MCP SERVER | NO CODE | CLOUD HOSTED

Hologram.io MCP

Manage your entire global IoT fleet from chat.

Hologram.io MCP connects your AI agent directly to global cellular connectivity data. Manage your entire IoT device fleet—from checking current billing cycle usage to pausing specific SIM cards and pinpointing approximate locations using cell tower triangulation. Control everything about your connected devices, all through conversation.

A+ Quality Score 98.33/100

cellular-connectivity | sim-management | device-fleet | iot-monitoring | data-usage

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

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.

Hologram.io MCP

11 tools available
Cloud-hosted on Vinkius

Managing a large-scale Internet of Things (IoT) deployment means juggling dozens of dashboards for data usage, connectivity status, and device location. This MCP lets you skip the clicks entirely. You connect it to your preferred AI client and treat it like having a dedicated IoT Operations Manager sitting next to you. Instead of manually exporting logs or jumping between five different screens just to check if a remote sensor is online, you ask your agent to do it. It retrieves everything: list all devices, get the approximate GPS coordinates for your entire fleet, or even pause data services on a specific SIM card if usage spikes unexpectedly. This kind of deep operational control, accessible via natural conversation, is what Vinkius makes possible across thousands of specialized MCPs.

Core Capabilities

01 — Assess Overall Data Consumption

Review current billing cycle usage and check aggregated daily data statistics for the entire fleet.

02 — Inventory and Diagnose Devices

List all connected IoT devices in your account, retrieve specific device metadata, and view recent data sessions to troubleshoot connectivity issues.

03 — Manage Cellular Connectivity

See all active SIM cards and their plans, then instantly pause or unpause data services for a single card via chat command.

04 — Track Physical Assets

Get the approximate GPS coordinates for your devices based on cell tower triangulation data.

05 — Audit Communications History

Retrieve logs of device-originated SMS messages for compliance or troubleshooting purposes.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and input your Hologram API Key into the Vinkius platform.

02 Authorize your AI client to use the connectivity tools, giving it access to your device fleet data.

03 Ask your agent a natural language question—for example, 'Show me where all my devices are' or 'What is the total usage this month?'

The bottom line is that you get real-time operational visibility into complex, distributed hardware without ever touching an API call.

Built For

This MCP is for the Infrastructure Engineer who gets frustrated by spending hours clicking through vendor dashboards just to confirm a simple status check. It's also for the Operations Lead whose job requires constant, real-time oversight of billing and location data across dozens of remote assets.

IoT Field Engineer

Checks device connectivity and retrieves detailed data session logs on demand to diagnose why a sensor is reporting stale readings.

Fleet Operations Manager

Monitors for unexpected data overages, uses the list_sim_cards tool to verify plans, and pauses inactive SIMs proactively to save money.

Network Reliability Lead

Maintains a continuous overview of device location via cell tower triangulation and reviews SMS messages for unusual activity patterns.

What Changes When You Connect

01 Stop manually compiling usage reports. You can check the billing cycle usage or run get_daily_usage_stats to see consumption patterns instantly, without leaving your conversation window.

02 Control connectivity with single commands. If a device is acting up, use pause_sim_data on the specific SIM card instead of needing to log into a separate portal and flip a switch.

03 Know where your assets are, even if they aren't transmitting GPS data. Use get_device_locations to map approximate coordinates using available cell tower triangulation points.

04 Gain full visibility into device health by listing all IoT devices with list_iot_devices or getting detailed readouts via get_device_details when troubleshooting a single unit.

05 Streamline maintenance tasks by reviewing the history of messages through list_sms_messages, allowing you to track communications without digging through raw logs.

Real-World Applications

Detecting Data Overages

A Field Manager asks their agent: 'What is our usage this month?' The MCP runs get_billing_cycle_usage, showing a 50% spike. The manager then uses list_iot_devices to identify the culprit unit and checks its recent data sessions using get_recent_data_sessions.

Locating Lost Equipment

An Ops Lead needs to know where a piece of equipment is. They ask for device location, and the MCP runs get_device_locations, providing an immediate map overview based on cell tower triangulation so they can send recovery teams.

Simulating Downtime

An Engineer needs to test network resilience. They use list_sim_cards and then invoke pause_sim_data on a non-critical link, verifying that the system reacts correctly before testing with real hardware.

Compliance Audits

A Compliance Officer needs to verify all communication history for a given week. They ask the agent to list_sms_messages and retrieve detailed metadata on device activity, compiling an instant audit trail.

Patterns to Avoid

Using separate dashboards

✗ AVOID

The user has to jump between the Billing page (for usage), the Device list (for status), and the Location tab (for GPS) just to answer one question.

✓ INSTEAD

Instead, ask your agent to get_device_details for a specific unit and combine that with checking the billing cycle usage. Your AI client handles the cross-referencing instantly.

Copying logs manually

✗ AVOID

After finding an error code on a device, the user must download the data session log file, open it in Excel, and search for the relevant timestamp.

✓ INSTEAD

Simply prompt your agent to get_recent_data_sessions. The structured output provides exactly the log entry you need without any manual copy-pasting.

