MCP SERVER | NO CODE | CLOUD HOSTED

Jiguang Aurora MCP

Control push messaging, scheduling, and reporting.

Jiguang Aurora / 极光 MCP connects your AI agent to China's leading push notification platform for real-time communication control. Manage device targeting, set complex message schedules, and pull detailed delivery reports—all through natural conversation. Stop navigating portals; just tell your agent what you need done.

A+ Quality Score 100/100

push-notifications | cpaas | mobile-engagement | device-targeting | real-time-messaging | api-integration

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

SCAN TO VIEW

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.

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.

Jiguang Aurora / 极光 MCP

10 tools available
Cloud-hosted on Vinkius

Managing targeted messaging used to feel like juggling multiple dashboards and secret APIs. Now, your AI client handles the complexity of major push notification infrastructure. You can instruct your agent to send specific alerts to highly segmented user groups, pull metadata on devices by their registration ID, or set up campaigns that only fire at perfect times. It's about treating a complex communication system like chatting with an expert teammate.

This MCP gives you full control over scheduling and delivery auditing without ever having to log into the Jiguang portal. Whether you're running large-scale promotions or just automating user verification, your agent acts as your dedicated messaging assistant. We built this connection through Vinkius so you get access to all of these features from one place, letting you focus on strategy instead of clicks.

Core Capabilities

01 — Send Targeted Alerts

Instantly sends customized push notifications to specific users or defined user segments.

02 — Manage Device Details

Retrieves detailed device information and updates tags or aliases for better targeting accuracy.

03 — Set Message Timers

Creates, modifies, and deletes scheduled push tasks to ensure messages arrive at the ideal moment.

04 — Audit Delivery Status

Accesses real-time reports detailing message receipt status, user engagement metrics, and overall campaign performance.

05 — Monitor System Health

Checks your account's API usage limits and quota to prevent communication failures.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP, then input your required Jiguang App Key and Master Secret.

02 Direct your AI client: tell it exactly which action you want—like 'Send a push notification to ID X' or 'List all scheduled tasks.'

03 The agent executes the command using the correct tool, giving you an immediate status update on success or failure.

The bottom line is that you communicate complex messaging tasks in plain language; your AI client translates it into technical actions and delivers the results.

Built For

Product Operations Managers, Marketing Leads, and DevOps Engineers. If you're tired of manually checking multiple dashboards just to know if a campaign message actually got delivered, this MCP saves hours of painful clicking.

Marketing Lead

Coordinates scheduled campaigns, checks user activity reports for the last month, and sends targeted alerts based on segmented behavior.

Product Operations Manager

Automates system-level notifications (like password resets or feature updates) and monitors delivery status to ensure zero downtime communication.

DevOps Engineer

Monitors API quota usage daily, manages device tags for A/B testing groups, and ensures the push infrastructure remains healthy.

What Changes When You Connect

01 Saves time by automating alerts. Instead of manually calling the `send_push` tool for every segment, you just tell your agent to 'Notify X group about Y.'

02 Improves targeting accuracy using `update_device`. You can adjust device tags—like marking a user as 'Beta Tester'—so future messages only hit the right people.

03 Eliminates scheduling guesswork. Use `create_schedule` and `list_schedules` to guarantee promotional alerts run exactly when they should, without manual intervention.

04 Provides immediate accountability. When you need to know if a message worked, simply ask for the report using `get_push_report`; no more guessing about delivery status.

05 Maintains system stability by checking quotas. Running `get_account_quota` before a major campaign prevents frustrating service outages due to API limits.

Real-World Applications

Handling an Urgent Feature Rollout

A PM needs 5,000 users who haven't opened the app in two weeks to get a notice about a new feature. They tell their agent: 'Send a push notification using `send_push` only to inactive users.' The agent executes the tool and confirms delivery immediately.

Running Weekly Campaign Cycles

A marketing manager needs to ensure a product announcement hits all users next Tuesday at 9 AM. They tell their agent: 'Schedule a push for next Tuesday.' The agent uses `create_schedule`, and the manager can check it later with `list_schedules`.

Investigating Low Engagement

An operations analyst notices that one specific device group is ignoring alerts. They ask their agent to run a report on device data using `get_device_info` and then cross-reference it with `get_user_report` to understand the pattern.

Preparing for Peak Sales Season

A DevOps engineer needs to ensure they have enough API capacity for Black Friday. They simply ask their agent: 'What's our remaining quota?' The agent runs `get_account_quota`, giving them the necessary numbers before writing any code.

Patterns to Avoid

Treating it like a simple messaging API

✗ AVOID

A user assumes they just need to send text and forgets about scheduling or reporting. They might try to use `send_push` without realizing the campaign needs follow-up metrics.

✓ INSTEAD

Always pair immediate sending with status checks. After using `send_push`, immediately ask for a delivery check via `get_message_status`. If it's part of a bigger plan, always schedule it first using `create_schedule`.

