MCP SERVER | NO CODE | CLOUD HOSTED

# Epic Games EOS MCP

## Check player data, friends lists, and store catalogs.

Epic Games EOS MCP connects your AI client directly to the Epic Online Services backend. Use it to check public account details, pull friend lists, and search products across the entire Epic Games Store catalog. It's how you turn natural language queries into actionable game data.

A+ Quality Score 100/100

player-metadata | social-graph | account-lookup | game-services | user-profiles

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/epic-games-eos

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.

# Epic Games EOS MCP

3 tools available
Cloud-hosted on Vinkius

This connector lets your agent talk to Epic's core gaming infrastructure. Instead of logging into multiple developer portals or running complex scripts, you ask your AI client a question—like 'What devices is this account linked to?'—and get the answer instantly. You can pull public metadata for accounts, check who a player's friends are, and search deep within the store's catalog using simple conversational prompts. It eliminates hours of API setup and data stitching. With Vinkius managing the connection, you connect once and gain access to this massive dataset through any compatible AI client. It's essential for anyone who needs reliable, structured player identity and game metadata without writing a single line of boilerplate code.

## Core Capabilities

**01 — Determine Account Status**

Retrieve public details, display names, and linked platforms for specific Epic Account IDs.

**02 — Manage Social Connections**

Fetch a player's current friends list based on their unique account ID.

**03 — Browse Game Products**

Search the live Epic Games Store catalog using specific deployment or sandbox IDs.

## One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/epic-games-eos** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your required credentials: the Epic Client ID and Client Secret from the developer portal.

02 Your AI client authenticates with Vinkius, granting it access to the live EOS tools.

03 You ask your agent a question in plain language; it maps that request to the appropriate tool (e.g., fetching friend lists) and returns structured data.

The bottom line is you talk to your AI client, and it handles all the complex API calls necessary to get accurate game data back to you.

## Built For

Game developers who are tired of writing repetitive boilerplate code for player validation. Community managers struggling with manual profile lookups. Product analysts needing rapid access to large-scale social and product metadata.

**Game Developer**

Using the MCP, you quickly verify account data or check friend connections during a test build without setting up complex local testing environments.

**Community Manager**

You look up player profiles and external links for specific users to manage community outreach efficiently.

**Product Analyst**

You pull large datasets of account metadata or store catalog details to spot trends that manual analysis would miss.

## What Changes When You Connect

01 Stop writing boilerplate API code. Instead of crafting multi-step scripts just to check a user's account status, you simply ask your agent, and it uses the `get_account_info` tool to return clean JSON data immediately.

02 Quickly validate player identity. You can use the MCP to pull public metadata for any Epic Account ID, making onboarding and verification much faster than manual database checks.

03 Understand social dynamics instantly. Needing a full friend list? The `get_friends_list` tool gives you this data point without ever leaving your chat window or IDE.

04 Research the market effortlessly. Use the MCP to search the store catalog, pulling product details via `search_store_catalog` for competitive analysis—all from a single prompt.

05 Work across platforms and services. The ability to check cross-platform identity links means you can confirm if an account is linked to Steam or PlayStation directly through your agent's query.

## Real-World Applications

**Auditing a Player's Profile**

A community manager needs to know everything about a high-value player. They ask their agent, and it first runs `get_account_info` to grab the display name and linked platforms. Then, it uses `get_friends_list` to identify key contacts in that social graph.

**Validating a New Game Feature**

A developer needs to confirm if a specific product exists in the store for testing. They prompt their agent with the necessary IDs, and it executes `search_store_catalog`, returning confirmation on availability before any actual deployment.

**Generating Onboarding Reports**

A product analyst wants a report showing account data trends. They use the MCP to systematically run `get_account_info` for several test IDs, building a structured dataset that would otherwise require dozens of manual API calls.

**Checking Account Linkage**

A QA engineer must confirm if an account ID is linked to multiple external services. They prompt the MCP, which utilizes its identity verification capabilities to report on Steam and PlayStation links simultaneously.

