

MCP SERVER

NO CODE

CLOUD HOSTED

Steam MCP

Analyze Player Behavior & Game Stats

Steam MCP gives your AI client direct access to millions of data points about Steam users and games. It lets you query a player's entire history—from their current profile status and owned game library to specific achievements, stats, and even recent news updates.

Analyze user progression and deep performance metrics without ever leaving your chat interface.

A+ Quality Score 100/100

player-profiles

game-library

achievements

gaming-stats

user-data

api-integration



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

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.

Steam MCP

10 tools available
Cloud-hosted on Vinkius

This connector gives your AI agent the ability to pull granular data directly from the Steam Web API. You can ask it for anything related to a player's account: what games they own, how much time they've spent in specific titles, or if they earned a certain achievement. It doesn't just read profiles; it digs into community progress and performance metrics. When you connect this MCP through Vinkius, your AI agent treats Steam like another data source—just as easy to query as checking an internal database. You can automatically check for recent news about an app or list out the badges a user has earned in-game. It's all done using natural language prompts.

Core Capabilities

01 — Analyze Player Profiles

Retrieve basic and advanced public information, like account age, status, and avatar details for any Steam user.

03 — Track Achievement Status

Check the progress and unlock history for specific achievements within any supported game title.

05 — Get Community Badges and Levels

Check a user's community standing by accessing their equipped badges or current Steam level.

02 — List User Game Libraries

Get a complete list of games owned by a specific user along with their recorded playtime hours in those titles.

04 — Monitor Recent Activity

See which games a user has played most recently, including detailed session times over the past couple of weeks.

06 — Access Performance Statistics

Pull detailed in-game metrics and performance stats for specific titles played by the target user.

One Click on Vinkius — From Prompt to Execution

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

- 01** Subscribe to this MCP and enter your private Steam Web API key within Vinkius.
- 02** Instruct your AI agent using natural language, specifying what data you need (e.g., 'What games does user X own?').
- 03** The MCP uses the provided key to query the Steam API and returns structured game data directly into your chat window.

The bottom line is that instead of logging into Steam's website, you just ask your AI client for the data you need.

Built For

This MCP is essential for content creators, QA engineers, or market analysts who spend too much time manually checking player forums and web dashboards. If your job requires understanding user behavior across gaming titles—the pain of sifting through millions of data points by hand ends here.

Game Data Analyst

You use this MCP to track how specific features or updates impact player engagement, quickly checking metrics like game ownership using `get_owned_games`.

Content Strategist

You analyze player profiles and achievements to generate content ideas or understand the community's current focus, such as getting recent news for a specific app.

Product Manager (Gaming)

You determine if new features are valuable by checking user stats for game titles using `get_user_stats_for_game` to benchmark performance changes.

What Changes When You Connect

- 01** You can instantly audit a user's entire library. Instead of clicking through dozens of game pages, simply ask your agent to run `get_owned_games` and see everything in one response.

-
- 02 Stop guessing about player engagement. Use `get_recently_played_games` to find out exactly what users played last week—it shows you the real hot titles right now.

 - 03 Track user prestige without manual lookups. You can check a player's status, their community level using `get_steam_level`, and even see which badges they've earned with `get_player_badges`.

 - 04 Get performance metrics that matter for development. Running `get_user_stats_for_game` lets you benchmark user skill or usage rates for specific titles against a baseline.

 - 05 Stay current on the industry. Need to know about an update? Use `get_app_news` to pull recent articles and updates directly, saving you from trawling through developer blogs.
-

Real-World Applications

Auditing Competitive Play

An esports analyst needs to compare the skill gap between two top players. Instead of manually checking their profiles, they prompt the agent to run `get_user_stats_for_game` for both accounts on a specific map. The AI client returns a clear comparison of headshot percentages and K/D ratios.

Researching Game Updates

A product manager needs to know if a new feature is being discussed. They use `get_app_news` for the game's ID, immediately seeing the latest official updates and developer announcements without visiting any external websites.

Generating Content Ideas

A content creator wants to write about popular games. They ask the agent to run `get_owned_games` for 50 random users, instantly generating a list of the top 10 most owned games right now.

Investigating Account Status

A developer needs to validate if a user has completed all tutorial challenges. They ask the agent to check `get_player_achievements`, getting an instant report on which key milestones the player is missing.

Patterns to Avoid

Treating it like a simple search tool

✗ AVOID

Writing 'Tell me about all games.' This prompt only gets general info and misses specific user data.

✓ INSTEAD

Be precise. To get the list of titles, use the explicit command to run `get_owned_games` for the target user ID. Don't generalize; ask for the exact function.

Forgetting required parameters

✗ AVOID

Asking 'Show me stats.' The agent fails because it doesn't know *which* game or *which* player you mean.

