MCP SERVER | NO CODE | CLOUD HOSTED

Checkout.com MCP for AI Agents

Process global payments, refunds, and vaulting operations.

Checkout.com MCP lets you manage complex global payment operations directly through your AI client. Handle everything from initiating payments and capturing funds to issuing refunds or voiding transactions—all without leaving your chat window. It gives your agent full control over vault instruments, payment history, and detailed billing records right where you're working.

F Quality Score 3.6/100

payment-vaulting | card-processing | transaction-management | fintech | api-payments

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/checkoutcom-1

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.

Checkout.com MCP

10 tools available
Cloud-hosted on Vinkius

Managing global money moves used to mean jumping between multiple dashboards: one for authorization, another for vaulting cards, and a third just to check the status of a refund. This MCP changes that entirely. You connect your Checkout.com account once, giving your AI client direct access to core payment logic. It handles everything from requesting initial payments and capturing those funds to securely managing customer instruments in the Vault. Need to fix an error? Your agent can retrieve detailed payment actions or audit historical billing data instantly. Everything is managed via natural conversation—whether you're debugging a complex instrument update, processing a batch of refunds, or simply checking why a transaction failed. If you already use Vinkius as your central hub for connecting different services, adding this MCP means all your financial operations run through one place.

Core Capabilities

01 — Process and finalize payments

Your agent can initiate payment requests, capture authorized funds to settle them, or void transactions before the final settlement.

02 — Manage stored customer card data

You can provision new payment instruments, update existing ones, or delete records held in the Checkout.com Vault safely.

03 — Audit and track transaction history

The system provides a full log of all actions taken on payments, helping you pinpoint exactly where an authorization succeeded or failed.

04 — Execute refunds and clawbacks

You can automatically route refunds back to the originating Visa/MC network while validating any potential clawback logic required by the transaction rules.

05 — Retrieve payment details for billing checks

It extracts specific properties of a payment, giving you all the necessary data points needed for reconciliation or debugging.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/checkoutcom-1** — connect your AI agent in three steps.

01 First, subscribe to this MCP on Vinkius and provide your Checkout.com Secret API Key.

02 Next, select whether you're working in the Sandbox (for testing) or the Live environment.

03 Finally, start asking your AI client to perform payment actions via natural language conversation.

The bottom line is that once it’s connected, your agent handles all the API calls for payments and vaulting so you don't have to write any code.

Built For

This MCP is essential for e-commerce operations teams and finance professionals. If your job involves manually checking transaction statuses, processing refunds across multiple systems, or debugging card vaulting issues without jumping between specialized dashboards, this is for you.

E-commerce Operations Manager

You use this MCP to process refunds and voids quickly. Instead of opening a separate dashboard for every payment, you just tell your agent what needs fixing.

Software Developer

You test complex payment flows in the sandbox environment using natural language or debug instrument vaulting logic without writing boilerplate API calls.

Financial Analyst

You audit transaction history and verify capture statuses for monthly reconciliation. You need a complete, auditable record of every financial movement.

What Changes When You Connect

01 Stop jumping between dashboards. Use this MCP to manage payment actions like `refund_payment` or `void_payment` entirely within your agent conversation.

02 Maintain a clean audit trail of all funds. Run `get_payment_actions` whenever you need to understand exactly why an authorization succeeded or failed months later.

03 Handle card data securely and efficiently. Use `create_instrument` and `update_instrument` to keep customer payment details current without manual data entry.

04 Speed up reconciliation. With simple calls like `get_payment_details`, you get the exact properties needed for finance teams to verify billing records quickly.

05 Reduce development time. Developers can test complex payment flows in sandbox using tools like `request_payment` and `capture_payment` via natural language prompts.

Real-World Applications

Customer calls about a missing refund

A support agent asks their AI client, 'What's the status of payment pay_123?' The agent uses `get_payment_details` and reports back immediately if the refund was successfully processed via `refund_payment`, resolving the customer query in minutes.

Debugging a failed batch capture

A developer needs to know why some payments authorized yesterday didn't settle. They use their agent to check all payment actions for a range of IDs, pinpointing the exact failure code using `get_payment_actions`.

Updating a customer's saved card details

The operations team needs to change the billing address and name on an existing customer card. They use their agent to call `update_instrument` directly, ensuring the Vault record is current for the next sale.

Preemptively canceling pending charges

Before a client leaves your service, you need to ensure all temporary holds are released. You instruct your agent to run `void_payment` on specific transactions, immediately freeing up gateway limits and preventing unnecessary fees.

Patterns to Avoid

Confusing Refunds with Voids

✗ AVOID

A user tries to 'undo' a payment by running a void when the payment has already settled and captured funds. The attempt will fail because voids only work on pending authorizations.

✓ INSTEAD

If the money was successfully taken out of the bank, you must use `refund_payment`. Only use `void_payment` if the transaction is still in an uncaptured, authorized state.

Manually updating card records

✗ AVOID

A developer manually updates a JSON file with new customer card details and forgets to run validation checks against live gateway rules.

✓ INSTEAD

Always use the `update_instrument` tool. This ensures the change is validated against real-time Vault limits and logging standards before it goes live.

Overlooking payment history

✗ AVOID

A finance team only checks the main billing ledger, missing critical details about why a specific authorization failed (e.g., insufficient funds).

