MCP SERVER | NO CODE | CLOUD HOSTED

Stoplight MCP

Audit, inspect, and map complex API schemas conversationally.

Stoplight MCP connects your AI client directly to an API design and documentation system. Use it to audit, review, and map entire API lifecycles without ever leaving your chat window. Quickly inspect schemas, track team activity logs, or check project ownership details using simple conversation prompts.

A+ Quality Score 100/100

openapi | api-design | documentation | api-governance | schema-management | technical-writing

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026

vinkius.com/mcp/stoplight

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.

Stoplight MCP

7 tools available
Cloud-hosted on Vinkius

Managing APIs used to mean jumping between complex dashboards: the schema view, the project roadmap, the activity feed—all separate tabs that kill momentum. This MCP pulls all of that context into your agent. You can ask your AI client for an overview of every workspace or pull up a full OpenAPI schema for a specific endpoint just by talking to it. It lets you run architectural reviews and documentation checks conversationally. When you need this level of technical depth, Vinkius makes sure the Stoplight connection is ready for any MCP-compatible client, letting your agent handle complex tasks like checking recent changes or listing all contributors across multiple projects. You get instant visibility into API structures and development status without needing to click through a single dashboard.

Core Capabilities

01 — Map the full architecture

List every accessible workspace, project repository, and individual documentation node within your entire organization.

02 — Audit team governance

See exactly who owns a project or which members are authorized to contribute to specific API workspaces.

03 — Track development activity

Pull up recent logs of changes, seeing who updated what endpoint and when across the whole system.

04 — Inspect schema details

Retrieve the full, structured documentation for any given API node, including its expected data types and parameters.

One Click on Vinkius — From Prompt to Execution

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

01 First, authenticate your AI client by installing the Stoplight MCP module into your environment.

02 Next, configure the connection by providing your unique `STOPLIGHT_WORKSPACE` slug and a valid API token.

03 Finally, simply prompt your agent with an action, like 'Show me all projects in core-billing' or 'What were the last 5 changes to this workspace?'

The bottom line is that you talk to your AI client, and it handles the complex navigation across Stoplight's entire library of API definitions.

Built For

This MCP is for senior technical staff—API Architects, Platform Engineers, and Technical Leads. If your job involves reviewing dozens of endpoints or keeping track of cross-team dependencies across large codebases, you'll hate clicking through multiple dashboards just to get an overview.

API Architect

Reviews project readiness by listing projects and then diving into specific documentation nodes to validate schemas before design sign-off.

Backend Engineer

Resolves integration dependencies quickly by querying endpoint models across multiple, independent Stoplight projects without context switching.

Technical Writer

Summarizes complex API documentation updates by pulling recent activity logs and project details into a single, coherent narrative.

What Changes When You Connect

01 You cut out the dashboard-hopping. Instead of navigating through multiple tabs to see project structure, you simply ask your agent for a list of all projects using `list_projects` and get immediate results.

02 Stop manual documentation reviews. Need to know if an endpoint changed? Use `list_workspace_activity` to pull up recent logs instantly, letting you track changes without digging through date-stamped audit tables.

03 Deep dive into schemas on demand. Want the full structure of a payment node? Calling `get_node_details` gives you the raw OpenAPI schema immediately, perfect for validation checks.

04 Understand team ownership fast. Use `list_workspace_members` to instantly confirm who has admin or editor rights in any given workspace, solving governance questions right away.

05 See your whole picture at once. Start with `list_workspaces` to map the entire organizational container structure before drilling down into specific projects like 'core-billing-api'.

Real-World Applications

Reviewing a cross-functional dependency update

A backend engineer needs to know if two separate services (Billing and Inventory) are using the same version of a user ID field. They ask their agent to list all nodes in both projects, then use `get_node_details` on the relevant endpoints from each project to compare schemas directly.

Onboarding a new team member

A platform architect needs to assign roles. They ask their agent to call `list_workspace_members`, then review the output to see who has editing permissions, and finally check project details using `get_project_details` to confirm scope.

Tracking a critical bug fix deployment

A technical writer asks their agent for recent activity. The agent calls `list_workspace_activity`, providing the exact timestamp and user who updated the API, allowing the writer to write accurate release notes.

Mapping out a new feature scope

An architect wants to see all potential areas for a new payment gateway. They first use `list_workspaces` to define the boundaries, then run `list_projects` inside that workspace to catalog every existing API repository.

Patterns to Avoid

Searching manually through UI filters

✗ AVOID

A user is stuck in the Stoplight dashboard trying to filter projects by owner, then finding documentation nodes one by one.

✓ INSTEAD

Instead, ask your agent to run `list_projects` first. Then, use `get_project_details` for metadata and `list_workspace_members` if you need ownership information.

Asking the AI client general questions

✗ AVOID

Prompting 'Tell me about our APIs' results in a vague summary because the agent can’t know what to search for.

