MCP SERVER | NO CODE | CLOUD HOSTED

Fireblocks MCP

Control complex digital asset movements from your agent.

Fireblocks MCP lets your AI agent run complex digital asset operations across multiple blockchains. You can manage vault accounts, check balances in hundreds of vaults instantly, and execute transactions while adhering to strict compliance rules—all via natural conversation. It's built for high-value treasury management.

A+ Quality Score 98.33/100

digital-assets | vault-management | crypto-transactions | institutional-custody | blockchain-api

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

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.

Fireblocks MCP

40 tools available
Cloud-hosted on Vinkius

This connector hooks up your Fireblocks workspace directly to your AI client, letting you handle institutional-level digital asset movements using just plain language. Instead of jumping between multiple dashboards and manually inputting data for every step, you tell your agent what you need done. You can ask it to list all owned NFTs or check the balance of a specific vault account across different assets. Need to move funds? Your agent handles everything from checking network fees to creating the necessary transaction request, even simulating compliance checks like validating travel rules before submission. Connecting Fireblocks through Vinkius means your entire team connects once and gets access to this powerful financial toolset alongside thousands of others.

Core Capabilities

01 — Manage Wallet Containers

Create, list, or inspect both internal and external wallet accounts.

02 — Execute Complex Transactions

Initiate various movements, such as transfers, mints, or burns, across different blockchains.

03 — Monitor Asset Positions and Balances

Get the current balance of a specific asset within any managed vault account, or list all supported assets for review.

04 — Secure Compliance Checks

Run required compliance validations on transactions, like checking travel rules or retrieving screening data.

05 — Automate Data Collection

Pull transaction history, list supported blockchains, or get details for specific contract assets.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Fireblocks API Key and Private Key credentials.

02 Your AI agent validates the connection and establishes access permissions within the Fireblocks platform.

03 You interact with the system by asking natural language questions, guiding your agent to perform tasks like checking balances or creating a transaction.

The bottom line is you use natural conversation to manage complex digital asset operations without needing to touch a single dashboard.

Built For

This MCP is for treasury managers and compliance officers who deal with large volumes of assets across multiple chains. It solves the pain point of having to manually switch between specialized dashboards just to get a full picture of asset movement or execute a single transfer.

Treasury Manager

You use this MCP to instantly check balances across dozens of vaults and initiate transfers without clicking through multiple account pages.

Compliance Officer

You run compliance validations, like checking travel rules or retrieving transaction screening results, before any funds can move.

DevOps Engineer

You automate the creation of new vault accounts and necessary deposit addresses as part of a deployment pipeline.

What Changes When You Connect

01 Cut down on manual checking. Instead of running separate queries to check balances, you can ask your agent to get_vault_account_asset and instantly know the status of any asset across hundreds of vaults.

02 Manage risk before it happens. You don't just send funds; your agent first uses estimate_fee or estimate_network_fee so you know the total cost upfront, preventing overspending.

03 Simplify setup. Creating a new account is simple. You can tell your agent to create_vault_account and then immediately ask it to create_external_wallet for the necessary holding spot.

04 Ensure compliance every time. Before any transaction goes out, you run validate_travel_rule through your agent, guaranteeing adherence to regulatory requirements before submission.

05 Audit easily. Instead of manually digging through logs, your agent can list_transactions or get_transaction details so you have a complete history ready for review.

Real-World Applications

Executing an International Fund Transfer

A treasury manager needs to send 50 ETH from the main vault, but they must pass compliance. They ask their agent to validate_travel_rule first. The agent runs the check, confirms it's compliant, estimates_fee for the transaction, and finally uses create_transaction to submit everything in one flow.

Automating New Asset Onboarding

A DevOps engineer needs a new testing wallet. They ask their agent to create_vault_account first. Then they instruct it to create_internal_wallet and generate the necessary deposit address using create_vault_account_asset_address, automating deployment steps.

