MCP SERVER | NO CODE | CLOUD HOSTED

Web3 Indexing MCP

Query assets across EVM and Solana chains.

The Graph (Web3 Indexing) gives your agent direct access to real-time blockchain data across major chains like Ethereum, Polygon, and Solana. It lets you query everything from token transfers and wallet holders to complex DEX swap events using standardized tools and custom GraphQL subgraphs. Stop guessing what's on the chain; start querying it directly.

A+ Quality Score 100/100

web3 ethereum solana graphql indexing defi

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/the-graph-web3-indexing

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.

The Graph (Web3 Indexing) MCP

8 tools available
Cloud-hosted on Vinkius

This MCP connects your AI agent directly to decentralized ledger data. You can pull historical and real-time information across multiple major ecosystems, including EVM chains like Ethereum and Solana. Instead of opening a dozen different block explorer tabs, you use standardized tools to query specific actions. Need to know who holds the most tokens for a particular asset? Use the tool designed to get those top holders. Want to track how much an address's balance changed over months? You can pull historical records formatted for analysis. Even if your data is highly specialized, running raw GraphQL queries against custom subgraphs gives you deep control over what information you retrieve. It's a huge amount of on-chain intelligence made available through simple commands, which Vinkius hosts and manages so any MCP-compatible client can use it right away.

Core Capabilities

01 — Track token movements

It fetches detailed records of every token transfer that happened across specified EVM or Solana chains.

02 — Identify major asset holders

You get comprehensive lists and counts of wallets holding specific tokens on both EVM and Solana networks.

03 — Analyze decentralized swaps (DEX)

It gathers records detailing token swaps that occurred on popular automated market makers like Uniswap or Raydium.

04 — Review historical balances

You pull time-series data showing how an address's balance changed over a specific period, perfect for portfolio tracking.

05 — Execute custom queries

It allows you to run raw, specialized GraphQL requests against any existing subgraph ID for unique data points.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/the-graph-web3-indexing** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your Graph API Token, URL, and key details.

02 Your AI client calls the specific tool, telling it exactly what chain (EVM or Solana) and what type of data you need (e.g., 'swaps' or 'holders').

03 The MCP executes the request against the live blockchain indexer and returns a structured list of data points to your agent.

The bottom line is that it turns complex, multi-chain blockchain reads into simple commands for your AI client.

Built For

This MCP is built for technical people who deal with digital assets and need verifiable data. It's for the DeFi analyst tired of switching between Etherscan, Solscan, and Google Sheets, or the developer who needs to debug smart contract events without writing complex Web3 SDK code.

Data Analyst

Runs historical balance queries (get_evm_historical_balances) and aggregates token holder distributions across different chains for market reports.

Web3 Developer

Uses the query_subgraph tool to test specific smart contract events or verify on-chain state directly within their IDE context.

DeFi Trader/Researcher

Checks recent swaps (get_svm_swaps) and tracks token transfers across both EVM and Solana to spot market activity patterns.

What Changes When You Connect

01 Instantly track all token movements using get_evm_transfers or get_svm_transfers. You don't have to manually check multiple block explorers; your agent just runs the tool.

02 Pinpoint market concentration by listing top holders with get_evm_holders and get_svm_holders, giving you immediate insight into asset distribution.

03 Analyze trading activity across major DEXs. Whether it's getting_evm_swaps or get_svm_swaps, the data is structured so your agent can interpret volume instantly.

04 Go beyond standard queries. The query_subgraph tool lets you hit niche subgraphs for specialized metrics—something general tools can't touch.

05 Understand portfolio performance over time by using get_evm_historical_balances, which provides change data in a format ready for spreadsheet analysis.

Real-World Applications

Determining market dominance of an asset

A researcher asks their agent: 'Who controls the most tokens for Asset X?' The agent uses get_evm_holders to pull a list, quickly identifying the largest wallet and determining if concentration risk is high.

Tracking rapid arbitrage activity

A trader needs to know about recent price discrepancies. They ask their agent to run get_svm_swaps and get_evm_swaps, comparing the volume of swaps on two different chains in minutes.

Auditing a contract's lifecycle

A developer needs to verify if funds moved correctly. They use get_evm_transfers and then query_subgraph with specific parameters to trace the full movement path of the asset.

Measuring investment growth over quarters

An analyst wants a quarterly report on portfolio performance. The agent executes multiple calls to get_evm_historical_balances, compiling the necessary time-series data points for the final document.

Patterns to Avoid

Trying to find all data in one place

✗ AVOID

A user tries to copy and paste a complex GraphQL query into an unstructured text prompt, expecting the agent to 'just figure it out' or run multiple basic tools sequentially.

✓ INSTEAD

Don't try to force a single tool. Instead, use the specialized get_evm_swaps for swap data, then follow up with query_subgraph if you need metrics that require a custom ID.

Mixing chain data types

✗ AVOID

A developer tries to run get_svm_holders but inputs an EVM address because they think the tool is universal.

✓ INSTEAD

Always check the specific naming convention. Use get_svm_holders only with Solana addresses, and use get_evm_holders for Ethereum or Polygon style addresses.

