MCP SERVER | NO CODE | CLOUD HOSTED

Marketplacer MCP

Audit Products, Orders, and Vendors via Conversation

Marketplacer MCP connects your AI client directly to a complex enterprise marketplace platform. Audit every part of your e-commerce business—from product listings and vendor accounts to detailed order invoices and live shipment tracking—using natural conversation.

A+ Quality Score 100/100

marketplace-management | dropshipping | product-catalog | seller-network | enterprise-commerce

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/marketplacer-enterprise-marketplace-platform

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.

Marketplacer (Enterprise Marketplace Platform) MCP

9 tools available
Cloud-hosted on Vinkius

Managing a large, multi-vendor marketplace means jumping through endless dashboards just to get a full picture of operations. This MCP lets you do that audit work using only plain language with your AI agent. Instead of manually running reports on seller lists or sifting through webhook logs, you talk to this connector and it handles the data retrieval. You can ask your agent to list every active vendor, pull up recent invoices with line item breakdowns, or even trace a shipment's entire fulfillment history. When you connect Marketplacer via Vinkius, your agent gains immediate access to all these deep systems—the product catalog, order audit trails, and logistics data—without needing any boilerplate code. It's about turning complicated backend operations into simple conversations.

Core Capabilities

01 — Audit Product Data

Retrieve detailed product information, including pricing and variants, directly from the marketplace catalog.

02 — Review Order Financials

Track specific orders by listing recent invoices and extracting financial details like tax distributions and line items.

03 — Manage Vendor Networks

Fetch lists of third-party sellers and vendors to check their active status or current product listings.

04 — Track Logistics Flow

List and inspect shipment records to monitor where an order is in the fulfillment process across multiple sellers.

05 — Execute Custom Queries

Run complex, custom GraphQL queries against any specific data point or schema entity within the platform.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/marketplacer-enterprise-marketplace-platform** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your Marketplacer API URL and API Key.

02 Connect this MCP to your preferred AI client, like Claude or Cursor.

03 Ask your agent a question—for example, 'What are the latest 5 invoices?'—and it executes the required tool call.

The bottom line is you talk about what data you need, and the MCP handles talking to the marketplace backend for you.

Built For

This MCP is essential for Marketplace Managers, Operations Engineers, and E-commerce Developers. It's for anyone who spends too much time switching between dashboards or writing complex API calls just to audit the health of a multi-vendor network.

Marketplace Manager

Audits order flows and vendor performance by asking natural language questions, eliminating the need to manually generate reports.

Operations Engineer

Tracks cross-vendor invoices and shipment statuses efficiently, ensuring high fulfillment standards are maintained without constant manual checks.

E-commerce Developer

Tests complex GraphQL queries and verifies product attribute mappings directly within their development environment or terminal.

What Changes When You Connect

01 Instead of generating a spreadsheet report to see vendor performance, you simply ask your agent to list vendors using the `list_sellers` tool. It gets you the data instantly.

02 Stop hunting for order details in separate systems. You can use `list_invoices` and then dive deep into line items with `get_invoice`, giving you a complete financial picture of any sale.

03 Need to know why an order is delayed? Use `list_shipments` to track the fulfillment lifecycle across multiple sellers, all in one conversation flow.

04 Don't rely on fixed reports. With the `graphql_query` tool, you can fetch highly specific data points—like a legacy ID alongside a title—that standard tools miss.

05 Verify your entire system integration stack by calling `list_webhooks`. You confirm if all automated event subscriptions are set up correctly, eliminating manual audit time.

Real-World Applications

Auditing Seller Compliance

A Marketplace Manager needs to ensure no vendor is inactive. They ask their agent to run `list_sellers` and then filter the results to identify vendors who haven't listed any products recently, allowing them to address compliance issues immediately.

Investigating a Missing Shipment

An Operations Engineer discovers an order is stuck. They prompt their agent to `list_shipments` using the order ID, tracing the physical movement of goods and pinpointing exactly which seller failed to update the tracking status.

Deep Data Extraction for Reporting

A Developer needs a specific data point—the product's internal SKU linked to its GraphQL identifier. They use `graphql_query` to pull this precise, cross-referenced information that isn't exposed by the standard product listing tools.

Reviewing Financial Discrepancies

A manager spots a tax discrepancy on an invoice. They ask for `list_invoices` to find the transaction, then use `get_invoice` to view the full payload and confirm exactly how the taxes were distributed across line items.

Patterns to Avoid

Treating it like a simple database tool

✗ AVOID

Asking the agent simply to 'Show me all product data.' This will return raw, unorganized lists without context or filters.

✓ INSTEAD

Start by asking specific questions. To list products, use `list_adverts`. If you need details on one item, follow up with `get_advert` and specify the node ID.

Ignoring the complexity of cross-vendor data

✗ AVOID

Trying to find out who sold a product without knowing if it was managed by the main platform or an external vendor.

✓ INSTEAD

Always check seller status first using `list_sellers`. If you need logistics, use `list_shipments` which tracks fulfillment across the entire network.

Over-relying on general API calls

✗ AVOID

Running a generic GraphQL query without knowing exactly what data fields are needed, resulting in massive, unusable data dumps.

✓ INSTEAD

First, check the system's structure by running `list_categories` to understand the hierarchy. Then, narrow your scope with `graphql_query` using specific field names.

