MCP SERVER | NO CODE | CLOUD HOSTED

Xiaomi Push Service MCP

Manage every device notification globally.

Xiaomi Push Service / 小米推送: Send reliable push notifications across the entire Xiaomi ecosystem. This MCP lets your AI agent target specific users by unique IDs, send alerts to defined groups, or broadcast announcements globally, all without touching a developer console.

A+ Quality Score 100/100

push-notifications | mobile-engagement | android-development | message-broadcasting | alias-mapping | real-time-alerts

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/xiaomi-push-service

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.

Xiaomi Push Service / 小米推送 MCP

6 tools available
Cloud-hosted on Vinkius

Need to get a message out to millions of Xiaomi devices? Instead of logging into complex web consoles, your AI client handles the entire process. This MCP connects you directly to Xiaomi's notification platform, giving your agent real-time control over device communication and mobile engagement.

Your agent can send alerts using a unique Registration ID, target users via custom aliases, or broadcast an update across large groups of devices. You even get granular control by subscribing or unsubscribing a specific device to certain topics. This means you can coordinate everything from high-priority system warnings to general marketing announcements—all within your natural conversation flow. If you're looking for centralized access to global and regional push delivery, connecting this MCP through Vinkius makes it available alongside thousands of other tools your agent might need.

Core Capabilities

01 — Send alerts to a specific device ID

Sends a targeted notification directly to a user's unique Xiaomi Registration ID.

02 — Target users by custom name

Dispatches notifications using a predefined user alias instead of the raw device ID.

03 — Broadcast to all connected devices

Sends an announcement that reaches every single app-registered device for your application.

04 — Send alerts to specific groups

Delivers a message only to devices subscribed to a certain content topic.

05 — Manage subscriptions and topics

Allows you to programmatically add or remove device membership from defined communication topics.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/xiaomi-push-service** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your Xiaomi App Secret, making sure to select the correct regional gateway (e.g., Global, China).

02 Connect your preferred AI client—like Claude or Cursor—to Vinkius and authorize access.

03 Ask your agent to perform a specific action, such as 'Send an urgent alert to user alias X' or 'Broadcast this update globally.'

The bottom line is that after setting up the credentials for the right region, you just talk to your AI client and it handles all the complex message routing.

Built For

This MCP is essential for platform developers or operations teams who need reliable, multi-region notification delivery. If managing device alerts requires jumping between separate developer dashboards, this tool saves you time.

DevOps Engineer

Automating transactional alerts, like password resets or system maintenance notices, and monitoring the delivery status of those critical messages.

Product Operations Manager

Coordinating global marketing campaigns or user onboarding broadcasts across different regional markets simultaneously.

Mobile App Developer

Integrating professional-grade push capabilities into an AI workflow, allowing them to test targeting methods—like using the `push_to_alias` tool—on demand.

What Changes When You Connect

01 Stop logging into the MiPush Console. Your agent handles all targeting, whether you need to use `push_to_regid` for a single user or send a broad blast using `push_to_all`. This is pure automation.

02 Achieve precise audience segmentation by utilizing topics. Instead of spamming everyone, your agent can first use `subscribe_to_topic` and then `push_to_topic` to hit only the relevant users.

03 Support global reach from one place. You manage regional gateways for China, Europe, India, and more without needing multiple integrations or complex API calls.

04 Simplify user targeting. If you don't have a full Registration ID handy, `push_to_alias` lets your agent find the right user using only their custom name.

05 Maintain clean communication channels by controlling membership. Use `unsubscribe_from_topic` when a user leaves a group to ensure they stop receiving irrelevant messages.

Real-World Applications

The App Team needs an urgent security warning

A developer realizes 5,000 users are affected by a bug. Instead of manually writing 5,000 API calls, they ask their agent to 'Broadcast the maintenance alert using `push_to_all`'. The MCP instantly sends the message everywhere.

A Marketing Team needs to segment an announcement

The team wants to advertise a new feature only to premium users. They use their agent to check which users are subscribed to the 'premium_tier' topic and then run `push_to_topic` on that group.

An Ops Team needs to confirm a user migrated accounts

The team verifies if an old account ID is still active. They ask their agent to 'Send a test message using `push_to_regid`' to the specific unique device identifier, confirming connectivity.

A System Administrator needs to clean up inactive devices

The admin wants to remove all old accounts from mailing lists. They use their agent to run `unsubscribe_from_topic` for a list of known defunct IDs, keeping the notification system clean.

Patterns to Avoid

Assuming global reach is automatic

✗ AVOID

A user attempts to send a message globally without setting up regional credentials or specifying which region's gateway to use.

✓ INSTEAD

First, ensure your MCP subscription specifies the correct target Region. Then, use `push_to_all` for maximum coverage, making sure you handle any necessary credential mapping.