# Patterns to Avoid

### Hardcoding API calls

✗ AVOID

Writing a 100-line Python script that requires manually handling OAuth tokens, passing multiple parameters for account IDs, and parsing JSON responses before doing anything useful.

✓ INSTEAD

Just ask your agent. Use the MCP to initiate data retrieval with a simple prompt like 'Get account info for ID X.' Your AI client handles all the token management and structure through the `get_account_info` tool.

### Ignoring Scope Limitations

✗ AVOID

Trying to use generic database queries when you actually need specific, structured game data like friend lists or store catalog items.

✓ INSTEAD

Don't query the general database. Use the specialized tools in this MCP. For social connections, always start with `get_friends_list`.

### Bypassing Credentials

✗ AVOID

Running tests without providing valid Epic Client IDs and Secrets because you're trying to keep your setup simple.

✓ INSTEAD

You must provide the required credentials (Client ID/Secret) when subscribing. The system needs them for the MCP to execute any of its tools successfully.

## The Right Fit

Use this MCP if your core problem revolves around structured player identity, social graphs, or product metadata within the Epic Games ecosystem. You need reliable access to public account details via `get_account_info` or friend lists via `get_friends_list`. Don't use it if you simply need general knowledge about gaming history; for that, a pure search engine tool is better. Also, don't use it if your goal is to *modify* the data (e.g., changing a username); this MCP only allows reading and querying information. If you are looking for broad competitor analysis across multiple unrelated services, you might need a more general multi-source connector.

## Dealing with fragmented gaming data points is exhausting.

Right now, getting a full picture of a player's standing requires jumping through hoops. You check the account ID in one dashboard, then you open another tool to find their friends list. If you need store details, you switch contexts entirely and start manually looking up product IDs.

With this MCP, all that data comes together. Your agent handles the cross-referencing. Instead of juggling multiple tabs and copy-pasting credentials, you ask a single question, and it pulls account metadata, friend links, and store details into one coherent answer.

## Accessing Player Identity Data

The manual steps—verifying the account link status, pulling the friends list manually, and cross-referencing public metadata—are gone. The MCP handles all of that complex logic behind the scenes.

Now you just talk to your agent. It knows exactly which tool to run to give you the clean data you need, making deep player analysis instantaneous.

# Epic Games EOS: 3 Tools Available

These three tools allow your agent to manage everything from querying individual user accounts to searching the entire product catalog.

| # | TOOL | DESCRIPTION |
|---|---|---|
| 01 | `get_account_info` | Retrieves public profile information for one or more Epic Account IDs. |
| 02 | `get_friends_list` | Pulls the current list of friends associated with a specific Epic account ID. |
| 03 | `search_store_catalog` | Searches the live Epic Games Store catalog using required Sandbox or Deployment IDs. |

# See It in Action

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

U Get account info for Epic ID '0123456789abcdef'.

Retrieving data for ID '0123456789abcdef'... Display name: 'EpicExplorer'. This account is linked to Steam and PlayStation Network.

# Frequently Asked Questions

01 **What does the Epic Games EOS MCP help with?**

This MCP connects your AI agent to Epic Online Services. It lets you look up public account metadata, retrieve friend lists, and search product details in the store catalog.

02 **Do I need developer credentials for get_account_info?**

Yes, you must subscribe with your valid Epic Client ID and Client Secret from the dev portal. This grants the MCP permission to query account data.

03 **Can I use this MCP just for friend lists?**

Absolutely. You can isolate the functionality by only querying the `get_friends_list` tool, which retrieves a player's current social network connections.

04 **Does search_store_catalog work for all products?**

No, `search_store_catalog` requires you to provide valid Sandbox or Deployment IDs. You must know where in the ecosystem you want to search.

05 **Is this MCP designed only for developers?**

Not necessarily. While useful for development, community managers and product analysts find it equally valuable for reporting and research purposes.

# 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 `"epic-games-eos": { "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

# Epic Games EOS 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 Epic Games EOS. 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 | Epic Games EOS MCP |
| Server ID | `019d8435-478b-7085-9a66-9b25b6468ab4` |
| 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/epic-games-eos.