✓ INSTEAD

Always specify both the user ID and the necessary context. For performance data, run `get_user_stats_for_game(user_id=X, app_id=Y)`.

Trying to scrape web pages

✗ AVOID

Asking your agent to 'Scrape Steam's main page for news.' This is inefficient and often incomplete.

✓ INSTEAD

Use the dedicated tool `get_app_news`. Give it the specific App ID, and the MCP handles the structured extraction of only the latest articles.

The Right Fit

Use this Steam MCP if your goal involves quantifying user behavior—you need data on *what* a player owns, *how much* they play, or *when* they achieved something. You must be analyzing specific metrics like playtime (`get_owned_games`), progress toward goals (`get_player_achievements`), or community status (`get_steam_level`).

Don't use this MCP if you just need to know general marketing information or casual opinions. If your goal is simply 'What are people saying about the game?'—use a generic messaging tool for social media sentiment analysis. If you just want basic info like, 'Is this user online?'—`get_player_summaries` works, but don't rely on it for deep behavioral analysis.

Tracking Gaming Data Used to Be Pure Guesswork

If you're doing market research or analyzing user retention today, what does that look like? You're logging into Steam, checking a profile for ownership details. Then you switch tabs to check recent activity—copying down playtime hours and then opening another tab just to see their achievements. It's slow, tedious copy-pasting between multiple web interfaces.

With this MCP, that entire process collapses into a single prompt. You ask your agent what the user owns and how much they play; it runs `get_owned_games` and gets you the full library list, including playtime stats. You get structured data in seconds.

Accessing Player Profiles with `get_player_summaries`

Before this MCP, getting a user's basic details meant navigating to their profile URL and manually reading the account creation date or current status. This was always unreliable and required multiple clicks.

Now, you just ask your agent for a player summary using `get_player_summaries`. It pulls that data directly into your workflow, clean and ready to use in reports.

Steam MCP with 10 Tools

Use these tools to pull specific gaming data—like a user's library or achievement status—directly into your chat conversation.

#	TOOL	DESCRIPTION
01	<code>get_app_list</code>	Retrieves a complete list of all apps available on the Steam platform.
02	<code>get_badge_progress</code>	Checks how close a user is to earning certain community badges.
03	<code>get_app_news</code>	Pulls recent news articles and updates for any specific Steam app ID.
04	<code>get_owned_games</code>	Lists every game currently owned by a given Steam user account.
05	<code>get_player_achievements</code>	Determines the achievement progress and unlock status for a player in a specific game.
06	<code>get_player_badges</code>	Retrieves all community badges that a Steam user currently has equipped.
07	<code>get_player_summaries</code>	Fetches basic profile information for any specified Steam user, including account details and status.
08	<code>get_recently_played_games</code>	Shows a list of games the user has played in the last two weeks, along with session times.
09	<code>get_steam_level</code>	Gets the current community level and standing of the specified Steam user.
10	<code>get_user_stats_for_game</code>	Retrieves detailed performance metrics and statistics for a player in one specific game title.

See It in Action

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

U Show me the profile of Steam user 76561197960287930.



Player 'Robin' is currently online. Account created in 2003, currently at Steam Level 80. Profile is public with avatar and location visible.

U What games does user 76561197960287930 own and how much have they played?



This user owns 347 games. Top played: Counter-Strike 2 (257 hours), Dota 2 (833 hours), Team Fortress 2 (142 hours). Would you like details on any specific game?

U Get recent news updates for Cyberpunk 2077 (App ID 1091500).



Found 3 recent news articles for Cyberpunk 2077. Latest: 'Update 2.1 — Now Available' released yesterday with new features and bug fixes.

Frequently Asked Questions

01 How do I find out what games a user owns with the Steam MCP?

You run `get_owned_games` for the target user ID. This tool provides an immediate list of every game they own, along with their total playtime in each title.

02 Can I check if a player completed all achievements using Steam MCP?

Yes, use `get_player_achievements`. You provide the user ID and the specific App ID, and it tells you exactly which milestones they've unlocked and when.

03 What is the best tool for checking a player's current status?

Use `get_player_summaries`. This function provides a quick snapshot of the user's profile details, including their avatar visible status and account creation date.

04 How do I check community progress with Steam MCP?

You can use two tools: `get_steam_level` for general standing, or `get_player_badges` to see which specific badges they've earned in the community.

05 Does Steam MCP include information on recent app news?







Yes. You run `get_app_news` and specify the App ID. The MCP retrieves the latest articles and updates for that game, keeping you current without leaving your workspace.

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 <code>"steam": { "url": "..." }</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

Steam 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](https://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 Steam. 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	Steam MCP
Server ID	019d8485-18f4-72da-aae5-679ac1e8569b
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/steam.