✓ INSTEAD

Be specific. Prompt: 'List all projects and show recent activity.' This triggers both `list_projects` and `list_workspace_activity`, giving you concrete data.

Copying and pasting schemas into a document

✗ AVOID

A developer copies raw schema text, which is difficult to read or verify for type safety.

✓ INSTEAD

Ask your agent to use `get_node_details`. It will pull the structured OpenAPI schema definition directly into your chat, making it clean and ready for review.

The Right Fit

Use this MCP if your workflow involves auditing large-scale API documentation or managing complex architectural governance. If you spend more time navigating dashboards than actually writing code or content, this is for you. You need conversational access to system metadata—things like who owns a project (`list_workspace_members`), what changes happened recently (`list_workspace_activity`), and the full technical definition of an endpoint (`get_node_details`). Don't use this if your only goal is

general knowledge retrieval or simple text generation. If you just need to write a blog post about APIs, your AI client handles that fine without needing this MCP. But if you need to *validate* the API definitions themselves, stick with Stoplight.

The Overhead of Context Switching

Today, reviewing a single API dependency requires juggling at least four different screens: one for the project overview, one to list all endpoints, another to check activity logs, and finally, a fourth page just for the raw schema. You spend more time clicking through tabs and navigating menus than actually analyzing the data.

With this MCP, you keep everything in your chat window. Tell your agent what you need—say, 'What are the latest changes on the Billing API?'—and it pulls that information from `list_workspace_activity` and presents a clean, actionable summary. The result is instant context without leaving your development environment.

Stoplight MCP: Instant Schema & Project Insights

You don't have to run multiple reports or export CSVs just to get a status check. Instead, you can ask the agent to execute `list_projects` to map all available APIs and then use `get_project_details` on any one of them for full visibility.

It fundamentally changes your workflow from 'data gathering' to 'decision making.' You stop managing the tabs and start solving the architectural problem.

Stoplight MCP – 7 Tools

These seven tools allow your AI agent to interact with every part of the Stoplight API, from listing all workspaces down to retrieving specific documentation node schemas.

#	TOOL	DESCRIPTION
01	`get_node_details`	Extracts the full, detailed schema and raw text definition for one selected documentation node.
02	`get_project_details`	Gives high-level metadata and status details for a chosen Stoplight project.
03	`list_workspace_activity`	Fetches recent logs detailing who made changes and what was updated within the workspace.
04	`list_workspace_members`	Lists all users associated with the workspace and their current roles or permissions.
05	`list_project_nodes`	Lists all individual components—files, endpoints, or models—inside a specific API documentation project.
06	`list_projects`	Lists all specific API documentation repositories housed within a single workspace.
07	`list_workspaces`	Retrieves a list of all top-level organizational containers accessible in your Stoplight account.

See It in Action

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

U List my Stoplight projects and show recent workspace activity.

Connected successfully.
Projects (`list_projects`):
- `Payment V2 Refactor`
- `Internal Logistics API`

Recent Activity (`list_workspace_activity`):
- *alex.dev* updated the `POST /transactions` endpoint with new security parameters.

U Retrieve the detailed schema documentation for the processing node in our core billing API project.

Accessing workspace files...

Node Details (`get_node_details`):
Extracted the OpenAPI schema for the `processing` endpoint. It expects a payload including `transactionId`, `amount`, and `currency`. Validation is active. How would you like to review these definitions?

U List all active members in the current workspace.

Checking organizational permissions...

Workspace Members (`list_workspace_members`):
- Jane Doe (Admin)
- John Smith (Editor)
- Alex Dev (Viewer)

All 3 members are currently active and mapped to active API projects.

Frequently Asked Questions

01 **How does the Stoplight MCP help with API governance?**

The Stoplight MCP provides tools like `list_workspace_members` to track project ownership. You can instantly confirm who has admin rights and which users are contributors across your entire organization's APIs.

02 **Can I use the Stoplight MCP to check API schema details?**

Yes, calling `get_node_details` retrieves the full OpenAPI schema for any specific node. This is crucial for ensuring type safety across your documentation.

03 **What if I need to see what changed recently in my APIs using Stoplight MCP?**

You use `list_workspace_activity`. It provides an immediate log of recent changes, telling you which endpoints were updated and by whom, saving you manual checks.

04 **How many different kinds of API projects can I list with Stoplight MCP?**

You first call `list_workspaces` to see the top-level groupings. Then, running `list_projects` inside that workspace shows all contained documentation repositories.

05 **Does the Stoplight MCP require specific credentials?**

Yes, you must authenticate and provide your `STOPLIGHT_WORKSPACE` slug along with a valid API token to connect the agent to your account data.

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 `"stoplight": { "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

Stoplight 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 Stoplight. 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	Stoplight MCP
Server ID	`019d760d-7deb-708b-b556-12b444403f68`
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/stoplight.