MCP SERVER | NO CODE | CLOUD HOSTED

Tyk MCP

Manage all API keys and policies through conversation.

Tyk MCP connects your AI agent directly to your API Gateway dashboard. You manage everything from creating user keys and defining security rules to listing APIs, all through conversation. It gives you conversational control over critical API governance tasks.

A+ Quality Score 100/100

api-gateway | api-management | security-policies | rate-limiting | key-management

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

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.

Tyk MCP

12 tools available
Cloud-hosted on Vinkius

This connector lets you run your entire API infrastructure—from key creation to policy enforcement—using only natural language prompts. Instead of jumping between multiple dashboards or writing complex CLI commands, you talk to your agent and it handles the gateway operations for you. You can define security policies, manage rate limits, generate new keys, and even force a configuration refresh instantly. If managing API governance feels like juggling ten different UIs, this MCP helps centralize that control. It's hosted on Vinkius, making sure any AI client you use connects to all your tools in one place.

Core Capabilities

01 — Manage Access Keys

The agent can generate new API keys for users or organizations and delete existing credentials.

02 — Enforce Security Policies

You define, update, or remove security policies that control who gets access and how often they can hit your APIs.

03 — Audit API Definitions

List all active API definitions in the gateway dashboard, or create brand new ones using a specific format.

04 — Maintain Gateway State

Force an immediate hot reload on the gateway to apply policy and key changes without service interruption.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP, providing your Tyk URL and either a Gateway Secret or Dashboard Token.

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

03 Start by asking the agent to perform an action, such as 'List all payment API definitions' or 'Create a new rate-limit policy'.

The bottom line is, you talk through your desired change, and the MCP executes it against your live API Gateway.

Built For

This connector is for Ops Engineers and Security Analysts who hate manual dashboard clicking. If your job involves ensuring that every service key is properly restricted or if you spend too much time copy-pasting secrets, this MCP saves hours of tedious work.

DevOps Engineer

You automate critical gateway operations, like forcing hot reloads or updating policies, without ever leaving your chat interface.

Backend Developer

You quickly test and generate temporary API keys or check the status of new API definitions during local development cycles.

Security Analyst

You audit existing security policies and user access rights across all APIs to ensure compliance with internal standards.

What Changes When You Connect

01 Stop navigating complex dashboard menus. You simply tell your agent what needs to change—like generating a new key or updating a policy—and it executes the necessary commands on your behalf.

02 Reduce deployment risk by instantly forcing a hot reload using the 'hot_reload' tool. This ensures that any changes made to policies or definitions are live in seconds, not minutes.

03 Maintain strict compliance by having the agent run 'get_policy' and 'get_key'. You can audit access rights and rate limits across your entire ecosystem without manual checks.

04 Speed up development cycles. Instead of asking a teammate for temporary credentials, you use the agent to generate keys or list definitions instantly, using tools like 'create_key' and 'list_apis'.

05 Gain total control over your API structure. You can create new API definitions ('create_api_definition') and enforce governance by setting up granular security policies that limit access.

Real-World Applications

The Quarterly Security Audit

A security analyst needs to prove that all internal microservices are limited to 50 requests per minute. Instead of logging into the dashboard and clicking policy rule after policy rule, they prompt their agent: 'Check every service for rate limits.' The agent uses tools like get_policy and list_apis to build a comprehensive compliance report.

Hotfix Deployment

A backend developer just updated the payment processing logic. They need to ensure the live gateway sees the changes immediately without downtime. They prompt their agent: 'Force a hot reload on the Tyk Gateway.' The MCP runs the action, instantly updating the environment.

Onboarding a New Partner

The ops engineer needs to give a new third-party partner API access. They prompt their agent: 'Create a limited key for Acme Corp with only read permissions on the User profile endpoint.' The MCP uses create_key and create_policy together, guaranteeing scoped access.

API Clean Up

The team decommissioned an old experimental API. Instead of logging in to manually delete its definition and all related keys, the engineer prompts: 'Remove the deprecated reporting API.' The agent handles listing APIs, deleting the definition, and cleaning up associated credentials.

Patterns to Avoid

Manual Config Refresh

✗ AVOID

Making a policy update in the dashboard and then waiting 15 minutes for it to propagate across all environments. You spend time checking logs and sending Slack messages asking if the change went through.

✓ INSTEAD

Use 'hot_reload' via your agent. After updating any key or policy, you prompt: 'Force hot reload on Tyk Gateway.' The gateway refreshes instantly, guaranteeing immediate enforcement.

Key Credential Sprawl

✗ AVOID

A developer needs an old API key status but can't remember which dashboard it was under. They spend 30 minutes clicking through dozens of list views trying to find the right credential record.

✓ INSTEAD

Use 'get_key'. Simply ask your agent: 'What are the details for API key X?' The MCP fetches the exact metadata you need instantly.

