MCP SERVER | NO CODE | CLOUD HOSTED

Etherscan MCP

Audit every token movement and smart contract interaction.

Etherscan MCP connects your AI agent directly to live on-chain data for Ethereum and all EVM-compatible networks. You can instantly track native token balances, view transaction history, follow specific ERC-20 or NFT movements, and even pull smart contract source code—all through natural conversation.

F Quality Score 43.65/100

ethereum | evm | block-explorer | wallet-tracker | smart-contracts

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

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.

Etherscan MCP

19 tools available
Cloud-hosted on Vinkius

Stop opening ten different browser tabs just to audit a single wallet address. This MCP connects your AI agent directly to Etherscan's massive database, giving you instant access to every piece of on-chain data across Ethereum and other EVM chains. You can ask your agent for the native token balance of multiple addresses or pull detailed records of both normal and internal transactions without ever touching a block explorer interface. If you need to know exactly how many times an NFT moved or track complex token transfers, specialized tools handle ERC-20, ERC-721, and ERC-1155 movements instantly. Because this MCP is hosted on Vinkius, your agent can manage all these diverse data points from one connection point, allowing you to audit wallets and trace funds with simple conversation prompts.

Core Capabilities

01 — Check Wallet Balances

Retrieve the native token balance for a single address or check multiple addresses at once.

02 — Audit Transaction History

Fetch detailed lists of both normal and internal transactions associated with any wallet.

03 — Monitor Token Transfers

Track specific movements for different token standards, including ERC-20 tokens, NFTs (ERC-721), and multi-standard assets (ERC-1155).

04 — Analyze Contract Code

Pull the complete source code or the Application Binary Interface (ABI) for any given smart contract.

05 — Access Chain Metrics

Get real-time data points like the current gas price, total supply, and the latest block number.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius and provide your personal Etherscan API key.

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

03 Ask your agent a question, such as 'What were all the token transfers for address X?' Your agent executes the necessary tool call and provides the data.

The bottom line is you talk to your agent like talking to a person who already has access to the blockchain's entire transaction history.

Built For

This MCP is for crypto analysts and developers who spend too much time clicking through block explorers. If your job requires validating funds, tracing asset movements, or verifying contract logic, this tool saves hours of manual data scraping.

Crypto Analyst

Tracks large wallet movements and token distributions to identify market trends without manually reviewing thousands of transaction hashes.

Web3 Developer

Debugs smart contract interactions or verifies transaction statuses directly inside their IDE, pulling the necessary ABI or source code on demand.

DeFi Auditor

Monitors portfolio balances and checks complex internal transactions to ensure assets moved correctly across different protocols.

What Changes When You Connect

01 Trace complex funds movements easily. Instead of manually checking logs, you can use specialized tools like `get_token_tx` to pull the entire ERC-20 transfer history for any address in a single prompt.

02 Save time debugging contracts. Use `get_abi` or `get_source_code` to instantly grab the contract's logic right from your chat interface, making it ideal for Web3 developers working in their IDE.

03 Understand deep transactions. You don't just see simple transfers; you can use `get_tx_list_internal` to expose the complex, underlying interactions that happen between contracts.

04 Get a full financial picture. By combining tools like `get_balance_multi` and `get_address_token_balance`, your agent gives you an instant portfolio valuation across several wallets at once.

05 Verify data accuracy instantly. Need proof of code? The `verify_source_code` tool confirms that the source code is correctly published, removing guesswork from contract audits.

Real-World Applications

Investigating a Suspicious Transfer

A DeFi auditor notices an unusual outflow. They ask their agent to pull the full transaction record using `get_transaction_by_hash`, then follow up by requesting all internal transactions via `get_tx_list_internal` to determine which smart contract received the funds.

Tracking NFT Royalties

A collector wants to track a specific piece of digital art. They prompt their agent for the ERC-721 transfer history using `get_token_nft_tx` across multiple addresses, confirming every time the asset moved and who received it.

Checking Multi-Wallet Health

A development team needs to check if three different test wallets are funded. Instead of running three separate checks, they use `get_balance_multi` to get all native token balances in one query.

Debugging a Contract Failure

A developer encounters an error. They first run `get_source_code` on the contract address to review the code, and then ask for event logs using `get_logs` to pinpoint exactly where the failure occurred.

Patterns to Avoid

Using it for market news

✗ AVOID

Asking the agent: 'What's the general sentiment about Ethereum right now?' The agent will only provide data, not subjective analysis.

✓ INSTEAD

Use this MCP to get objective data points like the current gas cost using `get_gas_oracle` or the total circulating supply using `get_eth_supply`. It reports facts, never opinions.

Searching by vague keywords

✗ AVOID

Prompting: 'Show me all transactions that were weird.' The agent has no way to interpret 'weird' and will fail or return useless data.

✓ INSTEAD

Be specific. Use `get_tx_list` and specify the address, timeframe, and transaction type you want to investigate.

Ignoring token standards

✗ AVOID

Asking for all transfers when only ERC-20 tokens are involved. This misses critical NFT movements.

✓ INSTEAD

Use targeted tools. If it's an NFT, use `get_token_nft_tx`. If it's a standard token, use `get_token_tx`.