Overlooking SIM plans

✗ AVOID

A manager sees high usage but doesn't know if the device is on a cheap, limited plan or a premium one.

✓ INSTEAD

First, run list_sim_cards to see all associated data plans. Then use get_billing_cycle_usage to confirm if that plan was exceeded.

The Right Fit

Use this MCP when your primary need is real-time operational oversight of a distributed hardware fleet's connectivity, location, or billing status. It excels at synthesizing disparate data points—like linking device IDs to usage logs and approximate physical locations—into conversational answers.

Do NOT use this if you are managing software infrastructure (use an API gateway MCP) or if your core need is only generating reports for historical analysis that requires large-scale dataset processing. If all you need is a basic list of devices, the simple listing tools will suffice; don't overcomplicate it by asking for location data when you just want names.

The headache of managing scattered IoT dashboards.

Right now, checking on a device fleet means jumping across 3-4 different vendor portals. You check the main dashboard to see if it's online, then click into a separate 'Usage' tab to find out how much data it used yesterday, and finally open another section just to get its approximate location coordinates. It takes ten clicks and twenty minutes of context switching.

With this MCP, you ask your agent one question—like, 'What is the current status and location of all units in Sector 3?' The system runs list_iot_devices for the unit roster, gets_device_locations for the map data, and synthesizes it into a single answer. You get operational insight instantly, without ever leaving the chat.

Get full control using the Hologram.io MCP.

The biggest time sink is manually adjusting services. If you spot a device with unusually high usage after running get_recent_data_sessions, your old workflow required logging into the billing system and clicking through multiple menus to suspend service. That’s slow, prone to error, and expensive.

Now, you simply ask your agent to pause_sim_data using the specific SIM card ID. The action happens immediately, directly from your conversation thread. This MCP turns complex, multi-step infrastructure management into a simple chat command.

Hologram.io with 11 Tools

Use these tools to monitor fleet status, check billing cycle usage, get location data, manage SIM card states, and diagnose connectivity issues.

#	TOOL	DESCRIPTION
01	`get_billing_cycle_usage`	Checks the total data usage consumed within the current billing cycle period.
02	`get_daily_usage_stats`	Gathers aggregated data usage statistics for a specific daily period.
03	`get_device_details`	Retrieves detailed, specific information about one particular IoT device.
04	`get_device_locations`	Gets the approximate GPS coordinates for all linked devices using cell tower data points.
05	`get_recent_data_sessions`	Retrieves a chronological log of the most recent data connection sessions used by your fleet.
06	`list_available_data_plans`	Lists all possible cellular data plans you can activate across your organization.
07	`list_iot_devices`	Retrieves a comprehensive list of every IoT device registered in your Hologram account.
08	`list_sim_cards`	Lists all connected SIM cards and shows which data plans are associated with each link.
09	`list_sms_messages`	Retrieves the historical log of SMS messages originating from a device.
10	`pause_sim_data`	Temporarily suspends all data services for a specific SIM card based on its unique ID.
11	`unpause_sim_data`	Resumes data services that were previously suspended on an active SIM card.

See It in Action

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

U List all IoT devices and show which ones are currently active.

I've retrieved your device list. You have 12 devices in your fleet, including 'Sensor Unit 01' (Active), 'Gateway Beta' (Active), and 'Tracker 05' (Suspended). Would you like to see the recent data usage for the active devices?

U Show me the total data usage for the current billing cycle.

Accessing billing usage... For the current cycle, your fleet has consumed 450.2 MB of data. Your monthly allowance is 1 GB, leaving you with 549.8 MB remaining. Should I check which device has the highest consumption?

U Find the approximate location of all devices in my account.

Retrieving fleet locations... I've mapped your devices based on cell tower data. Most of your fleet is currently concentrated in the 'Greater Chicago Area', with 2 units appearing near 'San Francisco'. Would you like the specific coordinates for any device?

Frequently Asked Questions

01 **How does Hologram.io MCP know the location of my devices?**

The MCP gets approximate coordinates using cell tower triangulation data via the get_device_locations tool. This means it maps where the signal is strongest, not necessarily the GPS point on record.

02 **Can I use Hologram.io MCP to see historical billing records?**

Yes, you can check data usage for the current cycle using get_billing_cycle_usage and retrieve daily statistics with get_daily_usage_stats.

03 **What is the difference between list_iot_devices and list_sim_cards?**

list_iot_devices gives you a roster of all hardware units. list_sim_cards provides an inventory of the cellular links attached to those units, along with their specific plans.

04 **Does Hologram.io MCP support pausing services?**

Yes, you can pause or resume data service for a single SIM card using the pause_sim_data and unpause_sim_data tools, which is useful for cost control.

05 **Can I use Hologram.io MCP to find out if my devices sent SMS messages?**

The list_sms_messages tool allows you to retrieve the full history of device-originated SMS communication logs, which is vital for compliance.

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 `"hologramio": { "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

Hologram.io 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 Hologram.io. 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	Hologram.io MCP
Server ID	`019d75b2-0a2e-73f5-afec-57c3487864cb`
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/hologramio.