MCP SERVER | NO CODE | CLOUD HOSTED

Biconomy (Account Abstraction) MCP for AI Agents

Orchestrate Complex Smart Contract Transactions and Manage Gasless Payments

Biconomy Account Abstraction lets your AI agent manage complex, multi-step Web3 operations using supertransactions and account abstraction. It handles everything from generating optimized gas quotes for sponsored transactions to executing fully signed payloads on any compatible blockchain.

A+ Quality Score 100/100

account-abstraction | web3 | smart-accounts | gasless-transactions | erc-4337

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/biconomy-account-abstraction

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.

Biconomy (Account Abstraction) MCP

3 tools available
Cloud-hosted on Vinkius

Running decentralized applications often means dealing with complicated smart contract interactions, managing gas fees, and ensuring transaction integrity across multiple steps. This MCP lets your AI agent bypass those manual hurdles entirely. Instead of writing brittle scripts for every payment or flow, you describe the desired outcome in natural language. The system then calculates optimal execution paths—sometimes even making transactions 'gasless' through sponsorship options. You can monitor these complex flows and execute them all from one place. By connecting Biconomy via Vinkius, your agent gains the power to orchestrate entire Web3 intents, allowing you to focus on product logic instead of blockchain plumbing.

Core Capabilities

01 — Get Supertransaction Quotes

Calculate optimized cost estimates for complex transaction flows, including gasless options.

02 — Execute Signed Transactions

Submit fully signed payloads to trigger the execution of a supertransaction on-chain.

03 — Track Transaction Status

Monitor the real-time progress and final confirmation status of an executed supertransaction hash.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/biconomy-account-abstraction** — connect your AI agent in three steps.

01 Subscribe to this MCP on Vinkius and supply your Biconomy API Key.

02 Instruct your AI client or agent with the desired complex blockchain action (e.g., 'Execute a payment of X amount for Y recipient').

03 The system first uses the quoting tool, then submits the payload using the execution tool, providing real-time status updates via the tracking tool.

The bottom line is that your AI agent handles the entire lifecycle of complex Web3 transactions—from initial quote to final on-chain confirmation—without you needing to write custom scripts.

Built For

This MCP is built for people who live in the intersection of software engineering and decentralized finance. If your job involves scripting interactions with smart contracts, gas optimization, or managing complex payment flows, this tool saves you from manual setup.

DeFi Power User

Runs multi-step transaction sequences (e.g., staking and harvesting rewards) that require precise timing and optimized gas usage.

Web3 Developer

Rapidly prototypes and tests complex, real-world transaction flows without manually writing boilerplate smart contract interaction logic.

dApp Operator

Automates maintenance or user operations that involve triggering time-sensitive on-chain events using natural language instructions.

What Changes When You Connect

01 Eliminate scripting headaches: Use the `get_quote` tool to calculate optimal transaction costs, including gas-sponsored (gasless) routes, before committing funds.

02 Handle multi-step logic easily: The MCP allows agents to manage complex flows—like initial quote generation followed by execution—all within a single conversation thread.

03 Confirm success immediately: After executing the supertransaction with `execute_supertx`, you can use `get_explorer_status` to track its progress until final confirmation.

04 Support diverse accounts: It natively supports Smart Accounts, EOA, and EOA-7702 account modes, making it versatile for different types of web3 deployments.

05 Improve transaction efficiency: By abstracting complex signing requirements, you reduce the manual effort required to manage gas costs across various protocols.

Real-World Applications

Automating a cross-chain payment

A user needs to send tokens from Protocol A to an address governed by Protocol B. The agent uses `get_quote` to ensure the optimal, gas-efficient route is calculated before submitting the signed payload via `execute_supertx`.

Verifying smart contract deployment status

A developer deploys a new module and needs confirmation it worked. The agent calls for the supertransaction hash and then uses `get_explorer_status` to wait until 'Success' is confirmed on-chain.

Executing a sponsored transaction

A user wants to pay another user but doesn't want to deal with gas fees. The agent calculates the quote using `get_quote` and executes it as a sponsored supertransaction, simplifying the payment for both parties.

Building automated DeFi yield farming

A power user wants their agent to automatically rebalance assets across three different protocols. The MCP coordinates the necessary sequence of quotes and executions, ensuring every step is correctly signed and tracked.

Patterns to Avoid

Using simple API calls for complex flows

✗ AVOID

Trying to manually script a multi-step DeFi transaction using basic read/write functions. You'll get stuck when the gas price changes or if one step fails.

✓ INSTEAD

Always calculate the full intent first with `get_quote` to model all variables, and then use `execute_supertx` for reliable execution.

Ignoring transaction status tracking

✗ AVOID

Executing a supertransaction and assuming it worked just because the initial call succeeded. You might not know if it failed due to gas or contract logic.

✓ INSTEAD

Always follow up with `get_explorer_status` until you get final confirmation that the operation completed successfully on-chain.

Overlooking account type requirements

✗ AVOID

Writing code that only assumes a simple wallet (EOA) when the target contract requires advanced features, leading to failed transactions.

