MCP SERVER | NO CODE | CLOUD HOSTED

Ordergroove MCP

Manage Billing Cycles and Customer Retention

Ordergroove MCP manages your recurring revenue cycle directly through your AI client. This connector gives agents the ability to read customer data, check product availability, and manage subscriptions from start to finish. Need to cancel a user's service or update billing details? You can do it programmatically without leaving your chat window. It centralizes all subscription management functions for any e-commerce business model.

A+ Quality Score 100/100

recurring-billing | customer-loyalty | subscription-lifecycle | product-catalog | retention-automation

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

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.

Ordergroove MCP

9 tools available
Cloud-hosted on Vinkius

Managing recurring revenue used to mean jumping between the billing platform, the CRM, and the product catalog—a tedious mess of logins and copy/pasting. Now, you can let your AI agent handle that entire lifecycle. This MCP connects your preferred AI client directly to Ordergroove's core functions. Your agent accesses detailed customer profiles, tracks which products are available for recurring orders, and maintains a complete history of every subscription. You don't just get data; you gain control over the full billing cycle. Whether it’s changing a product frequency or reactivating an account after dormancy, your AI agent handles the commands in plain language. Since Vinkius manages this catalog, connecting your client to Ordergroove means all your critical revenue operations are available in one place.

Core Capabilities

01 — Manage Customer Records

Retrieve detailed customer profiles or list groups of customers for review.

02 — View Subscription Status

List all active, past, or paused subscriptions, filtering by a specific customer ID or status type.

03 — Control Billing Lifecycle

Cancel, update, reactivate, or modify the details of any recurring subscription.

04 — Check Product Inventory and Details

Look up specific product information or list all available products to ensure recurring orders are viable.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to the Ordergroove MCP on Vinkius and enter your API Key credentials.

02 Next, prompt your AI client with a natural language request, like 'Cancel subscription X because of Y reason.'

03 Your agent translates that command into the necessary tool calls, executes them against Ordergroove, and returns the resulting status update.

The bottom line is, you tell your AI what needs to happen with a customer's billing, and it makes it happen across the system.

Built For

Billing Managers and Customer Success Leads need this. They struggle when manual data verification slows down retention efforts or forces them into platform switching. This MCP lets agents manage complex billing actions without leaving their primary conversational workspace.

Billing Manager

Needs to check subscription statuses, list all customers for quarterly audits, and quickly cancel services when an account is flagged as delinquent.

Customer Success Agent

Handles retention calls by getting customer details, reactivating dormant accounts, or updating a service plan's frequency in real time.

E-commerce Operations Lead

Manages product availability and ensures that the products listed for recurring orders match current catalog specifications before launch.

What Changes When You Connect

01 Stop revenue leakage by instantly canceling any subscription. Use the `cancel_subscription` tool to terminate service when a customer leaves, ensuring proper record-keeping.

02 Deepen customer relationships using agent access to detailed records. The `get_customer` tool pulls all necessary profiles so your team can speak directly to the user's history.

03 Handle complex billing changes with precision. Instead of multiple forms, use `update_subscription` to modify quantities or change frequencies in a single command.

04 Speed up onboarding and support by verifying service status immediately. The `list_subscriptions` tool lets you quickly see every recurring order for any customer.

05 Maintain product consistency across billing cycles. Use the `get_product` tool before setting up new services to ensure pricing and details are correct.

06 Automate recovery workflows easily. If a user account goes dormant, your agent can use `reactivate_subscription` immediately upon confirmation.

Real-World Applications

Customer Retention After Service Lapse

A customer calls because their service stopped. The agent prompts the MCP to run `get_customer`, confirms account details, and then uses `list_subscriptions` to find the specific plan ID. Finally, they use `reactivate_subscription` and confirm the change with the customer.

Handling Bulk Account Audits

The billing team needs a list of all paused services for an audit. The agent calls `list_subscriptions`, filtering by 'paused' status, giving the team a clean manifest without manual dashboard exports.

Correcting Billing Errors Mid-Cycle

An employee realizes a customer's product quantity was wrong on their plan. They use `get_subscription` to confirm the ID and then call `update_subscription`, specifying the new desired quantity, instantly fixing the billing record.

New Product Launch Validation

Before marketing a new offering, an operations lead uses `list_products` to check against existing catalog entries. They then use `get_product` on the proposed item to validate required fields and pricing structures.

Patterns to Avoid

Trying to update without checking details

✗ AVOID

A user tries to 'Change the plan for customer X' but doesn't provide the correct subscription ID, leading to an error that the agent can't process.

✓ INSTEAD

First, ask your AI client to run `get_subscription` using the suspected customer and service name. Once the specific subscription ID is returned, use `update_subscription` with that exact ID for success.

