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/wsla-whatsapp.