Over-Permissive Policies

✗ AVOID

Creating a new policy that is too broad, granting read/write access everywhere because it was easier than defining granular limits. This creates massive security holes.

✓ INSTEAD

Always use 'create_policy' with specific rate limits and targeted resource IDs. Review the details using 'get_policy' before activating anything.

The Right Fit

Use this MCP if your core problem is API Governance, Key Lifecycle Management, or Policy Enforcement across a complex gateway setup. You need to interact with multiple layers of security (keys, policies, definitions) and the process involves state changes (create, delete, update). Don't use it if you just need to view simple data—for instance, if you only want to read logs from an external source, a dedicated log aggregator tool is better. However, if you need to list APIs AND then delete them based on criteria, this MCP handles the full operational cycle. You must be comfortable defining *what* needs to change (e.g., 'Increase rate limit for endpoint Z') rather than just asking for simple information.

API Governance is a series of tedious clicks and secret copy-pastes.

Right now, managing your API gateway means jumping between the dashboard, pulling up key details, manually defining rate limits in one pane, then navigating to another tab to create the policy that enforces those exact rules. You end by having to trigger a manual refresh and hope everything stuck.

With this MCP, you tell your agent what you need —for example, 'I need all services using the Payment API defined with a 10/minute limit.' The whole workflow, from checking definitions ('list_apis') to setting policies ('create_policy'), happens in one conversation. You get instant, verifiable governance.

Manage Keys and Policies with Tyk MCP

The manual steps that disappear are key creation and policy management. No more logging into the dashboard just to generate a temporary credential or spending time adjusting rate limits across multiple UI panels. You use 'create_key' and 'update_policy' conversationally.

You don't just get an answer; you execute the change immediately. It’s about operational certainty, letting your agent perform critical actions like deleting old credentials ('delete_key') or updating definitions without ever leaving your chat window.

Tyk MCP: 12 Tools for API Governance

These tools give you direct conversational access to every key operation in the Tyk dashboard, from creating a policy to forcing an API gateway reload.

#	TOOL	DESCRIPTION
01	`create_api_definition`	This tool creates a brand new API definition within the Tyk dashboard.
02	`create_key`	It generates and provisions a new, usable API key for a user or service.
03	`create_org_key`	This tool creates an elevated organization-level access key.
04	`create_policy`	You define and implement a new set of rules for controlling API access.
05	`delete_key`	This tool revokes an existing API key, making it unusable immediately.
06	`delete_policy`	It removes a defined security policy from the gateway settings.
07	`get_key`	Retrieve all the necessary details for an existing API key, helping you audit its status.
08	`get_policy`	Fetch and review the specific rules and limits of a single security policy.
09	`hot_reload`	Force the entire API Gateway to refresh its configuration, ensuring all changes apply immediately.
10	`list_apis`	This tool shows you a list of every active API definition managed by your gateway.
11	`update_key`	Modify the parameters or status of an existing, live API key.
12	`update_policy`	Change the rules or rate limits on a security policy you've already set up.

See It in Action

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

U List all my API definitions sorted by name.

I've retrieved your API definitions. You have 3 active APIs: 'Auth-Service', 'Payment-Gateway', and 'User-Management'. Would you like to see the details for any of these?

U Get the details for the security policy with ID 'pol-98765'.

Inspecting policy 'pol-98765'... This policy allows access to the 'Auth-Service' with a rate limit of 100 requests per minute and a quota of 10,000 requests per month.

U Force a hot reload on the Tyk Gateway.

Triggering hot reload... The Tyk Gateway has successfully reloaded its configuration and all API definitions are now up to date.

Frequently Asked Questions

01 **How do I manage API keys with Tyk MCP?**

You use the agent to create, read, update, and delete keys. You can ask it to 'create a new key for my staging environment' or 'get details for existing key X.' This keeps all your credential management in one place.

02 **Can Tyk MCP force an immediate configuration refresh?**

Yes, you use the hot_reload tool. After making any changes to policies or definitions, triggering a hot reload ensures the gateway applies those rules instantly without requiring manual intervention.

03 **What is the difference between listing APIs and creating them with Tyk MCP?**

You use list_apis to view all existing API definitions in your dashboard. If you need a new one, you use create_api_definition to build it out.

04 **Does Tyk MCP handle rate limiting and security policies?**

Absolutely. You can define or update any policy using the create_policy and update_policy tools, allowing you to set granular rate limits and access controls for your APIs.

05 **Is this good for auditing my current API setup?**

Yes. To audit everything, use get_key to check credentials, get_policy to review rules, and list_apis to confirm the definition status of every endpoint.

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 `"tyk": { "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

Tyk 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 Tyk. 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	Tyk MCP
Server ID	`019e3900-a486-738d-ba10-003b8edfa74c`
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/tyk.