MCP SERVER | NO CODE | CLOUD HOSTED

RongCloud MCP

Manage messaging and chatrooms via natural conversation.

RongCloud MCP connects your AI client directly to China's leading IM and RTC platform. It lets you manage complex communication tasks, like tracking user online status or broadcasting messages across large chatrooms—all through natural conversation. Forget logging into developer consoles; simply ask your agent to block a disruptive user, retrieve connection tokens for a new session, or list blacklisted contacts instantly.

A+ Quality Score 100/100

real-time-communication | instant-messaging | chatrooms | user-management | rtc

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

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.

RongCloud MCP

10 tools available
Cloud-hosted on Vinkius

Your AI agent handles the heavy lifting of complex communication systems like RongCloud (融云). This MCP lets you manage everything from real-time messaging to group chatroom moderation without ever touching a developer console. Instead of building custom scripts just to check if someone's online or sending a mass broadcast, your agent acts as a dedicated communications assistant. You tell it what needs doing—whether that's getting an IM token for a new user, checking detailed chatroom information, or updating profile data—and the conversation handles the rest. If you build high-volume social apps or manage enterprise communication, this connection keeps your flow accurate and your user data secure. When you connect this MCP through Vinkius, you gain access to all these controls from a single point: your AI client.

You can use it to send private messages to individuals or blast alerts across entire group chats. You'll also manage the community health by listing blocked users or keeping track of who is on the blacklist.

Core Capabilities

01 — Manage User Online Status

Check if a specific user is currently connected to the IM server and online.

02 — Send Direct or Group Messages

Send private messages to one person or send broadcasts across all participants in a chatroom.

03 — Control User Access

Block specific users from sending messages, or manage lists of blacklisted accounts.

04 — Manage Chatrooms

Create new group chatrooms and retrieve detailed information about existing rooms.

05 — Update User Data

Refresh user profiles to ensure your agent is working with the most current connection details.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius and provide your specific RongCloud App Key and App Secret credentials.

02 Connect the MCP to your preferred AI client (like Cursor or Claude).

03 Start talking. Your agent interprets natural language requests, translates them into API calls, and executes the required action.

The bottom line is you stop writing code for basic communication tasks; you just talk to your agent about what needs doing in RongCloud.

Built For

Product Managers who need real-time visibility into user behavior, Community Moderators tired of manual moderation queues, and DevOps Engineers needing a single pane of glass for system health.

Community Moderator

Manages user misconduct by blocking disruptive users or reviewing chatroom content without logging into the platform dashboard.

DevOps Engineer

Audits system health and connection histories, checking online status for key services or managing blacklists programmatically.

Product Manager

Coordinates feature rollouts by sending targeted messages to specific user groups or verifying if a chatroom needs creating before testing begins.

What Changes When You Connect

01 Stop manually checking dashboards. You can use `check_online` to instantly confirm if a user is available before your agent sends a message, saving time on failed deliveries.

02 Need to ban someone? Instead of navigating menus, simply ask the agent to execute `block_user`. It handles the moderation action immediately through conversation.

03 Coordinating group efforts is simple. Your agent can use `create_chatroom` and then send messages via `send_chatroom_msg`, making broadcast communication as easy as asking a question.

04 The system keeps your credentials safe while you work. The agent uses `get_token` to securely retrieve the necessary connection tokens for any user session, keeping your workflow moving without manual steps.

05 Maintain clean communities by using `list_blocked_users` and `list_blacklist`. Your agent pulls this data instantly so you can see who needs moderation attention.

Real-World Applications

Onboarding a new app feature

The product manager wants to notify 50 key users about the beta launch. Instead of writing a script and iterating through IDs, they ask their agent to first `query_chatroom` for the main group, then use `send_chatroom_msg` to broadcast the announcement to everyone.

Handling spam or toxic behavior

A moderator notices a user is repeatedly sending abusive messages. They prompt their agent to check the status using `check_online`, confirm the issue, and then use `block_user` instantly to prevent further damage.

Debugging connection issues

A dev team member needs to verify if a user's profile details are up to date. They ask their agent to run `refresh_user`, ensuring the data flowing into their application is accurate before sending any messages.

System audit after an incident

The engineering team needs to know who was communicating when a service went down. They instruct their agent to check the `list_blacklist` and review recent chatroom details using `query_chatroom` for all active groups.

Patterns to Avoid

Using APIs directly in code

✗ AVOID

Writing a large, complex script containing multiple API calls just to check status and send a message. This is brittle and hard to maintain.

✓ INSTEAD

Connect this MCP and let your agent handle the sequence. You simply ask: 'Check if Mario is online, and if so, send him a private message.' The agent manages `check_online` and `send_private_msg` for you.