✓ INSTEAD

The MCP handles flexible account modes, supporting Smart Accounts and EOA-7702 accounts for maximum compatibility.

The Right Fit

Use this MCP if your blockchain workflow is complex, multi-step, or requires gas optimization. Specifically, you need to calculate the cost of a full intent before committing funds (use `get_quote`), and then execute that entire sequence as one unit (`execute_supertx`). Don't use it if all you need is a simple read operation or checking data against an indexed database; general-purpose knowledge retrieval tools are better for those. You must use this MCP when gas management, account abstraction, or supertransaction capabilities are central to the task.

Biconomy Account Abstraction: Solving Complex Web3 Payment Flows

Today, orchestrating a multi-step payment across DeFi protocols means juggling multiple API calls. You check gas costs here, sign the payload there, and then monitor the status in a third dashboard. This process is tedious, error-prone, and requires deep knowledge of smart contract plumbing.

With Biconomy's MCP, your agent handles this complexity automatically. You simply tell it what you want to achieve—say, 'Pay 5 ETH, sponsored by my account.' The system calculates the optimal quote using `get_quote` and executes the entire intent in one go. You get a single confirmation of success for complex payments.

Biconomy Account Abstraction: Managing On-Chain Transaction States

Before this MCP, tracking a supertransaction involved constantly refreshing block explorers and waiting on manual confirmations. It was an opaque process where the status could change from 'Pending' to 'Failed' without clear context.

Now, your agent provides real-time visibility into every step. You send the transaction with `execute_supertx` and immediately get access to robust tracking that confirms if the flow succeeded or why it failed. The uncertainty of blockchain operations is dramatically reduced.

Biconomy (Account Abstraction) MCP: 3 Tools for Web3 Supertransactions

Use these tools to calculate transaction costs, submit signed payloads, and check the status of supertransactions on-chain.

#	TOOL	DESCRIPTION
01	`execute_supertx`	Sends a fully signed supertransaction payload to the blockchain for immediate execution.
02	`get_explorer_status`	Checks and returns the current, real-time status of any previously submitted supertransaction hash.
03	`get_quote`	Calculates an optimized quote for a complex transaction flow, including gas cost estimates.

See It in Action

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

U I need to process a payment for a new user account using sponsored gas.

Supertransaction Quote Generated

PARAMETER	VALUE
Target Flow	Single Payment (ETH)
Gas Cost Estimate	0.02 ETH
Sponsorship Fee	Included $
Total Estimated Cost	**0.03 ETH**

Please review and confirm if you want to proceed with execution.

U Check the status of my supertransaction for the asset transfer.

Supertransaction Status Report (Hash: 0xabc...):

- **Current State:** Success
- **Confirmation Block:** 18,452,993
- **Timestamp:** Today at 10:15 AM UTC
- **Details:** The payment was successfully confirmed on-chain. You can view the full details in the Biconomy explorer.

U What's the best way to execute this multi-step DeFi yield claim?

The optimal path requires three sequential operations. I've generated a quote for you:

Quote Summary:

- Step 1 (Initial Claim): 0.01 ETH
- Step 2 (Rebalance Swap): 0.005 ETH
- Step 3 (Final Deposit): 0.008 ETH

Total estimated cost: **0.023 ETH**. Ready to execute?

Frequently Asked Questions

01 **How does the Biconomy Account Abstraction MCP help with gas fees?**

It simplifies gas management by allowing you to get quotes for sponsored or gasless options. This means your agent can calculate and execute complex transactions without requiring you to pay standard gas fees upfront.

02 **Can I use Biconomy Account Abstraction MCP for multi-step payments?**

Yes, absolutely. The system is designed to handle entire intents—multiple steps like quoting, signing, and executing—as one cohesive supertransaction flow through the agent.

03 **How quickly can I confirm if my transaction succeeded?**

The MCP allows you to track transactions in real-time. You don't have to manually check a block explorer; your agent uses dedicated tools to monitor and report the final status, whether it succeeds or fails.

04 **What is the difference between using this MCP and writing custom smart contracts?**

You don't need to write complex contract code. This MCP lets your agent interact with the underlying blockchain logic using natural language instructions, abstracting away the low-level coding details.

05 **Is Biconomy Account Abstraction MCP compatible with different account types?**

Yes, it supports a wide array of account modes, including Smart Accounts and EOA-7702. This flexibility ensures the tool works across various Web3 setups.

06 **What if my supertransaction fails? How does Biconomy Account Abstraction MCP help?**

The tracking tools provide clear, granular status updates. Instead of just saying 'failed,' it tells you exactly where the transaction broke down in the sequence, helping you debug the process.

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 `"biconomy-account-abstraction": { "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

Biconomy (Account Abstraction) 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 Biconomy (Account Abstraction). 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	Biconomy (Account Abstraction) MCP
Server ID	`019e386d-36ab-725c-ad7c-260aece922e9`
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/biconomy-account-abstraction.