MCP SERVER | NO CODE | CLOUD HOSTED

Pagar.me MCP

Manage orders and subscriptions in Brazilian currency.

Pagar.me connects your AI agent to Brazil's payments system. Create orders using Pix, Boleto, or credit cards, manage recurring customer subscriptions, and track all transactions through natural conversation. Handle complex Brazilian financial flows without leaving your chat interface.

A+ Quality Score 100/100

payment-gateway | pix | boleto | subscription-management | transaction-tracking | order-management

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

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.

Pagar.me MCP

11 tools available
Cloud-hosted on Vinkius

Need to process sales in Brazil? This MCP lets you connect your payment processing directly into your AI client. You can initiate orders for any item using multiple methods like Pix, Boleto, or a credit card. Need recurring revenue? Set up and manage customer subscriptions right from the conversation. It also handles core tasks like registering new customers and listing past transactions to keep your records clean.

By connecting Pagar.me through Vinkius, you give your agent access to deep payment infrastructure. Your client can now track order status or create an instant Pix QR code—all without needing a web login. It's about giving your AI workflow the financial muscle it needs for the Brazilian market.

Core Capabilities

01 — Generate payments via Pix

The system creates an immediate payment order and generates the necessary QR code.

02 — Process card or Boleto orders

You can create new orders using traditional credit cards or generating a specific Boleto Bancário payment slip.

03 — Manage recurring billing

The MCP sets up and updates subscription plans for existing customers.

04 — Handle customer profiles

You can register new people or retrieve detailed history for established accounts using their CPF or CNPJ.

05 — Check order status

The agent retrieves specific details on any existing order, letting you know if it's paid, pending, or canceled.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Pagar.me Secret Key in the Vinkius catalog.

02 Your AI client sends a natural language request (e.g., 'Create a Pix order for R$100').

03 The MCP executes the payment flow, generating confirmation or necessary codes directly within the conversation.

The bottom line is that you get to manage complex sales and financial processes using simple chat commands, keeping everything contained in your AI workflow.

Built For

E-commerce managers running operations in Brazil. SaaS founders who need automated billing cycles. Any developer building an AI agent that must handle real money transactions.

SaaS Founder

They use the MCP to automatically create and renew customer subscriptions based on service usage, ensuring consistent revenue capture.

E-commerce Operations Manager

They manage the full sales cycle by generating orders via Pix or Boleto, checking payment status, and resolving transaction disputes without logging into a separate dashboard.

Developer/System Integrator

They connect the MCP to complex AI workflows to handle multi-step financial processes, like first creating a customer record and then generating an order for them.

What Changes When You Connect

01 Process payments immediately through Pix or Boleto directly from your agent, removing the need for manual QR code generation outside of the chat.

02 Handle complex billing lifecycles by using create_subscription to set up recurring revenue streams, which is crucial for any SaaS model.

03 Keep track of everything with list_orders and get_order. You can query transaction history and status checks in one conversational step.

04 Build customer records accurately using create_customer, ensuring your agent always has the proper CPF or CNPJ data before placing an order.

05 The ability to capture funds using capture_order means you can authorize a sale now and take the money later when all conditions are met.

Real-World Applications

Fulfilling a new client payment

A sales agent needs to onboard a new client. Instead of sending them a link, they ask their AI agent: 'Create an order for 3 months using the subscription tool and pay with a credit card.' The agent executes create_subscription and create_order in one go.

Handling bulk transaction checks

The ops team needs to know if payments went through yesterday. They simply ask their AI client: 'Show me the last 5 orders.' The agent uses list_orders and provides a formatted summary of payment statuses.

Troubleshooting an expired payment

A customer reports that a Boleto they paid never registered. The support agent asks: 'What is the status of order #987?' The agent uses get_order to confirm if the transaction was successful or expired.

Automating product launches

The marketing team wants to launch a limited-time sale. They instruct their AI agent: 'Generate an order for R$100 using create_pix_order.' The system instantly provides the payment details needed.

Patterns to Avoid

Using Pagar.me only for initial payments

✗ AVOID

Just running create_order and assuming the sale is final. This leaves you vulnerable if the customer doesn't pay or needs to cancel.

✓ INSTEAD

Always sequence your calls: first, use create_order; then, wait for confirmation; finally, execute capture_order only when payment is confirmed.

Manually managing subscriptions

✗ AVOID

Having a spreadsheet and updating dates or plan changes manually. This is slow, error-prone, and misses potential revenue.

✓ INSTEAD

Use create_subscription to automate the billing start. Then, periodically use get_subscription to verify status and next payment date.