The Right Fit

Use this MCP if you need a single conversational interface to audit and manage complex e-commerce systems involving multiple sellers, detailed product catalogs, and multi-stage logistics. You should use it when the pain point is 'I have too many dashboards to check.' Don't use this if your only goal is simple data entry or updating content; for those tasks, you need a dedicated write/update tool, not an audit connector like these listing functions. If you just want basic product search without financial context, `list_adverts` is fine, but if you need the full picture—invoices, sellers, and tracking—this MCP is required.

The headache of juggling marketplace dashboards.

Right now, auditing your enterprise marketplace means jumping between 4-5 different portals: one for seller accounts, one for product listings, another for order history, and a fourth just for tracking shipments. You copy IDs from the invoice dashboard into the shipping portal, then run a separate report to cross-check vendor status. It's clicking through tabs until your eyes blur.

With this MCP, you simply tell your agent what you need. Ask it to audit recent sales and ask it to check if those orders have corresponding shipments. The entire workflow—from listing invoices via `list_invoices` to verifying shipment records with `list_shipments`—happens in one conversation thread.

Accessing the full spectrum of market data with Marketplacer MCP.

You no longer have to manually cross-reference product metadata from the catalog, audit vendor listings using `list_sellers`, and then match that against fulfillment records. You can do all three tasks in a single prompt.

The result is an immediate, conversational answer that integrates seller performance, tax details, and current inventory status—a view you couldn't get by just looking at any one dashboard.

Marketplacer (Enterprise Marketplace Platform) with 9 Tools

Use these tools in your AI client to access specific functions like listing invoices, getting product details, or running custom GraphQL queries against the Marketplacer platform.

#	TOOL	DESCRIPTION
01	`list_adverts`	Lists all available product items (adverts) in the marketplace catalog.
02	`get_advert`	Retrieves detailed information about a specific product advert using its node ID.
03	`list_invoices`	Fetches a list of all registered orders on the platform for audit purposes.
04	`get_invoice`	Gets the full details and line items for one specific invoice using its node ID.
05	`list_sellers`	Retrieves comprehensive lists of active third-party vendors on the platform.
06	`list_categories`	Explores the marketplace's internal category structure and product hierarchies.
07	`list_webhooks`	Audits which event subscriptions are configured for automated integrations.
08	`list_shipments`	Generates a list of all tracked shipment records to monitor logistics.
09	`graphql_query`	Allows you to run sophisticated, custom GraphQL queries against any data point in the marketplace schema.

See It in Action

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

U List the last 5 invoices in my Marketplacer instance

I've retrieved the latest 5 invoices. Highlights include Invoice #INV-123 ($150.00 from Seller A) and Invoice #INV-456 ($85.50 from Seller B). Would you like to see the shipment status for any of these?

U Search for sellers containing 'Outdoor' in the network

Searching seller network... I've identified 3 vendors: 'Outdoor Adventure Gear' (ID: sel-123), 'Premium Outdoor Living' (ID: sel-456), and 'The Great Outdoors' (ID: sel-789). I can fetch the advert count for each seller if you'd like.

U Run a GraphQL query to fetch the 'legacyId' and 'title' for advert ID 'node-987'

Executing custom query... Done. For advert 'node-987', the Legacy ID is '554433' and the Title is 'Titanium Mountain Bike'. Would you like to fetch the inventory quantity using another query?

Frequently Asked Questions

01 **How do I check vendor accounts using Marketplacer MCP?**

You use the `list_sellers` tool. This fetches a comprehensive list of all active vendors, allowing you to quickly audit their account statuses and verify who is currently on your platform.

02 **Can I track an order's journey with Marketplacer MCP?**

Yes, use the `list_shipments` tool. It retrieves detailed records of fulfillment, allowing you to monitor where an item is in its journey across the entire distributed seller network.

03 **Is the GraphQL query tool for general data or specific reports?**

The `graphql_query` tool is for highly customized data retrieval. Use it when standard listing tools don't expose a niche piece of information, allowing you to fetch precise data points across any schema entity.

04 **How do I check if my integrations are working?**

Use the `list_webhooks` tool. This audits your configured event subscriptions and webhooks, confirming that all automated connections are set up correctly without manual checks.

05 **Can I retrieve product details using GraphQL identifiers through my agent?**

Yes. Use the `get_advert` tool and provide the specific GraphQL Node ID. Your agent will fetch the full record, including descriptions, pricing arrays, and variants directly from the Marketplacer schema.

06 **How do I check recent orders on my marketplace through a conversation?**

The `list_invoices` tool allows your agent to retrieve recent order invoices. You'll see the amount, status, and associated seller for each order, helping you monitor high-level marketplace activity instantly.

07 **Can my agent run custom GraphQL queries against the Marketplacer endpoint?**

Absolutely. Use the `graphql_query` tool to execute sophisticated custom queries. You can provide the query string and optional JSON variables, and your agent will return the raw GraphQL response payload securely.

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 `"marketplacer-enterprise-marketplace-platform": { "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

Marketplacer (Enterprise Marketplace Platform) 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

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	Marketplacer (Enterprise Marketplace Platform) MCP
Server ID	`019d75cf-a4d1-70bc-b90c-5df410498be6`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE