MCP SERVER | NO CODE | CLOUD HOSTED

Unkey API Management MCP

Control, audit, and manage developer keys with natural language.

Unkey API Management lets your AI agent handle all developer key operations. You can create new keys, check if an existing key is valid, update credentials, and manage the entire lifecycle of your API access directly from conversation. Stop switching between dashboards to audit usage or revoke old tokens.

A+ Quality Score 100/100

api-keys | authentication | rate-limiting | usage-tracking | developer-experience | metadata

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/unkey-api-management

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.

Unkey API Management MCP

8 tools available
Cloud-hosted on Vinkius

Managing APIs shouldn't mean juggling a dozen separate panels. This MCP connects your agent to Unkey, giving it direct control over your entire API infrastructure. You can treat key management like chatting with an engineer—just tell your AI client what you need done. For instance, if a developer loses their credentials, the agent instantly verifies if the key is good and checks its remaining usage credits without you leaving your chat window. Need to audit who's using what? The agent pulls detailed verification analytics for you. Whether you're auditing keys or checking an API configuration, this MCP centralizes that oversight. By connecting Unkey through Vinkius, you give your AI client a single point of truth for all things related to key issuance and usage tracking.

Core Capabilities

01 — Generate New Keys

The agent creates fresh API keys for users, assigning them custom metadata or prefixes.

02 — Verify Key Status

It checks if a provided key is active and reports its current usage limits or remaining credits.

03 — Manage Key Lifecycles

You can list all existing keys for an API, update their details, or permanently delete them.

04 — Monitor Usage Data

The agent retrieves detailed analytics showing how your developers are actually using the service over time.

05 — Inspect APIs

It pulls up a list of all defined APIs and shows their specific configuration details.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/unkey-api-management** — connect your AI agent in three steps.

01 Subscribe to this MCP and enter your Unkey Root Key, which you'll find in your main Unkey dashboard settings.

02 Your AI client authenticates the connection, giving it direct permissions to interact with your API key infrastructure.

03 You simply prompt the agent using natural language—for example, 'List all keys for my billing service.'—and get immediate results.

The bottom line is you control complex developer services and usage data entirely through conversation prompts.

Built For

This MCP is built for platform owners, DevOps engineers, and product managers. If your job involves auditing who has access to what, or tracking how users consume API credits, this saves you from jumping between the Unkey dashboard, internal databases, and monitoring tools.

DevOps Engineer

They use it to quickly audit key deployments during maintenance windows, revoking old keys or listing all associated APIs in minutes.

Product Manager

They track API adoption and usage trends by requesting verification analytics to understand which features are most popular.

Customer Support Specialist

They verify developer keys for customers, resetting or updating credentials based on support tickets without needing internal admin access.

What Changes When You Connect

01 Instant key verification: Instead of logging into a separate portal to check if a developer's key is valid, the agent runs `verify_api_key` instantly. This means faster support responses.

02 Full oversight on infrastructure: Use `list_apis` and `get_api_details` to get a current inventory of all your APIs without navigating complex dashboards. You see everything in chat.

03 Proactive key management: Need to retire an old feature? Simply use `revoke_api_key` to instantly cut off access, then confirm the action with the agent. No more manual cleanup lists.

04 Understanding consumption patterns: The `get_verification_analytics` tool provides usage reports that help you justify pricing tiers or identify underutilized services.

05 Zero context switching: You manage key issuance (`create_api_key`), status updates (`update_api_key`), and usage tracking all in one conversation with your agent.

Real-World Applications

A developer reports their key isn't working.

The customer support specialist asks the agent to `verify_api_key` using the provided token. The agent confirms the key is valid, notes it has remaining credits, and tells the user exactly what plan they are on. Issue solved in three conversational steps.

Quarterly audit of developer access.

The DevOps engineer asks the agent to `list_api_keys` for a specific API ID. The agent returns a complete list, allowing them to cross-reference every key against internal teams and identify keys that should be flagged for revocation.

Launching a new service endpoint.

The product manager tells the agent they need a new API endpoint. The agent uses `list_apis` first to check if the structure exists, then confirms the necessary configuration using `get_api_details` before proceeding.

Identifying inefficient service usage.

The manager prompts the agent to run `get_verification_analytics`. The agent pulls a detailed report showing that one specific API is being hit far more often than expected, allowing the team to optimize billing or performance.

Patterns to Avoid

Treating keys like passwords

✗ AVOID

A user manually tries to find a key in an old spreadsheet and then asks the agent to 'make it active.' This is ambiguous.