The Right Fit

Use this MCP if your primary need is absolute verification of state on the blockchain. You want to know: 'Did X move to Y? What code governed that movement? How much was left at Z?' This tool excels when you need definitive, auditable data points like checking a wallet's balance using `get_balance` or confirming contract logic by pulling the source code via `get_source_code`. Don't use it if you are trying to gauge market sentiment, predict future price movements,

or analyze general news flow. For that, you need a different type of data analysis tool; this is purely for ledger verification.

Manually tracking funds across chains is painful work.

Today, figuring out where funds went requires opening the block explorer in a browser, searching by address or hash, and then manually clicking through tabs for token transfers, internal calls, and basic balances. You end up copy-pasting data from one source into another just to get a full picture of what happened.

With this MCP, your agent does all that heavy lifting. You simply ask the question—'Where did these funds go?' The agent runs multiple specialized checks, pulls the necessary transaction lists, and gives you the complete history right back in plain language.

Get full on-chain visibility with Etherscan.

You no longer have to juggle multiple API calls or piece together data from different pages. Whether you need the contract's source code for debugging, a list of every single ERC-721 movement, or just the current gas cost, it's all handled by targeted tools like `get_abi` and `get_gas_oracle`.

The result is immediate clarity. You stop guessing about blockchain activity and start acting on verifiable data.

Etherscan: 19 Blockchain Data Retrieval Tools

Use these tools to query specific data points across the blockchain, including balances, transaction logs, token movements, and smart contract details.

#	TOOL	DESCRIPTION
01	`get_address_token_balance`	Determines the total portfolio value of a specific address holding multiple types of tokens.
02	`get_balance_multi`	Checks the native token balance for several specified addresses simultaneously.
03	`get_balance`	Retrieves the current native token balance for a single address.
04	`get_abi`	Pulls the Application Binary Interface (ABI) used by a specific smart contract.
05	`get_address_tag`	Looks up a common name or tag associated with a wallet address.
06	`get_block_no_by_time`	Finds the specific blockchain block number that corresponds to a given timestamp.
07	`get_block_number`	Retrieves the current, latest block number on the chain.
08	`get_eth_price`	Gets the most recent market price for Ether (ETH).
09	`get_eth_supply`	Retrieves the total circulating supply of Ether.
10	`get_gas_oracle`	Provides up-to-date information regarding gas costs for transactions.
11	`get_logs`	Fetches event logs generated by smart contract interactions.
12	`get_source_code`	Retrieves the human-readable source code for a given smart contract address.
13	`get_transaction_by_hash`	Looks up all details about a single transaction using its unique hash identifier.
14	`get_token_1155_tx`	Lists the movement history for ERC-1155 multi-standard tokens related to an address.
15	`get_token_nft_tx`	Provides a list of transfers specifically involving Non-Fungible Tokens (ERC-721) for an address.
16	`get_token_tx`	Retrieves the transaction history for standard ERC-20 tokens related to an address.

#	TOOL	DESCRIPTION
17	get_tx_list_internal	Shows all internal transactions, which detail contract interactions that happened behind the scenes at an address.
18	get_tx_list	Retrieves a list of standard, visible transactions for an address up to 10,000 records.
19	verify_source_code	Confirms that the contract's publicly available source code matches what is recorded on the blockchain.

See It in Action

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

U What is the ETH balance of address 0xde0B295669a9FD93d5F28D9Ec85E40f4cb697BAe on Ethereum?

I've checked the balance for that address on Chain ID 1. The current balance is approximately 425.12 ETH (425123456789012345678 wei).

U Show me the last 5 ERC-20 token transfers for 0x742d35Cc6634C0532925a3b844Bc454e4438f44e.

I've retrieved the ERC-20 transfer history. The most recent transfers include 500 USDT from 0xabc... and 100 LINK to 0xdef... Would you like more details on any of these transactions?

U List the normal transactions for address 0x123... on Polygon (Chain ID 137).

Querying Polygon scan... I found 10 recent transactions for that address. The latest was a contract interaction with Uniswap V3. Would you like me to list the specific transaction hashes?

Frequently Asked Questions

01 **Can Etherscan MCP track token transfers for non-ERC-20 tokens?**

Yes, it handles specific standards like ERC-721 (NFTs) using the `get_token_nft_tx` tool, and also multi-standard assets via `get_token_1155_tx`.

02 **How do I check balances for multiple addresses with Etherscan MCP?**

You use the `get_balance_multi` tool. This allows you to pass up to 20 different wallet addresses and get all their native token balances in one single API call.

03 **What is the difference between normal and internal transactions?**

Normal transactions are the visible, standard transfers. Internal transactions, retrieved with `get_tx_list_internal`, show the complex communication that happens *between* smart contracts.

04 **Does Etherscan MCP only work on Ethereum?**

No, this MCP supports multiple chains by allowing you to specify a chain ID. This lets you query data across different EVM-compatible networks.

05 **Can I get the contract code using the Etherscan MCP?**

Yes, use the `get_source_code` tool to pull the human-readable source code for any smart contract address you provide.

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 `"etherscan": { "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

Etherscan 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 Etherscan. 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	Etherscan MCP
Server ID	`019e3892-bdf5-725d-88d9-3d230d1783ab`
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/etherscan.