MCP SERVER | NO CODE | CLOUD HOSTED

Paddle MCP

Manage Billing and Subscriptions via AI.

Paddle provides direct access to merchant-of-record billing data. Use this MCP to manage SaaS subscriptions, inspect full transaction ledgers, and look up detailed customer financial profiles—all through your AI agent. You can check if a payment was prorated, pause an active account, or list every product sold without ever logging into the Paddle dashboard.

A+ Quality Score 100/100

billing | subscription-management | tax-compliance | payment-processing | revenue-analytics | invoice-management

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

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.

Paddle MCP

10 tools available
Cloud-hosted on Vinkius

Running a subscription business means dealing with complex billing logic: prorations, tax boundaries, and varied plan cycles. This MCP lets you pull all that financial data directly into your AI workflow. Instead of manually navigating dashboards to find out why a payment failed or when an account is due to expire, your agent handles it. You can look up specific customer accounts, check their exact tax ID status, and even list every product in your catalog with its localized pricing. Need to adjust an account? You can pause an active subscription or cancel one immediately, controlling the full lifecycle of a user's paid access. Because Vinkius hosts this MCP, you connect once from any compatible client—Claude, Cursor, Windsurf, or others—and gain immediate control over your entire billing structure and transaction history.

Core Capabilities

01 — Review Customer Financial Status

Get specific details on a customer's account to verify payment methods, tax IDs, and current subscription tiers.

02 — Manage Subscriptions Lifecycle

Pause an active plan or cancel a paid subscription, setting the effective date for when service ends.

03 — Audit Transaction History

Pull detailed lists of all billing transactions, including one-off payments, prorations, and renewal charges.

04 — Check Product Catalog & Pricing

List every product you sell and retrieve its localized pricing details for checkout or reporting.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Paddle API Key.

02 Instruct your AI client, like Cursor or Claude, to perform a billing action (e.g., 'List all active subscriptions').

03 Your agent sends the request to the MCP, which returns structured data on customer accounts, transactions, or pricing definitions.

The bottom line is you get your AI client to execute complex financial operations that used to require multiple dashboard logins and manual clicks.

Built For

This MCP is for anyone who manages recurring revenue or subscription services. If you're tired of having finance teams manually checking payment status, pulling ledger reports, or adjusting account access via a complex dashboard, this is for you.

Customer Support Agent

Checks a customer's current billing status and pauses an account without needing elevated administrative permissions.

Revenue Operations (RevOps)

Pulls structured data to evaluate tax boundaries or runs reports on transaction histories for quarterly audits.

Finance/Billing Analyst

Retrieves full ledger arrays of atomic transactions and product pricing definitions for reconciliation and reporting.

What Changes When You Connect

01 Stop jumping between tabs. You can check a customer's full payment history or pause an account directly from your chat interface using tools like `pause_subscription` and `list_transactions`.

02 Avoid manual data entry errors. Your agent pulls complex financial metrics, such as tax boundaries or prorated charges, automatically when you use `get_transaction_details`.

03 See your entire product lineup in one place. Use `list_catalog_products` to quickly verify names and retrieve pricing definitions without leaving your workflow.

04 Improve support efficiency. When a customer calls about their payment, your agent uses `get_customer_details` to pull up the necessary financial proof instantly.

05 Handle billing changes with precision. Need to cancel an account? The `cancel_subscription` tool lets you specify if termination should happen immediately or at the end of the paid term.

Real-World Applications

A user asks, 'Why was my payment only $10?'

The agent uses `get_transaction_details` to pull up the specific transaction record. It identifies that the payment of $19 was prorated down to $10 because the service period was shortened mid-cycle. The agent explains this clearly and provides a direct reference to the billing logic.

RevOps needs an audit list of all active clients.

The agent runs `list_customers` followed by `get_customer_details`. It compiles a clean, structured report showing every customer's name, account status, and tax ID boundaries for finance review.

A developer needs to test billing failure paths.

The agent uses `list_catalog_products` and then simulates trying to create a subscription using `get_subscription_details`. This allows the developer to validate edge case pricing or plan failures before pushing code.

Support needs to temporarily lock an account due to fraud.

The agent uses `pause_subscription` on the customer's specific subscription ID. It confirms that the user will remain active until their prepaid cycle expires, mitigating immediate risk without service interruption.

Patterns to Avoid

Trying to find tax rules manually

✗ AVOID

A team member copies a transaction ID and has to go through three different backend dashboards (billing, tax settings, reports) just to verify the local sales tax applied.

✓ INSTEAD

Use the `get_transaction_details` tool. It pulls all necessary billing data, including localized tax boundaries, into one structured output for immediate review.

Listing products without pricing context

✗ AVOID

A developer uses a simple product list and gets names, but has no idea what the current checkout price is or if it's taxed differently by region.

✓ INSTEAD

Use `list_catalog_products` combined with `list_catalog_prices`. This gives you both the item name and the corresponding localized pricing definition needed for accurate billing.

