

MCP SERVER

NO CODE

CLOUD HOSTED

Polar MCP

Manage your entire digital commerce lifecycle.

Polar MCP connects your AI client to digital commerce operations. Manage everything—from listing products and tracking subscriptions to creating checkouts and auditing revenue—all through natural conversation. You control your entire storefront, discounts, customer data, and order history without opening the web dashboard.

A+ Quality Score 100/100

creator-economy

digital-products

subscription-billing

checkout-flow

revenue-tracking

monetization



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.

Polar MCP

10 tools available

Cloud-hosted on Vinkius

Your agent gains full access to your digital commerce backend via this MCP. Instead of logging into the Polar dashboard to answer questions about sales or inventory, you just ask. Need to know if a specific customer has an active subscription? Your agent checks that instantly. Want to see all products and check their current pricing structure? It pulls the catalog for you. You can create new discount codes on the fly or audit previous checkout sessions to track abandoned carts. This gives your AI client the power of a dedicated commerce engineer, allowing it to run complex reports and execute operations right where you work. Connecting this MCP through Vinkius means you keep all your core business tools visible in one place, making your digital product management instantaneous.

Core Capabilities

01 — Audit the entire product catalog

List every product, whether it's a single purchase item or an ongoing subscription, along with its pricing details.

03 — Review order history and revenue

List completed transactions showing the customer, product purchased, final amount, and payment status for revenue tracking.

05 — Manage promotional codes

List existing discount codes or create entirely new ones with fixed amounts or percentage reductions.

02 — Track customer billing status

Retrieve detailed information on customers and monitor their active, canceled, or past-due subscriptions.

04 — Generate checkout sessions

Create a new billing session for specified products, optionally applying a discount code to generate a unique payment link.

One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to this MCP and enter your Polar Personal Access Token.
- 02 Select your preferred AI client (Claude, Cursor, etc.) within the Vinkius Marketplace.
- 03 Ask your agent a natural language question like, 'Show me all active subscriptions for Jane Doe.'

The bottom line is that you talk to your AI agent, and it talks to Polar to get you the answer or complete the task.

Built For

This MCP is built for people who manage digital goods—the e-commerce owner, the billing manager, or the developer building a product around subscriptions. You're tired of clicking through multiple dashboards just to get basic sales data.

E-commerce Manager

Needs to run reports on revenue and audit active discount codes without leaving their main workflow.

Developer / Integrator

Wants to build features that calculate checkout sessions or pull product metadata directly into code from their IDE.

Marketing Specialist

Needs the ability to quickly generate promotional discount codes and track conversion rates from checkout attempts.

What Changes When You Connect

- 01 Stop jumping between tabs. You can check a customer's subscription status or pull their full billing history instantly using `list_subscriptions` and `list_customers`. Your agent handles the lookup; you get the answer.

-
- 02 Run real-time sales reports without exporting data. Need to know total revenue from last month? Asking your agent to run `list_orders` provides immediate, structured insights for decision making.

 - 03 Handle promotions instantly. You can create a new discount code with `create_discount` based on a conversation and apply it immediately when simulating a checkout using `create_checkout`.

 - 04 Audit everything in one chat session. Reviewing all products or checking the status of abandoned carts via `list_checkouts` is as simple as asking your agent to compile the list.

 - 05 Build sophisticated automation workflows. You can use this MCP's tools to check product details (`get_product`), confirm customer identity, and then initiate a secure checkout flow.
-

Real-World Applications

Investigating an unusual payment failure

A client reports they were charged twice. Instead of manually searching the dashboard, they ask their agent to list all orders for that customer in the last week using `list_orders`. The agent filters and pinpoints both transactions, allowing immediate resolution.

Onboarding a new developer

A dev needs product details for an upcoming feature. They prompt their agent to use `list_products` first to see available categories, then use `get_product` on a specific UUID to pull all metadata.

Preparing a seasonal promotion

The marketing team needs a limited-time discount. They ask their agent to create a new promotional code via `create_discount`, set it for a fixed 30% off, and confirm its immediate availability before the launch.

Handling customer service inquiries

A user calls about renewal status. The support tech asks their agent to check the account using `list_subscriptions`. The agent immediately reports if the subscription is active, past due, or canceled, letting the tech resolve the issue instantly.

Patterns to Avoid

Assuming data availability

✗ AVOID

The user tries to ask for 'all revenue' without specifying a time frame or filtering by status, leading to an incomplete or ambiguous response.

✓ INSTEAD

Always narrow the scope. If you want historical data, use `list_orders` and specify date ranges. For current inventory checks, always call `list_products` first.

Bypassing checkout steps

✗ AVOID

The user tries to calculate a total price for three products without initiating a formal session, leading to an inaccurate quote.