Investigating Missing Funds

A team member spots a discrepancy in the asset balance. They ask their agent to get_vault_account and then specify which assets they want details on using get_vault_account_asset, pinpointing exactly where the funds are.

Handling Staking Requirements

A portfolio manager wants to start earning yield. They ask their agent to list_staking_chains first, then use stake_position to initiate a new staking position without leaving the chat interface.

Patterns to Avoid

Checking balances manually

✗ AVOID

The user has to open the main dashboard, find Vault A's balance; then switch tabs to check Vault B, and finally run a separate query for NFT holdings.

✓ INSTEAD

Instead, just ask your agent to get_vault_account_asset or list_owned_nfts. It aggregates all necessary balance information across multiple vaults in one prompt.

Forgetting compliance checks

✗ AVOID

A user quickly creates a transaction using create_transaction without realizing the destination country requires travel rule validation.

✓ INSTEAD

Always include validate_travel_rule when submitting funds. Your agent will prompt you for the necessary screening results before finalizing the transfer.

Confusing asset types

✗ AVOID

A user tries to find the balance of a custom token, but only knows the name and not the contract ID.

✓ INSTEAD

First, use list_assets or get_contract_asset to confirm the correct asset type. Then you can correctly reference it when using create_vault_account_asset.

The Right Fit

Use this MCP if your job involves moving, tracking, or auditing digital assets across multiple chains and requires high compliance control. You need a single interface to manage everything from account creation (create_vault_account) to fund movement (create_transaction). Don't use it if you just need to read simple public blockchain data; for that, a general blockchain explorer tool is fine. However, if your goal involves setting up automated workflows or complex asset registration (register_asset), this MCP provides the

necessary depth and control over internal/external wallet management.

Handling crypto transfers used to mean opening three different dashboards.

Today, if you need to move assets across your organization's vaults, you typically have to log into the main dashboard. Then, you switch tabs to check balances in Vault A versus Vault B. If you're moving a custom token or an NFT, you might run a separate query just for that asset type.

With this MCP, all of that complexity disappears. You simply ask your agent to create_transaction, specifying the source and destination. The system handles checking required assets (like listing_assets) and ensures the full lifecycle is tracked from start to finish.

Fireblocks provides comprehensive control over your digital asset infrastructure.

Manual workflows often fail when compliance changes or new assets are introduced. You waste time manually validating travel rules and updating gas station settings whenever the network environment shifts.

Now, you can integrate validation checks like validate_travel_rule directly into your workflow. This gives you real-time control over every single transaction before it ever leaves the vault.

Fireblocks: 30 Tools for Asset & Vault Management

These tools give you direct access to every core function of the Fireblocks platform. Use them to create wallets, check balances, manage contracts, and process complex blockchain transactions.

#	TOOL	DESCRIPTION
01	`add_asset_to_wallet`	Attaches an asset or address to an existing wallet container.
02	`add_contract`	Adds a specific contract identifier to the system's whitelist.
03	`cancel_transaction`	Stops a transaction that is currently waiting for confirmation.
04	`create_external_wallet`	Sets up a new wallet container that resides outside the Fireblocks system.
05	`create_internal_wallet`	Creates a new, controlled wallet container within the Fireblocks environment.
06	`create_transaction`	Builds and submits various types of transactions, including transfers and minting operations.
07	`create_vault_account_asset_address`	Generates a unique deposit address for a specific asset within a vault account.
08	`create_vault_account_asset`	Establishes a new dedicated wallet space for an asset inside a vault account.
09	`create_vault_account`	Creates an entirely new, isolated vault account container.
10	`create_webhook`	Sets up a new webhook to notify external systems of events.
11	`drop_transaction`	Submits an ETH or EVM transaction directly to the network.
12	`estimate_fee`	Predicts the cost associated with a potential transaction before it's sent.
13	`estimate_network_fee`	Retrieves current network fee estimates for a specific digital asset.
14	`get_contract_asset`	Fetches detailed information about a registered contract asset.