Listing customers then guessing status

✗ AVOID

A user runs `list_customers` but gets a massive list. They can't tell which ones are active or if they have billing issues, requiring them to manually cross-reference spreadsheets.

✓ INSTEAD

Immediately follow up the customer listing with `list_subscriptions`, making sure you filter by the 'inactive' status. This isolates exactly who needs attention.

Asking about products and services separately

✗ AVOID

A user asks for product details, then later asks for billing details, creating two separate, unlinked conversations.

✓ INSTEAD

Combine the requests: 'Check if Product Z is available and list all active subscriptions for Customer A.' This lets your agent do both `get_product` and `list_subscriptions` in one go.

The Right Fit

Use this MCP if your core pain point involves managing recurring billing, subscription status, or customer retention actions. If you need to know: 'Is service X active?' or 'Can I change the frequency of Y?', this is for you. Don't use it if your primary goal is general account ledgering (you need a dedicated accounting tool) or simple contact list management (a basic CRM tool suffices). This MCP excels at the transactional layer—the actions taken *on* an existing subscription, like cancelling it (`cancel_subscription`) or modifying it (`update_subscription`). If you only want to read data and nothing else, consider if `list_customers` is enough; but for true operational workflow, this connector is necessary.

Billing cycles are a mess of manual cross-referencing.

Today, when you need to understand why a customer's billing stopped, you have to jump through hoops. You pull the account number from the CRM, open the separate billing portal using that number, then manually check if the product listed matches what they originally signed up for. Then you copy-paste all those details into a spreadsheet just so your team can review it.

With this MCP, you tell your agent to investigate the issue. It runs `get_customer` and then links that data with `list_subscriptions`. Your AI client pulls together every piece of required information—the customer history, the current plan status, and even the associated product details—and gives it back in one cohesive summary.

Manage your entire billing lifecycle through Ordergroove MCP.

The tedious parts that vanish are manual data lookup and status changes. You never have to log into a separate portal just to cancel service or reactivate an account; the `cancel_subscription` and `reactivate_subscription` tools handle those commands directly, using only natural language.

It's not about reading data anymore; it's about taking action. Your agent becomes the single point of truth that can verify product details (`get_product`) and then execute a billing change (`update_subscription`). That's how efficient retention works.

Ordergroove MCP: 9 Tools for Subscription Management

Use these tools to list customers, get product details, and perform lifecycle actions like canceling or updating any subscription through your AI client.

#	TOOL	DESCRIPTION
01	`cancel_subscription`	Cancels a specified subscription, requiring you to state why the cancellation is happening.
02	`get_customer`	Retrieves full details about a single customer by identifying them.
03	`get_product`	Fetches all necessary information for one specific product listing.
04	`get_subscription`	Looks up the complete details of a single subscription record using its ID.
05	`list_customers`	Retrieves a list of all customers associated with your Ordergroove account.
06	`list_products`	Gets a comprehensive list of every product available in the catalog.
07	`list_subscriptions`	Lists all subscriptions, allowing you to filter by customer ID or current status (active, paused, etc.).
08	`reactivate_subscription`	Turns back on a subscription that was previously canceled or suspended.
09	`update_subscription`	Modifies core details of an existing subscription, such as frequency or quantity.

See It in Action

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

U List active subscriptions for customer 'cust_123'.

Fetching subscriptions... I've found 2 active recurring orders for this customer.

U Cancel subscription #sub_abc because the customer requested a refund.

Processing cancellation... Success! Subscription #sub_abc is now cancelled with reason: 'Refund Requested'.

Frequently Asked Questions

01 **Can I list all subscriptions with Ordergroove MCP?**

Yes, you can use `list_subscriptions`. This tool lets you filter the results by specific customer IDs or current statuses (like active or suspended) so you only see what you need.

02 **How do I cancel a subscription using Ordergroove MCP?**

You use the `cancel_subscription` tool. Be sure to include a specific reason for the cancellation when prompting your agent, as that information is required by the system.

03 **Does Ordergroove MCP let me check product availability?**

Yes. Use the `list_products` and `get_product` tools to get detailed information about every item in your catalog, helping you confirm if a recurring order is viable.

04 **Can I update a customer's plan details with Ordergroove MCP?**

Yes, the `update_subscription` tool modifies existing plans. You can change quantities or adjust frequencies for active services in one command.

05 **What if I need to find information on a specific customer?**

Use the `get_customer` tool and provide the necessary ID. This returns all associated details, allowing your agent to understand the full context of their account immediately.

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 `"ordergroove": { "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

Ordergroove 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 Ordergroove. 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	Ordergroove MCP
Server ID	`019d75ec-45b8-71b4-8467-5beab503e035`
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/ordergroove.