Ignoring customer data requirements

✗ AVOID

Trying to place an order without knowing who the buyer is or if they are a new user. The transaction will fail later.

✓ INSTEAD

Always start by using create_customer first, capturing their CPF/CNPJ details, before attempting any major payment action.

The Right Fit

Use this MCP if your primary business function is e-commerce or SaaS billing within Brazil. You need to initiate payments via Pix, Boleto, or credit card; you manage recurring subscriptions; and you must interact with these tools using natural conversation rather than a web portal.

Don't use it if you only process refunds (use dedicated refund APIs instead), or if your primary market is outside of Brazil. If all you need is to read data without initiating payments, then simply listing orders might be enough, but this MCP gives you the full write access needed for money movement.

Dealing with Brazilian Payments Feels Like a Chore

Today, setting up an order requires jumping between your CRM, the payments portal, and sometimes even emailing a payment slip to accounting. You have to copy transaction IDs, manually calculate due dates for Boleto, and check statuses in five different tabs just to confirm if money actually moved.

With this MCP, those manual jumps disappear. Your agent handles it all: you ask it to generate an order, specify the payment method—whether that's a credit card or a custom Pix code—and get confirmation immediately. It keeps the entire flow conversational and contained.

Pagar.me MCP Gives You Complete Control Over Money Movement

You no longer need to separately track customer details, then go back to create an order, and finally use a separate tool just to check the status. The agent links these actions: it can first call create_customer, then create_order, and finally retrieve all necessary payment information using get_order.

The result is financial automation that flows naturally with your conversation. It’s not just about connecting an API; it's about making complex finance management feel simple.

Pagar.me: Payment & Order Mgmt with 11 Tools

These tools allow your agent to execute specific financial actions, from creating customer records to generating instant payment codes.

#	TOOL	DESCRIPTION
01	`create_boleto_order`	Creates a new order that requires payment using the Boleto method.
02	`cancel_order`	Marks an existing order as canceled in the system.
03	`capture_order`	Finalizes a previously authorized order, capturing the funds.
04	`create_customer`	Registers a new user account using their CPF or CNPJ identifiers.
05	`get_customer`	Retrieves all relevant details about an existing customer profile.
06	`get_order`	Fetches the full details, including items and payment status, for one specific order number.
07	`get_subscription`	Retrieves current billing cycle information for a customer's subscription plan.
08	`list_orders`	Lists the most recent orders processed through the gateway, giving you an overview of activity.
09	`create_order`	Generates a new order that accepts payments via credit card or custom payment methods.
10	`create_pix_order`	Creates an urgent, time-sensitive order specifically for immediate Pix payments.
11	`create_subscription`	Sets up a continuous billing cycle, defining the recurring plan for a customer.

See It in Action

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

U Create a Pix order for R$50.00 for customer João Silva.

Pix order created! QR Code generated. Waiting for payment.

U Show me the last 5 orders.

Found 5 orders: 1. R$ 50.00 (Pix - Paid). 2. R$ 120.00 (Card - Pending). 3. R$ 15.00 (Boleto - Expired).

U Create a monthly subscription of R$99.90 for customer 123 on plan 456.

Subscription created successfully! Status: Active. Next billing date: 2026-04-01.

Frequently Asked Questions

01 **How do I get my Secret Key?**

Log in to the Pagar.me dashboard, navigate to **Settings > API Keys** and copy your **Secret Key** (starts with `sk_...`). No complex OAuth apps, no webhooks — just connect and start processing payments.

02 **Does it support Pix and Boleto?**

Yes! Generate instant Pix QR Codes or Boletos right in the chat. Your agent handles the entire checkout flow — creating the order, retrieving the payment code, and allowing the customer to pay without ever leaving the conversation.

03 **Can I manage subscriptions?**

Yes. Just tell your AI agent to subscribe a new customer. It will register the customer, attach them to your predefined SaaS recurring plan, and confirm the active billing status — all in a single prompt without touching a dashboard.

04 **Can I manage complete customer profiles and track their transaction histories?**

Absolutely. View detailed customer records, pull entire order histories, and verify payment statuses instantly — perfect for SaaS founders, customer support teams, and digital agencies handling multiple accounts.

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 `"pagarme": { "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

Pagar.me 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 Pagar.me. 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	Pagar.me MCP
Server ID	`019d75ee-05c0-734c-961c-69ba8d6e49ad`
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/pagarme.

© 2026 Vinkius · vinkius.com · support@vinkius.com