Sending alerts without knowing the user's alias

✗ AVOID

The developer tries to send an alert using generic email addresses or usernames that don't map correctly to a Xiaomi device profile.

✓ INSTEAD

Use `push_to_alias`. This tool specifically targets users via their custom, known-good aliases instead of relying on potentially incorrect external identifiers.

Overwhelming all users with general messages

✗ AVOID

The system sends a generic announcement to every single user in the database, regardless of whether they care about the update.

✓ INSTEAD

Before broadcasting, use `push_to_topic` after ensuring the target group is properly added using `subscribe_to_topic`. This keeps communication relevant.

The Right Fit

Use this MCP if your primary need is sending high-volume, real-time alerts to devices running in the Xiaomi ecosystem. If you are managing device communications for a mobile app that relies on push notifications, this is essential. Don't use it if you just need general email blasts or SMS messaging; those require different types of connectivity tools. However, if your existing flow involves complex message routing—like sending an alert only after checking a user's subscription status via `subscribe_to_topic` and then running `push_to_topic`—this MCP is built exactly for that complexity.

Getting notifications out used to be a mess of dashboards.

Think about today's process: you have to switch between the MiPush Console, your database, and maybe an analytics dashboard. You copy unique device IDs here, paste them into the console there, manually select the region, and then hit send. It’s tedious, prone to human error, and slows down every critical alert.

With this MCP, that manual choreography vanishes. You just tell your agent, 'Send an urgent update.' The system handles the ID lookup, regional routing, and delivery confirmation automatically. What you get is instant, reliable communication without ever touching a web console.

Xiaomi Push Service / 小米推送: Targeted Delivery at Scale

You eliminate the need for complex API scripting just to manage basic targeting. You don't have to write separate code blocks to check if a user belongs to 'Group A' before sending them an alert. The MCP handles that logic flow.

Now, managing device notifications means writing natural language prompts. It's simple, powerful, and immediately actionable.

Xiaomi Push Service / 小米推送: 6 Tools

These tools allow your agent to perform every possible action related to device communication and notification management within the Xiaomi ecosystem.

#	TOOL	DESCRIPTION
01	`push_to_alias`	Send a notification to any user simply by their custom alias.
02	`push_to_all`	Broadcast an announcement that goes out to every device linked to your app.
03	`push_to_regid`	Send a notification directly using the unique hardware registration ID of a specific device.
04	`push_to_topic`	Deliver an alert only to devices that are subscribed to a particular content topic.
05	`subscribe_to_topic`	Adds a device to a specified communication topic so it can receive future messages.
06	`unsubscribe_from_topic`	Removes a device from a content topic, stopping all further notifications for that group.

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 Xiaomi RegID 'ABCD...1234' with title 'Welcome' and body 'Glad to have you here!'.

I've successfully sent the notification to the Registration ID. The Xiaomi message ID is `1234567890`. Should I also verify if this RegID is correctly formatted?

U Push an urgent alert to user alias 'user_8821' saying 'Your account has a new login'.

I've initiated the high-priority push to the user alias `user_8821`. The message has been submitted to the Xiaomi gateway. Would you like me to check the connectivity status of your configured region?

U Broadcast a global announcement to all users: 'Maintenance starts at 10 PM tonight'.

I've started the broadcast to all devices registered to your application. The maintenance announcement has been successfully sent via Xiaomi Push. Should I verify which region is currently active for these operations?

Frequently Asked Questions

01 **How do I send a message to a single Xiaomi user using the push_to_regid tool?**

You must provide the unique Registration ID. Your agent uses this specific identifier to target and deliver the notification directly to that one device.

02 **Can I broadcast to all users without listing every single ID? (push_to_all)**

Yes. The `push_to_all` tool sends an announcement across your entire application's user base, making it perfect for system-wide maintenance notices.

03 **Is the push_to_topic method better than targeting by alias?**

It depends on your goal. Use `push_to_alias` when you know the person's specific name, but use `push_to_topic` when you need to hit a defined group of interest.

04 **What is the difference between subscribe_to_topic and push_to_topic?**

Subscribing (`subscribe_to_topic`) adds a device's membership to a topic. Pushing (`push_to_topic`) sends the message only after that subscription link has been established.

05 **Does this MCP support different geographic regions?**

Yes, it manages specialized regional gateways for various markets (like China, Europe, and Global), ensuring your alerts reach devices correctly wherever they are located.

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 `"xiaomi-push-service": { "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

Xiaomi Push Service / 小米推送 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 Xiaomi Push Service / 小米推送. 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	Xiaomi Push Service / 小米推送 MCP
Server ID	`019d84a0-65e2-732a-a82a-be7af235ca03`
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/xiaomi-push-service.