Hardcoding user IDs

✗ AVOID

When sending an alert to 10 different people, manually listing all their User IDs in your code. This breaks if a single ID changes.

✓ INSTEAD

Ask the agent to `query_chatroom` first to get all current participants, and then prompt it to use `send_chatroom_msg` for everyone simultaneously.

Ignoring moderation lists

✗ AVOID

Sending messages that should go to banned users because you forgot to check the system's block list.

✓ INSTEAD

Always start by asking your agent to run `list_blacklist` or `list_blocked_users`. This confirms who shouldn't be receiving any communication.

The Right Fit

Use this MCP if your core business relies on managing user identity, message flow, and group communications within the RongCloud ecosystem. Specifically, you need to programmatically send messages (using `send_private_msg` or `send_chatroom_msg`), manage who can talk (via `block_user`, `list_blacklist`), or track real-time presence (`check_online`). Don't use this if your primary goal is data storage—use a database MCP instead. Also, don't use it just because you need to read user profiles; if all you need is the current profile view, simply calling `refresh_user` might be enough, but for full orchestration, stick with this MCP.

Manual moderation and status checks are a huge time sink.

Right now, if you want to know who's online or ban someone disruptive, you have to jump between multiple developer consoles. You check the user list for blocked accounts; then you open the chatroom details just to see member counts. It’s a tedious cycle of logging in, copying IDs, and clicking through settings.

With this MCP, your AI agent does all that heavy lifting for you. You simply tell it to 'Block User X' or 'Show me the current status.' The conversation handles the complex sequence of calls—the lookups, the updates, the confirmations—and gives you a clean answer.

RongCloud MCP Gives Instant Control Over Messaging and Chatrooms

You no longer have to write specific code blocks just for basic tasks. You don't need to manually retrieve connection tokens before messaging or worry about the complex syntax for broadcasting an alert across a group.

Now, you talk to your agent like talking to a teammate who already knows how RongCloud works. The entire communication flow is managed through plain language, making deployment and moderation faster than ever.

RongCloud MCP: 10 Communication Tools

These tools allow you to control every aspect of communication within the platform, from checking user presence to running full moderation audits.

#	TOOL	DESCRIPTION
01	`block_user`	Prevents a specified user from communicating with the system.
02	`check_online`	Confirms if a user is currently active and connected to the IM server.
03	`create_chatroom`	Sets up a new group chatroom for communication.
04	`get_token`	Retrieves the unique connection token required to initialize a user session.
05	`list_blacklist`	Fetches the list of users who are globally banned from communicating.
06	`list_blocked_users`	Shows which specific users have been blocked by an administrator or user.
07	`query_chatroom`	Retrieves all details about a specified chatroom, including members and settings.
08	`refresh_user`	Updates the stored profile information for an existing user account.
09	`send_chatroom_msg`	Sends a message to every participant within a specific chatroom.
10	`send_private_msg`	Delivers a direct, private text message to one user.

See It in Action

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

U Get an IM token for user 'mario-123'.

I've retrieved the connection token for Mario. You can use this token to initialize the client SDK for this user session.

U Check if user 'renato-456' is currently online.

I've checked the status for Renato. He is currently 'Online' and connected to the IM server.

U Send a text message to chatroom 'room-8821' from user 'bot-01'.

Done! I've sent your message to all participants in chatroom 'room-8821'. The message has been successfully broadcasted.

Frequently Asked Questions

01 **How do I use the `send_private_msg` tool with RongCloud MCP?**

You ask your agent to 'Send a message to User X.' The agent handles the formatting and execution of the private delivery, ensuring it reaches the intended recipient.

02 **What is the difference between `list_blocked_users` and `list_blacklist`?**

`list_blocked_users` shows people blocked by an admin or user in a specific context. `list_blacklist` gives you the master list of users banned system-wide.

03 **Can I create a chatroom using the `create_chatroom` tool?**

Yes, simply ask your agent to 'Create a new group chat for Project Alpha.' The MCP handles the room setup and provides you with the necessary details.

04 **Does RongCloud MCP handle token management automatically?**

The `get_token` tool allows your agent to retrieve connection tokens when needed, so you don't have to manage that sensitive credential flow manually or write code for it.

05 **How do I know if a user is online with RongCloud MCP?**

Just ask the agent 'Is User Y currently online?' It executes `check_online` and reports back their real-time status directly in your conversation.

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 `"rongcloud": { "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

RongCloud 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 RongCloud. 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	RongCloud MCP
Server ID	`019d847a-3833-70cf-81d9-b00fd11bc4c4`
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/rongcloud.