✓ INSTEAD

Always use `verify_api_key` first. If the key is valid, you can use `update_api_key` if changes are needed; otherwise, you need to follow proper key generation procedures using `create_api_key`.

Ignoring usage limits

✗ AVOID

A team launches a new feature and immediately gets error messages about exceeding rate limits, but no one knows why.

✓ INSTEAD

Before launching, run `get_verification_analytics`. This reports historical consumption data, letting you predict load and adjust your key's allocated usage.

Over-relying on dashboards

✗ AVOID

The engineer spends 20 minutes clicking through the Unkey dashboard tabs just to get a list of all APIs.

✓ INSTEAD

Just ask the agent to `list_apis`. It retrieves the entire API inventory and configuration instantly, saving you time and clicks.

The Right Fit

Use this MCP if your core pain point is managing the *lifecycle* and *access control* of tokens. Specifically, when you need instant verification (`verify_api_key`), bulk management (revoking keys with `revoke_api_key`), or centralized oversight of usage data (`get_verification_analytics`). Don't use this if your problem is purely UI/UX; for example, if you just want to visually display a status badge on a dashboard, that requires front-end code. If your need is simply logging key access events *without* controlling them,

an event streaming tool might be better. But when the action needs to be 'create,' 'read details,' or 'delete'—this MCP has the tools for you.

The headache of manual API audits and cleanup

Today, managing developer credentials feels like a scavenger hunt. You open the dashboard to list APIs; then you navigate to a second panel just to check usage analytics; if something is wrong, you have to copy a key ID, switch tabs, and finally go somewhere else to revoke it. This multi-step process means missing keys or slowing down critical fixes.

With this MCP, the entire process happens in conversation. You tell your agent to audit an API's usage, and it pulls together the list of APIs, the key analytics, and the ability to revoke them—all without you ever leaving the chat window. It's centralized control.

Unkey API Management gives you total command over your keys

The tedious steps of listing all APIs, then manually checking each key's status against a separate usage spreadsheet, are gone. The agent handles the full sequence: `list_apis` to see what exists, and `get_verification_analytics` to know who is using it.

Now, managing your entire API ecosystem feels like asking a knowledgeable teammate for help. You get precise actions done instantly.

Unkey API Management: 8 Tools

These tools let you programmatically create new keys, check key validity, list APIs, and pull detailed usage reports on demand.

#	TOOL	DESCRIPTION
01	`create_api_key`	This tool generates and issues a brand new API key for your users to use.
02	`revoke_api_key`	It permanently deletes an existing API key, immediately removing all access rights associated with it.
03	`get_api_details`	You can use this to pull up specific configuration details for any defined API within your project.
04	`get_verification_analytics`	This tool retrieves detailed usage stats, allowing you to see how often and how much keys are being used.
05	`list_apis`	It pulls up a comprehensive list of every API that has been defined in your Unkey project.
06	`list_api_keys`	This function displays all active and inactive keys linked to a specific API ID you provide.
07	`update_api_key`	You can change details on an existing key, such as its metadata or usage limits, without revoking it.
08	`verify_api_key`	This tool instantly confirms if a given API key is currently valid and active for use.

See It in Action

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

U List all APIs in my Unkey project.

Fetching APIs... I found 2 APIs: 'Public Storefront' (ID: api_123) and 'Admin Dashboard' (ID: api_456). Which one would you like to manage?

U Verify the API key 'user_sk_abcdef123'.

Verification result: Key is VALID. Owner: 'user_789'. Remaining credits: 4,500. This key is associated with the 'Premium Plan'.

Frequently Asked Questions

01 **How do I use the Unkey API Management MCP to check if a key works?**

You run `verify_api_key` and provide the token you want checked. The agent immediately tells you if it's valid, who owns it, and how many credits are left.

02 **Can I use Unkey API Management to find out what APIs exist?**

Yes, just ask the agent to `list_apis`. It will retrieve a complete rundown of every API endpoint defined in your project for you.

03 **If I need to delete an old key, which tool do I use? Is it Unkey API Management?**

Use the `revoke_api_key` tool. It permanently deletes access rights for that specific key immediately.

04 **How do I get usage data from Unkey API Management MCP?**

You run `get_verification_analytics`. This pulls detailed stats showing how developers have used the service over a specified period.

05 **Does this MCP let me create new keys for users?**

Yes, you use the `create_api_key` tool to generate fresh API keys and assign them specific metadata or ownership details.

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 `"unkey-api-management": { "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

Unkey API Management 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 Unkey API Management. 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	Unkey API Management MCP
Server ID	`019d8495-bb53-7161-ab45-377da5487f1b`
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/unkey-api-management.