#	TOOL	DESCRIPTION
15	`get_gas_station`	Retrieves the current settings for gas fee management thresholds.
16	`get_public_key_info`	Gets the public key associated with a specific derivation path.
17	`get_screening_transaction`	Retrieves compliance screening results for an attempted transaction.
18	`get_transaction`	Fetches all details about a specific, previously submitted transaction.
19	`get_vault_account_asset`	Checks the available balance of a specific asset held within a vault account.
20	`get_vault_account`	Pulls detailed information about an entire vault account container.
21	`list_assets`	Shows every digital asset type that the system supports tracking.
22	`list_blockchains`	Lists all supported blockchain networks for transactions.
23	`list_contracts`	Retrieves a list of whitelisted and recognized smart contracts.
24	`list_external_wallets`	Shows all external wallet containers connected to the system.
25	`list_internal_wallets`	Displays all internal wallet containers managed by Fireblocks.
26	`list_owned_nfts`	Retrieves a list of NFT tokens that the vault owns.
27	`list_staking_chains`	Lists all supported networks for staking positions.
28	`list_staking_positions`	Shows active staking positions and their current status.
29	`list_transactions`	Retrieves a filtered history of all past transactions.
30	`list_vasps`	Displays information about connected Virtual Asset Service Providers (VASPs).
31	`list_vault_account_asset_addresses`	Lists all available deposit addresses for a specific asset within a vault.
32	`list_vault_accounts`	Retrieves a paginated list of all existing vault accounts.
33	`list_webhook_notifications`	Shows notifications that were sent out via webhooks.

#	TOOL	DESCRIPTION
34	refresh_nfts	Updates the displayed data for NFT tokens within a vault account to ensure accuracy.
35	register_asset	Allows you to register and track custom asset types, like specific ERC-20 tokens.
36	resend_webhooks	Resends webhooks that failed delivery previously.
37	stake_position	Initiates a new staking position on a selected chain.
38	update_gas_station	Modifies the automatic fueling thresholds for gas costs.
39	update_vault_account	Changes the name of an existing vault account container.
40	validate_travel_rule	Checks a transaction against international travel rule compliance requirements.

See It in Action

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

U List the first 10 vault accounts in my Fireblocks workspace.

I've retrieved your vault accounts. You have 'Main Vault' (ID: 0), 'Staking-ETH' (ID: 5), and 'Trading-Hot' (ID: 12) among others. Would you like to see the asset balances for any of these?

U What is the BTC balance in vault account ID 0?

In vault account 0 (Main Vault), the BTC balance is 1.45 BTC available, with a total balance of 1.50 BTC (0.05 BTC pending).

U Create a transaction to send 0.1 ETH from vault 0 to external wallet 'Exchange-Wallet'.

I've initiated the transaction request. Transaction ID: tx-98765. The status is currently 'SUBMITTED' and is awaiting approval in your Fireblocks console.

Frequently Asked Questions

01 **How do I check my balance using Fireblocks MCP?**

You ask your agent to get_vault_account_asset. You provide the specific asset and vault account ID, and it returns the current available balance.

02 **Can Fireblocks MCP handle compliance rules?**

Yes. You use validate_travel_rule before submitting any funds to ensure the transaction meets international regulatory requirements.

03 **What's the difference between create_internal_wallet and create_external_wallet in Fireblocks MCP?**

Internal wallets are controlled within the main vault system, while external wallets are separate containers that still need to be managed through the MCP.

04 **Do I need to list all_blockchains before creating a transaction with Fireblocks MCP?**

While you can use list_blockchains to see options, your agent handles compatibility. You just tell it the source and destination chains for create_transaction.

05 **How do I get NFT details using Fireblocks MCP?**

You first run list_owned_nfts to see what you hold, then use refresh_nfts if you suspect the data might be outdated before making any decisions.

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 `"fireblocks": { "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

Fireblocks 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 Fireblocks. 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	Fireblocks MCP
Server ID	`019e3897-adcf-7021-9dc2-a04a68dbe290`
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/fireblocks.