✓ INSTEAD

To ensure accuracy and apply discounts correctly, you must call `create_checkout` with all product IDs. The resulting URL is the authoritative payment path.

Ignoring existing codes

✗ AVOID

The user creates a new discount code when an active one already exists for that promotion type.

✓ INSTEAD

Before creating, always review `list_discounts` to audit existing promotions. You might find the required code is already live and simply needs activation.

The Right Fit

Use this MCP if your primary goal is managing digital goods—subscriptions, products, orders, and billing flow. It excels at providing a conversational layer over core commerce data; for instance, listing all subscriptions or creating checkouts is what it does best. Don't use it if you only need to read basic account details that don't involve money (e.g., just reading general user profile info). In those cases, a simple CRM connector might be better. Also, if your workflow relies heavily on complex inventory management (physical goods tracking), this MCP will fall short because its focus is purely digital commerce.

Dealing with the Dashboard Maze

Right now, checking a customer's subscription status means navigating to the 'Billing' tab. To audit recent sales, you click on 'Orders,' then filter by date range and payment status. If you need to see what products are available for sale, you open the 'Catalog' section, dealing with multiple clicks just to compile a list.

With this MCP, those tasks happen in plain English conversation. You simply ask your agent to check the subscription status or pull the entire product catalog. The answer is delivered directly back to you, skipping every click and filtering step.

Instant Commerce Control with Polar

You no longer have to copy-paste Product IDs or manually create coupon codes in a separate window. You can tell your agent, 'Set up a 15% discount for all Pro Plan users and check out.' The process is single-step.

It's not just about reading data; it's about executing complex commercial actions—like running `create_checkout` or auditing webhooks—all through natural dialogue. Your workflow moves at the speed of thought.

Polar: 10 Tools for Commerce Operations

These tools give your agent the ability to audit product catalogs, track customer history, create discount codes, and process checkout sessions directly through Polar.

#	TOOL	DESCRIPTION
01	<code>create_checkout</code>	Generates a new payment link for specific products, allowing you to optionally apply a discount code first.
02	<code>create_discount</code>	Creates an entirely new promotional coupon code in Polar with defined amounts and durations.
03	<code>get_product</code>	Retrieves all details for a single product using its unique ID.
04	<code>list_checkouts</code>	Shows the status, associated products, and customer data for abandoned or completed checkout sessions.
05	<code>list_customers</code>	Retrieves a list of customers, letting you filter by email or name to find specific accounts.
06	<code>list_discounts</code>	Provides an audit list of all current discount codes, including their type and active status.
07	<code>list_orders</code>	Lists every order placed, detailing the customer, product, amount, and payment confirmation status for revenue tracking.
08	<code>list_products</code>	Retrieves a full list of all available products, showing if they are one-time purchases or subscriptions.
09	<code>list_subscriptions</code>	Shows the status and billing details for every customer subscription, including start/end dates.
10	<code>list_webhooks</code>	Lists all configured event endpoints to audit which services are connected to Polar.

See It in Action

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

U Show me all active subscriptions and their total monthly revenue.



I found 23 active subscriptions. Monthly recurring revenue is \$1,847 across 12 products. The top product is 'Pro Plan' at \$29/mo (18 subscribers). Would you like a breakdown by product?

U Create a 20% discount code called 'LAUNCH20' for the summer sale.



Done! I've created the discount code 'LAUNCH20' with 20% off, duration set to 'once'. Customers can now apply this code during checkout. The discount is active immediately.

U Show me all orders from the last 30 days.



I found 47 orders totaling \$3,219.50. The highest-value order was \$199.00 for 'Enterprise Plan'. Average order value is \$68.50. Would you like details on any specific order?

Frequently Asked Questions

01 How can I check if a customer has an active subscription using Polar MCP?

You use `list_subscriptions` and filter by the customer's email or name. This tool shows you their current status, billing period, and amount due.

02 Can I create a discount code with Polar MCP?

Yes, calling `create_discount` lets you define the coupon's code, whether it's percentage or fixed money off, and set its duration immediately.

03 Is listing orders accurate for revenue tracking via Polar MCP?

The `list_orders` tool shows the customer, product, amount, currency, and payment status. This provides a clean record to track total sales and average order value.

04 What if I need to check multiple products in one checkout?

You use `create_checkout` and provide all the necessary product IDs. The tool then generates a single session URL that covers everything you specified for payment.

05 Does Polar MCP help with abandoned carts?

Yes, calling `list_checkouts` shows the status of sessions, letting you see which checkouts are currently open or have expired. This helps track potential lost revenue.

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 `"polar": { "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

Polar 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 Polar. 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	Polar MCP
Server ID	019d846f-d1db-7184-a747-c57a59695da9
Platform	Vinkius Cloud for AI Agents
Endpoint	<code>https://edge.vinkius.com/{token}/mcp</code>

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/polar.