Over-relying on the UI

✗ AVOID

A user spends an hour clicking through multiple menus in the Jiguang portal just to find out how many users responded last month.

✓ INSTEAD

Don't click around. Just tell your agent: 'Get me the full push delivery report.' The agent runs `get_push_report` and gives you the data instantly.

Ignoring device segmentation

✗ AVOID

Sending a general announcement to everyone, even users who are already part of an internal test group that shouldn't see it.

✓ INSTEAD

Before sending anything, use `get_device_info` and `update_device` to segment your audience. This ensures the right people get the message at the right time.

The Right Fit

Use this MCP if your primary need is managing complex, scheduled, or segmented push communications within a large-scale Chinese ecosystem. You need tools that handle device metadata, scheduling timers, and detailed delivery auditing—those are its core strengths. Don't use it if you just need to send simple emails; use a dedicated email connector instead. Also, don't use this MCP if all you want is to read static user data; for raw data retrieval, look at general database connectors. You must be dealing with push notifications and device targeting specifically. If your goal is simply logging usage, stick to `get_account_quota`; if it involves timing or segmentation, this is the tool.

The manual effort of managing user alerts is a huge time sink.

Right now, coordinating a large-scale alert means logging into a dedicated portal. You'll navigate through menus to find device segments, manually set up the schedule for next Tuesday, and then wait hours later—jumping between tabs—to check if the message was actually received by everyone you intended.

With this MCP, that whole process disappears. You simply tell your agent what needs doing: 'Send a targeted push notification.' The system handles the scheduling, the segmentation, and the delivery tracking automatically. You just get confirmation.

Jiguang Aurora / 极光 delivers instant control over your messaging campaigns.

You no longer need to manually cross-reference device IDs with campaign reports or adjust tags in a separate dashboard. The agent coordinates the entire workflow—from fetching detailed metadata using `get_device_info` to executing a mass send via `send_push`.

What changes is that communication becomes conversational. Your AI client turns complex, multi-step operational tasks into simple questions and answers.

Jiguang Aurora / 极光 MCP: 10 Tools

Use these ten tools to orchestrate every step of your push notification campaign workflow, from scheduling alerts to analyzing device data.

#	TOOL	DESCRIPTION
01	`create_schedule`	Sets up a future push notification task to run at a specified time.
02	`delete_schedule`	Removes an existing scheduled message from the system queue.
03	`get_account_quota`	Retrieves current API usage limits and remaining quotas for your account.
04	`get_device_info`	Fetches detailed metadata about a specific device using its unique identifiers.
05	`get_message_status`	Checks the current delivery status and outcome of a sent push message.
06	`get_push_report`	Retrieves a comprehensive report detailing how many users received and interacted with a specific push campaign.
07	`get_user_report`	Generates reports on overall user activity and growth metrics over a given time period.
08	`list_schedules`	Shows a list of all current or upcoming scheduled push tasks in the system.
09	`send_push`	Immediately sends a customized push notification to one or more specified devices.
10	`update_device`	Modifies tags and aliases associated with a device to refine future targeting criteria.

See It in Action

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

U Send a push notification to Registration ID '1a0918c...'.

I've sent the targeted push notification to the specified device. The message ID is '8821'. Would you like me to check the delivery status in a few minutes?

U Schedule a push task for tomorrow at 10 AM.

Done! I've created a scheduled push task. It is set to trigger tomorrow at 10:00 AM. You can find this task in your schedule list with ID '9920'.

U Show me the user activity report for the last 7 days.

I've retrieved the user activity report. Over the last 7 days, you had an average of 12,500 daily active users and 450 new device registrations. Should I break this down by platform (iOS/Android)?

Frequently Asked Questions

01 **How do I schedule a push notification using Jiguang Aurora / 极光 MCP?**

You use the `create_schedule` tool. Just ask your agent to set it for a specific date and time, and it handles making sure the task is queued up correctly.

02 **Can I check if my push notifications actually worked with Jiguang Aurora / 极光 MCP?**

Yes. Use `get_push_report` to pull detailed reports on message receipt and user engagement, giving you the full picture of campaign performance.

03 **What is the best way to target specific users with Jiguang Aurora / 极光 MCP?**

First, use `get_device_info` to gather metadata. Then, run `update_device` to apply tags (like 'VIP') and only send messages via `send_push` to the newly tagged segments.

04 **Does Jiguang Aurora / 极光 MCP handle API quotas?**

Absolutely. Running `get_account_quota` is a quick, vital check that tells you exactly how much usage you have left before starting any large campaign.

05 **How do I remove an old scheduled message using Jiguang Aurora / 极光 MCP?**

Use the `delete_schedule` tool. You just need to tell your agent which task ID you want gone, and it removes it from the queue.

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 `"jiguang-aurora": { "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

Jiguang Aurora / 极光 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 Jiguang Aurora / 极光. 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	Jiguang Aurora / 极光 MCP
Server ID	`019d844b-3336-70b1-af3b-c001416164dd`
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/jiguang-aurora.