Assuming subscription status is always 'Active'

✗ AVOID

Support sees a user's account and assumes they are fully paying, but they don't know if the account was paused or scheduled to expire next week.

✓ INSTEAD

Always call `get_subscription_details`. This provides the exact current state—active, pending cancellation, or paused—and when the service is actually set to expire.

The Right Fit

Use this MCP if your core business process involves managing recurring payments, understanding complex billing ledgers, or making real-time changes to user accounts (e.g., pausing a subscription). You need direct, structured access to the financial state of the customer and product catalog.

Don't use it if you simply want general marketing data about your users or internal HR records; this MCP is strictly for revenue actions. If your goal is only reading non-billing user metadata (like email addresses but no billing status), a simpler CRM connector might work better. However, because this tool handles payment life cycles —from listing all payments via `list_transactions` to modifying them via `cancel_subscription` —it's the authoritative source for SaaS financial operations.

The Billing Spreadsheet Nightmare

Right now, when a client asks about their invoice, someone has to jump into the Paddle dashboard. They pull up the customer record, then they might have to open another tab to see the specific transaction ID. Then, if they want to know what tax was applied, they're looking at three different screens and cross-referencing dates and amounts in a massive spreadsheet.

With this MCP, that whole process evaporates. Your agent handles it by requesting structured data directly. You don't read spreadsheets; you get clean, actionable JSON output showing the customer details, the payment amount, and every billing rule applied.

Control Billing Changes with Paddle

You no longer have to rely on a human agent or limited UI access. You can programmatically check if an account needs its service paused using `list_subscriptions`, verify the customer's current payment method via

The billing process is now automated, verifiable, and immediate. You own the data flow; you don't wait for permission or manual intervention.

`get_customer_details`, and then execute the change with `pause_subscription`—all in one conversational step.

Paddle: 10 Tools for Billing Operations

These tools give your AI agent specific powers to handle everything from listing customers to pausing accounts and auditing complex payment histories.

#	TOOL	DESCRIPTION
01	`cancel_subscription`	Stops an active user subscription, allowing you to set the date when service termination takes effect.
02	`get_customer_details`	Retrieves specific information about a single customer account managed by Paddle.
03	`get_subscription_details`	Fetches the complete details for one particular subscription, showing its status and billing cycle.
04	`get_transaction_details`	Gets full information on a single transaction, including amounts paid, tax rates, and proration data.
05	`list_customers`	Generates a list of every customer account connected to Paddle billing.
06	`list_catalog_prices`	Lists all defined pricing structures and tax-inclusive rates for your products.
07	`list_catalog_products`	Retrieves a complete list of every product available in the catalog, including its name and details.
08	`list_subscriptions`	Provides an overview and list of all currently active or inactive subscriptions.
09	`list_transactions`	Pulls a comprehensive report listing every billing transaction that has occurred in your system.
10	`pause_subscription`	Temporarily suspends an active subscription, preventing further charges while keeping the account record intact.

See It in Action

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

U Find the subscription details for sub_01h9z2...

Subscription sub_01h9z2... is currently Active. Next billing date is set for Nov 14th, 2026. Billed at $49/month (Pro SaaS Tier). It is tied to customer ctm_01jh1. Want me to pause it?

U List our most recent revenue transactions on Paddle.

I extracted the latest transactions from Paddle Billing. 1. txn_1w2... (Paid: $19.00 USD). 2. txn_4×9... (Prorated: $10.50 USD). 3. txn_9z2... (Paid: $99.00 USD). Want me to pull specific tax boundaries applied on the $99 payment?

U Cancel subscription sub_active123 at the end of the billing cycle.

Action complete. Subscription sub_active123 has been marked for cancellation effective mapping `next_billing_period`. The user will not be charged again by the MoR, but remains active until the pre-paid term expires.

Frequently Asked Questions

01 **Can I use Paddle MCP to see all my users?**

Yes, you can list all customers using `list_customers`. This gives you a directory of every customer account that has interacted with your billing system.

02 **How do I check if a payment was prorated on Paddle MCP?**

Use the `get_transaction_details` tool. This function pulls comprehensive data for any transaction, clearly specifying whether it's a full charge or a prorated amount.

03 **Does Paddle MCP let me cancel subscriptions immediately?**

Yes, the `cancel_subscription` tool allows you to set the cancellation effective date. You can decide if the account shuts down right away or remains active until the pre-paid period expires.

04 **What is the difference between listing transactions and subscriptions?**

Listing transactions (`list_transactions`) gives you a financial record of money moving (payments, refunds). Listing subscriptions (`list_subscriptions`) tells you which plans are currently active or pending.

05 **Can I check the pricing for new products using Paddle MCP?**

You can use `list_catalog_products` to get product names and then `list_catalog_prices` to retrieve all associated localized checkout prices, ensuring you're billing correctly.

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 `"paddle": { "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

Paddle 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 Paddle. 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	Paddle MCP
Server ID	`019d75ed-dd98-7274-b6bd-e30ad73036ee`
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/paddle.