MCP SERVER | NO CODE | CLOUD HOSTED

Sinch MCP

Send Bulk SMS & Track Delivery Status Instantly

Sinch connects your AI agent to global SMS messaging at scale. Send text messages to single numbers or large groups, track real-time delivery status, and manage contact lists directly through natural conversation with your AI client.

A+ Quality Score 100/100

sms-api | bulk-messaging | delivery-tracking | voice-api | customer-engagement | notifications

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

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.

Sinch MCP

10 tools available
Cloud-hosted on Vinkius

The Sinch MCP lets you use your AI agent to handle all things text messaging. You can send individual texts or launch bulk campaigns across multiple recipients from a simple chat prompt. Need to know if the messages actually got there? Your agent tracks delivery reports, telling you exactly which messages succeeded and which failed. Beyond sending, you manage contacts by creating and deleting targeted groups, keeping your outreach organized. It also lets you check on large message batches before they even leave—you can list recent activity or update parameters for future sends. This full suite of communication tools makes managing campaigns as easy as talking to a teammate. All this capability is accessible through Vinkius, the central hub connecting your agent to thousands of services.

Core Capabilities

01 — Send and manage bulk SMS messages

Your AI client sends text messages to multiple phone numbers or entire contact groups.

02 — Track message delivery status

The system retrieves detailed reports showing which texts were delivered, are still pending, or failed completely.

03 — Organize and maintain contacts

You can create new contact groups, view existing ones, get group details, or permanently delete them when they're no longer needed.

04 — Manage message batches

The tool allows you to list recent SMS batches, review batch details, update parameters for a pending batch, or cancel the entire batch irreversibly.

One Click on Vinkius — From Prompt to Execution

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

01 Connect Sinch MCP to your AI client and provide your required Service Plan ID and API Token.

02 Tell your agent what you need—for example, 'Send a message about the sale to the marketing group.'

03 Your agent executes the necessary steps through this MCP, sending texts, retrieving reports, or managing groups based on your conversational request.

The bottom line is that Sinch lets your AI client execute complex communication tasks using simple chat prompts, eliminating manual API calls and dashboard clicks.

Built For

Anyone who relies on text messaging for critical operations needs this. Think marketing leads frustrated by slow campaign reporting, or support staff tired of manually checking delivery logs after a major outreach effort.

Marketing Campaign Manager

Manages bulk SMS sends and tracks campaign performance instantly to optimize messaging timing.

Customer Support Lead

Sends transactional texts, like order confirmations or password resets, and verifies delivery status immediately after sending.

Software Developer

Integrates messaging workflows into pipelines, automating SMS sends and managing recipient lists without writing boilerplate API code.

What Changes When You Connect

01 You get instant delivery feedback. Instead of waiting hours for manual reports, your agent runs the `get_delivery_report` tool to tell you exactly what happened with a batch.

02 Targeting becomes simple. Use `create_sms_group` and then send messages using `send_sms`. You never have to manually compile lists again; just name the group.

03 Manage large campaigns safely. Before hitting send, use `list_sms_batches` or check details with `get_batch_details` to review everything in one chat exchange.

04 Stop wasted effort. If a campaign fails or needs adjustment, you can run `update_sms_batch` to change parameters without touching the underlying API console.

05 Complete lifecycle control: Need to pause outreach? Use `cancel_sms_batch`. It's a single prompt away from irreversible batch management.

Real-World Applications

Sending post-sale follow-ups

A sales rep needs to send 'Your invoice is ready' texts to 30 clients. They tell their agent, 'Send the invoice reminder.' The agent uses `send_sms` and then immediately calls `get_delivery_report` to confirm that at least 98% of the messages went through.

Auditing outreach lists

The compliance officer needs to know which contact groups exist. They ask their agent, 'Show all active groups.' The agent executes `list_sms_groups`, providing a clean list of names for audit purposes.

Pausing an accidental blast

The marketing team realizes they are about to send a message with the wrong pricing. They tell their agent, 'Stop batch XYZ.' The agent executes `cancel_sms_batch`, preventing any further messages from being sent.

Reviewing historical sends

A manager needs to check campaign performance from last week. They ask their agent to list recent activity, triggering the `list_sms_batches` tool and getting a summary of all past efforts.

Patterns to Avoid

Writing code for every send

✗ AVOID

The user writes boilerplate Python code that calls Sinch's API, manually managing recipient lists in arrays and checking status codes on failure.

✓ INSTEAD

Instead, let your agent run `send_sms` after you ask it to 'Send the promotion text.' The MCP handles the entire workflow—sending, tracking, and reporting—in a single conversation.

