MCP SERVER | NO CODE | CLOUD HOSTED

Alchemy MCP for AI Agents

Analyze wallet assets and audit blockchain transactions by address

Alchemy MCP lets your AI client audit and analyze any blockchain data—from wallet balances to NFT ownership and complex transaction histories. Instead of navigating block explorers, you ask your agent questions, and it instantly queries Ethereum and compatible networks for real-time asset intelligence.

A+ Quality Score 100/100

web3 | rpc-querying | nft-data | wallet-tracking | blockchain-infrastructure | transaction-history

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

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.

Alchemy MCP

6 tools available
Cloud-hosted on Vinkius

Blockchain analysis used to mean endless clicking through web interfaces. You'd copy addresses, paste them into a dozen different tabs, and piece together the puzzle yourself. This MCP changes that. It lets you talk to your data instead of digging through it.

With this connector, your AI agent acts as an instant Web3 analyst. You can ask complex questions—like 'Show me all assets owned by these addresses' or 'Did this specific smart contract interaction succeed?'—and get immediate, actionable answers. Your agent pulls real-time data on native currency holdings, ERC-20 tokens, and full NFT portfolios across multiple networks.

When you connect to the Vinkius catalog, your AI client gains access to this intelligence layer. You don't need to know RPC querying; you just need to ask a question. It's about making blockchain data conversational.

Core Capabilities

01 — Check Token Holdings

Retrieves the native currency balance and all specific ERC-20 token balances for any given Ethereum address.

02 — Audit NFT Collections

Scans a specified wallet to list every unique digital asset (NFT) it owns, including collection metadata.

03 — Review Transaction History

Pulls the detailed receipt for any transaction hash, showing its final status and outcome on the chain.

04 — Track Latest Network Status

Gets the most recent block number to confirm the current state of the blockchain network.

05 — Monitor Service Health

Verifies whether the Alchemy service is currently operational and ready for querying.

One Click on Vinkius — From Prompt to Execution

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

01 Start by connecting your preferred AI client to this MCP within Vinkius.

02 Provide a clear prompt, asking your agent to perform a specific blockchain task (e.g., 'What are the token balances for address X?').

03 The agent executes the necessary queries and returns the structured data directly in your chat interface.

The bottom line is you get real-time, deep web3 insights without manually querying any block explorer or API endpoint.

Built For

This MCP is for the crypto analyst who spends hours cross-referencing data points. It's built for security researchers needing rapid audit reports and developers building complex DeFi tools that need reliable, immediate on-chain data.

DeFi Analyst

Checks if a multi-step smart contract interaction succeeded or failed by running `get_transaction_receipt` against a specific transaction hash.

Crypto Investor

Quickly audits an opponent's wallet to see their total asset exposure, combining calls to `get_wallet_balance`, `get_token_balances`, and `get_owned_nfts`.

Security Researcher

Performs rapid due diligence by retrieving the latest block number (`get_latest_block_number`) before attempting to audit assets or contracts.

What Changes When You Connect

01 Audit NFT collections instantly. Instead of browsing a collection page, you can use `get_owned_nfts` to list every digital asset owned by an address in seconds.

02 Consolidate all token data. Combining the results from `get_wallet_balance` and `get_token_balances` gives you a single view of total crypto exposure for any wallet.

03 Understand transaction outcomes. Use `get_transaction_receipt` to verify if a complex smart contract call completed successfully or failed midway, saving hours of manual investigation.

04 Stay current on the network. Running `get_latest_block_number` confirms you are operating with the most up-to-date data available across supported networks.

05 Reliable data access. The ability to check API status first ensures that your deep research workflow doesn't fail because of a temporary service outage.

Real-World Applications

Investigating suspicious funds transfers

A security analyst wants to know how funds moved. They ask their agent, and it uses `get_transaction_receipt` to pull the full history of a transaction hash, revealing exactly which contracts were called and if any part failed.

Valuing an NFT portfolio

A client needs an asset summary. The agent runs three commands: `get_owned_nfts` to list the assets, then checks the overall wallet balance using `get_wallet_balance`, giving a complete picture of the user's wealth.

Checking DeFi protocol status

A developer building a monitoring tool needs confirmation that all necessary data is available. They call `check_api_status` first, ensuring their automated workflow runs against an active endpoint before querying balances or transactions.

Mapping token movements across addresses

A researcher wants to track a specific asset type. The agent uses `get_token_balances` for multiple related addresses in one query, mapping the flow of ERC-20 tokens and identifying potential leakage points.

Patterns to Avoid

Treating blockchain as a simple database

✗ AVOID

Assuming that just because an address exists, you can immediately find all its balances without specifying the token type or network. This leads to incomplete data sets.

✓ INSTEAD

Always use `get_token_balances` alongside `get_wallet_balance` to get a comprehensive view of native and specific ERC-20 assets for total asset accountability.

Ignoring transaction failure details

✗ AVOID

Seeing that a transaction was initiated but assuming it succeeded. This is dangerous because many smart contract calls fail silently or partially.

✓ INSTEAD

You must run `get_transaction_receipt` to verify the final status code and detailed outcome of the event, confirming if the action actually completed on-chain.