Ignoring data formatting requirements

✗ AVOID

A user asks to 'show me the balance changes' without specifying if they want current holdings or a time range.

✓ INSTEAD

Be precise. Use get_evm_historical_balances when you need to track *changes* over time, not just what the balance is right now.

The Right Fit

Use this MCP if your primary data source must be verifiable, real-time blockchain ledger activity. You need to know exactly how many tokens moved (get_evm_transfers) or who owns them (get_svm_holders). This is non-negotiable for DeFi research or smart contract auditing.

However, don't use this if your data comes from an internal database or a centralized exchange report. For that, you need tools connecting to APIs like Stripe or Salesforce.

Also, don't try to pull general market sentiment; the MCP only tracks transactions. If you just want to know 'what people *think* about' a token, you need a natural language processing tool, not one of these transaction-based indexers.

Tracking Cross-Chain Asset Flow is a Nightmare

Right now, if you want to know the full picture of an asset's movement, you have to open multiple tabs. You run one query for Ethereum swaps on Etherscan, then another set of tools for Polygon transfers, and finally switch over to Solscan just to see Solana holders. It’s tedious copy-pasting between different data sets.

With this MCP, your agent handles that complexity automatically. You ask it once: 'Show me all the swaps.' It compiles the necessary reads from EVM chains like Ethereum *and* Solana in one go. What you get is a single, consolidated narrative of asset movement.

---

get_evm_swaps and get_svm_swaps Give You Clarity

Before this, comparing swap volume across different chains meant running separate reports and manually reconciling the time stamps. You were always comparing apples to oranges because of the manual effort involved.

Now, your agent pulls get_evm_swaps for one chain and get_svm_swaps for another, side-by-side. It treats them as comparable data streams, letting you compare activity volume without ever leaving the chat interface.

The Graph (Web3 Indexing) - 8 Tools

Use these tools to pull verifiable blockchain data, including token movements, asset ownership records, and complex swap activity from both EVM and Solana chains.

#	TOOL	DESCRIPTION
01	`get_evm_historical_balances`	Retrieves the balance changes for a specific address on an EVM-compatible chain over time.
02	`get_evm_holders`	Gets a list of all token holders and their respective balances on any EVM chain.
03	`get_evm_swaps`	Retrieves records detailing DEX swaps that occurred on an EVM-compatible chain.
04	`get_evm_transfers`	Gets a comprehensive list of all token transfers for a given asset on an EVM chain.
05	`query_subgraph`	Allows you to run custom GraphQL queries against any specific subgraph ID using your API key.
06	`get_svm_holders`	Retrieves a list of token holders and their balances on the Solana (SVM) network.
07	`get_svm_swaps`	Gets records detailing DEX swaps that occurred specifically on the Solana network.
08	`get_svm_transfers`	Retrieves a comprehensive list of all token transfers for a given asset on the Solana (SVM) network.

See It in Action

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

U Get the top 10 holders for the EVM token at 0x7fc66500c84a76ad7e9c93437bfc5ac33e2ddae9.

I've retrieved the top holders for the specified token. The largest holder currently controls 12.4% of the supply, followed by a major exchange wallet with 8.2%. Would you like to see the full list of addresses and balances?

U Show me recent swaps for the Solana token DezXAZ8z7PnrnMcFWRSTQC8PGL9P8be3EFr9SptT5v7.

I found 15 recent swap events for that token on Solana. The most significant was a 500 SOL trade on Raydium. Most activity is currently concentrated on Jupiter. Should I analyze the price impact of these trades?

U Query the Uniswap V3 subgraph with ID 'ELUvFp... ' to get the latest pool prices.

Executing GraphQL query on the Uniswap V3 subgraph... I've successfully fetched the latest price data. The ETH/USDC pool is currently trading at $2,450.32 with $12M in 24h volume. Do you need details on any other specific pool?

Frequently Asked Questions

01 **Can The Graph (Web3 Indexing) MCP handle both Ethereum and Solana transactions?**

Yes, it handles both. It provides distinct tools for EVM chains (like get_evm_transfers) and specific tools for the Solana network (like get_svm_transfers), ensuring you get the right data source.

02 **What is the difference between getting_evm_holders and get_evm_transfers?**

get_evm_holders gives you a static list of who owns assets right now. get_evm_transfers provides an auditable history, showing every single movement that occurred over time.

03 **Do I need to know GraphQL when using query_subgraph?**

While the tool uses GraphQL internally, you don't need to be a developer. You just provide the specific subgraph ID and your agent executes the complex request for you.

04 **Does The Graph (Web3 Indexing) MCP track NFTs?**

Yes, it handles token data, which includes tracking unique assets. You can use various transfer tools to monitor the movement of NFT tokens across both EVM and Solana networks.

05 **How do I check historical performance with get_evm_historical_balances?**

You provide the specific token address and the desired time window. The tool returns a formatted, structured record of balance changes over that period, perfect for charting.

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 `"the-graph-web3-indexing": { "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

The Graph (Web3 Indexing) 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 The Graph (Web3 Indexing). 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	The Graph (Web3 Indexing) MCP
Server ID	`019e38fa-7e64-7055-b4af-fff8a407ee1d`
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/the-graph-web3-indexing.