Ignoring group management

✗ AVOID

The user keeps sending messages to individuals without organizing them, leading to repetitive requests for contact details.

✓ INSTEAD

Use `create_sms_group` first. Once the contacts are in a managed group, your agent can target them with one simple command.

The Right Fit

Use this MCP if your core business process involves sending high volumes of time-sensitive text messages and requires proof of delivery status. You need to know not just that you *sent* the message, but whether it was actually accepted by the recipient's device. If your primary goal is complex data storage or relational database manipulation—like managing user profiles in a CRM—this MCP won't help. For those cases, you need a dedicated record management tool instead. However, if you are sending notifications, running campaigns, or confirming transactions via SMS, this Sinch integration provides the full lifecycle control: from organizing contacts with `create_sms_group` to tracking every single delivery report using `get_delivery_report`. It's built for active communication flow.

Manual messaging workflows are a nightmare of tabs and logs.

Today, sending an SMS campaign means jumping between your CRM dashboard, the Sinch web portal, and a spreadsheet. You copy contact lists into one place, initiate the send, then wait for confirmation emails to manually track delivery failures or check batch details by logging into another service.

With this MCP, you just talk to your agent. Tell it to 'Send the new promotion to our top clients.' Your agent handles connecting the groups, sending the messages via `send_sms`, and immediately gathering a comprehensive report using `get_delivery_report`. You get confirmation, not logs.

Sinch MCP gives you complete control over your outreach lifecycle.

Before this tool, if you wanted to pause a campaign or update the text content, it required logging into multiple administrative consoles and manually modifying batch parameters. You could only react to problems, not prevent them.

Now, telling your agent 'Update batch X with the new message' is all it takes. It manages the complexity of `update_sms_batch` instantly, giving you proactive control over every outgoing text.

Sinch with 10 Tools

Use these tools to manage the entire lifecycle of SMS messaging—from creating contact groups to sending texts and verifying final delivery status.

#	TOOL	DESCRIPTION
01	`cancel_sms_batch`	Stops a pending group of text messages before they are sent out.
02	`create_sms_group`	Builds a new, dedicated contact list for sending targeted messages.
03	`delete_sms_group`	Permanently removes an entire contact group from the system.
04	`get_batch_details`	Retrieves all specific details about a recent SMS message batch.
05	`get_delivery_report`	Pulls a detailed report on an SMS batch, showing delivery success rates and failures.
06	`get_group_details`	Shows all the members and details associated with a specific contact group.
07	`list_sms_batches`	Displays a list of the most recently processed SMS message batches.
08	`list_sms_groups`	Retrieves a comprehensive list of all existing contact groups.
09	`send_sms`	Sends one or more text messages to specified phone numbers, which must be in international format.
10	`update_sms_batch`	Modifies the parameters of an SMS batch that has already been created but not yet sent.

See It in Action

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

U Send an SMS to +1234567890 saying 'Your order has been shipped!'

SMS sent successfully.
- Batch ID: batch-7k2m
- Recipient: +1234567890
- Message: 'Your order has been shipped!'
- Status: queued for delivery

U Get the delivery report for batch 9876.

Delivery report for batch 9876:
- Total recipients: 12,000
- Delivered: 11,842 (98.7%)
- Pending: 112
- Failed: 46
- Sender ID: 'Vinkius'

Would you like details on the failed deliveries?

Frequently Asked Questions

01 **How do I check if my SMS messages actually delivered using Sinch MCP?**

You use the `get_delivery_report` tool. This pulls a detailed status report showing total recipients, how many were successfully delivered, and exactly how many failed.

02 **Can I send texts to specific contact groups with Sinch MCP?**

Yes. First, you use `create_sms_group` to build the list. Then, your agent sends messages using `send_sms` and targets that group for bulk sending.

03 **What if I need to stop a batch of texts before they go out?**

You use the `cancel_sms_batch` tool. This immediately stops any pending messages associated with a specific batch ID, preventing accidental sends.

04 **Does Sinch MCP support international phone numbers?**

Yes. When using the `send_sms` tool, you must provide all phone numbers in the full international format (like +1234567890) for accurate sending.

05 **How do I list my existing contact groups with Sinch MCP?**

Simply call `list_sms_groups`. This retrieves a complete roster of all defined contact lists, letting you know what groups are ready to use.

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 `"sinch": { "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

Sinch 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 Sinch. 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	Sinch MCP
Server ID	`019d7608-4cf4-70e4-90e5-5a9474d74440`
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/sinch.