Overlooking network timing

✗ AVOID

Running a time-sensitive analysis without knowing the absolute latest state of the blockchain. This means missing critical data points.

✓ INSTEAD

Start by calling `get_latest_block_number` to anchor your research in the most current, verified block height before proceeding with any asset or transaction audit.

The Right Fit

Use this MCP if your core task involves deep financial analysis of decentralized assets. You need to verify token balances (`get_token_balances`), track NFTs (`get_owned_nfts`), and review the immutable history provided by `get_transaction_receipt` . Don't use it if you are simply writing basic code or managing simple records that don't involve a public blockchain ledger—those tasks require standard database connectors. If your goal is merely to check if an API endpoint is up, while `check_api_status` works, remember this MCP is designed for data retrieval, not pure uptime monitoring.

Alchemy MCP: Auditing Web3 Asset Ownership and NFT Data

Today, auditing an entire digital portfolio means opening multiple block explorers. You copy the address, check the ETH balance, then switch tabs to query ERC-20 tokens, and finally perform a separate search just for owned NFTs. It's tedious, slow, and easy to miss crucial details.

With this MCP, you simply ask your agent: 'Show me everything related to this wallet.' The system combines calls like `get_wallet_balance`, `get_token_balances`, and `get_owned_nfts` into one coherent response. You get a single source of truth about the assets without leaving your chat window.

Alchemy MCP: Analyzing Blockchain Transaction History

Manually tracing transaction failures is a nightmare. You have to find the hash, then read through all the raw data to determine *why* it failed—was it an insufficient gas payment? Did the smart contract reject the call? This manual process eats up hours.

Now you pass the transaction hash directly to your agent. It uses `get_transaction_receipt` and delivers a clean, human-readable breakdown of the outcome, showing exactly what happened when the funds moved or the contract executed.

Alchemy: 6 Tools for Blockchain Data Access

These tools let your agent query native balances, list NFTs, check token holdings, and review transaction receipts directly from the chain.

#	TOOL	DESCRIPTION
01	`check_api_status`	Confirms if the Alchemy service connection is currently working and operational for querying data.
02	`get_wallet_balance`	Retrieves the current native currency balance (in wei) for any specified Ethereum address.
03	`get_latest_block_number`	Returns the number of the most recent block published on the network.
04	`get_owned_nfts`	Lists all Non-Fungible Tokens owned by a specific wallet address.
05	`get_token_balances`	Gets the balance of every ERC-20 token associated with an address.
06	`get_transaction_receipt`	Provides a full, detailed receipt for any specific transaction using its unique hash.

See It in Action

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

U What's the total value of all assets for '0xabc...'?

💰 **Wallet Summary Report**

- **ETH Balance:** 12.5 ETH (Current Value: $4,875)
- **ERC-20 Tokens:**

* USDC: 5,000 USDC ($5,000)
* AAVE: 20 AAVE ($320)

- **NFTs Owned:** 6 (Total Collection Value Estimate: $15,000+)

Analysis Complete. The wallet holds a strong diversified portfolio across native tokens and major assets.

U Show me all the NFTs owned by this address.

💎 **NFT Portfolio Dump**

Found 7 unique digital assets:

- 🖼️ Art Blocks Genesis #14 (Metadata Link: /art/genesis-14)
- 🐒 Bored Ape Yacht Club #9021
- 👾 CryptoPunk #5678
- ... and 4 others.

Need the full metadata for any of these collections? Just ask.

U What happened with transaction hash 0x...?

✅ **Transaction Audit: Success**

DETAIL	VALUE
Status:	Successful
From:	0xabc...
To:	smart-contract-xyz.eth
Value Transferred:	5 ETH
Gas Used:	120,000 units

The transaction successfully executed the contract logic and transferred funds as intended.

Frequently Asked Questions

01 **How does the Alchemy MCP help me track my crypto holdings?**

It provides a single, comprehensive view of all your digital assets. You don't just see ETH; you get separate balances for every ERC-20 token and a list of owned NFTs, giving you a true picture of your portfolio value.

02 **Can I use the Alchemy MCP to audit smart contract interactions?**

Yes. By querying transaction receipts using this MCP, you can review the full execution details of any smart contract call. You'll see if it succeeded, failed, or transferred funds as expected.

03 **I need to know the latest blockchain state; how does Alchemy help?**

You simply ask your agent for the latest block number. This confirms that all data you are analyzing is based on the most current and validated state of the network, preventing stale research.

04 **What if I need to check balances across multiple addresses?**

The MCP lets your agent handle multiple address lookups in a single workflow. You can ask it to audit several wallets at once to compare asset distributions or track ownership changes across different parties.

05 Is this suitable for developing DeFi monitoring tools?

Absolutely. It gives developers reliable, real-time access to core blockchain data—from token movements to transaction history—making it perfect for building robust automation.

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 `"alchemy": { "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

Alchemy 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 Alchemy. 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	Alchemy MCP
Server ID	`019d8414-1484-7106-b0bb-36a43dc3565d`
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/alchemy.