✓ INSTEAD

Use `get_payment_actions`. This provides an explicit audit trail of all gateway state changes and failure reasons, helping you reconcile discrepancies.

The Right Fit

You should use this MCP if your business model relies heavily on complex payment flows: managing recurring subscriptions, handling refunds, or dealing with volatile card vaulting requirements. If you need to perform any action that touches captured funds (voids, refunds, captures), this is the tool. Don't use it if you simply need to read a static list of users; for that, a simpler directory MCP would suffice. You also don't need it if your only goal is simple payment initiation without any refund or void capability. If you are building an agent that needs deep financial audit capabilities and must handle the full lifecycle of money movement, this MCP provides the necessary depth.

Managing Checkout.com Payments: Automate Funds Capture with AI Agents

Today, processing a payment often requires multiple manual steps. You might have to log into your e-commerce platform to request the charge, switch tabs to a separate dashboard to capture the funds, and then open a third tool just to verify the transaction status. This process is slow, prone to human error, and creates significant operational friction.

With this MCP, those steps disappear. You simply tell your agent what you need—'Capture payment pay_xyz.' The entire sequence, from authorization request to final settled limit conversion, happens automatically in a single chat turn. Your agent gives you the confirmation, leaving you with immediate certainty.

Checkout.com Vaulting: Control Payment Instruments via AI Agents

Managing payment instruments manually is a nightmare of data entry and version control. You have to ensure that every card record, whether created or updated, adheres to the latest security standards and has proper logging attached.

Now, managing the Vault is conversational. To add a new card, you just ask the agent to `create_instrument`. Need to update it? Call `update_instrument` with natural language instructions. You gain absolute control over your most sensitive data without needing complex API payloads.

Checkout.com: 10 Payment Operations for AI Agents

Use these tools to manage payment flows, provision card instruments, audit transactions, and finalize funds movements via your agent's conversation.

#	TOOL	DESCRIPTION
01	`capture_payment`	Finalizes an authorized payment by locking the funds into settled limits and verifying the capture process.
02	`create_instrument`	Adds a new, highly available customer card record to the Vault system for future use.
03	`delete_instrument`	Permanently removes an instrument from the Vault after irreversible validation steps.
04	`get_payment_actions`	Lists all structured rules attached to a payment, showing its current gateway status and history.
05	`get_instrument`	Retrieves details on specific card records currently stored in the Vault logic.
06	`get_payment_details`	Pulls out all key data points related to a payment, crucial for understanding its current billing status.
07	`refund_payment`	Sends money back to the original cardholder and automatically validates any required clawback logic for the refund.
08	`request_payment`	Identifies customer records that can be used as a payment source within the Checkout.com Gateway.
09	`update_instrument`	Modifies an existing card record in the Vault, which is useful for updating holder names or expiry dates.
10	`void_payment`	Cancels a payment authorization before capture occurs, preventing unnecessary gateway fees and closing limits.

See It in Action

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

U A customer claims they were charged twice for the same service last month.

Billing Audit Report: pay_789

- **Transaction ID:** pay_789
- **Status:** Captured (Success)
- **Amount:** $120.00 USD
- **Action History:** Authorized 10/15; Captured 10/16.

We found a second transaction, pay_990. It was voided on 11/1 last week because the billing address didn't match the vault record. You should run `get_payment_actions` to confirm this history.

U I need to process a refund for an order that happened three months ago.

Refund Request Status

✅ Refund initiated for payment ID: pay_456.

- **Amount:** $75.00 USD
- **Target Network:** Visa/MC
- **Status:** Processing (Clawback logic verified).

Please allow 3-5 business days for the funds to appear on the original account.

U Can we add a new card for John Doe and make sure it's ready for payments?

Instrument Provisioning Complete

The instrument record was successfully created in the Vault.

- **Holder Name:** John Doe
- **Card Type:** Visa
- **Last 4 Digits:** 5678
- **Status:** Active and ready for use with `request_payment`.

Frequently Asked Questions

01 **How do I handle refunds using the Checkout.com MCP for AI Agents?**

You simply ask your agent to refund a specific payment ID and amount. The tool handles all the necessary validation, ensuring that any required clawback logic is applied before sending the money back to the cardholder.

02 **Can I update customer card details using this MCP?**

Yes, you can manage the Vault by updating instruments. You tell your agent which card needs changing and what the new details are, like a name or address, and it handles the secure record modification.

03 **What if I need to cancel an authorization before payment?**

You run the void function on the specific transaction ID. This cancels the authorized hold, immediately removing the limit from your gateway account and preventing fees.

04 **Does this MCP help me audit failed payments?**

Absolutely. You can use the payment actions tool to look up a payment ID and get an exhaustive list of all events that happened, telling you exactly why the bank declined or rejected the transaction.

05 **Is this MCP for AI Agents good for e-commerce billing?**

It's ideal for e-commerce. You can manage the entire payment lifecycle—from initial request to final refund—all in one place, drastically reducing manual effort and improving reconciliation accuracy.

06 **Can I test payments without using real money?**

Yes. When setting up your MCP, you choose between Sandbox or Live mode. You can safely run all tests on the sandbox environment until everything is perfect.

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 `"checkoutcom-1": { "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

Checkout.com 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 Checkout.com. 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	Checkout.com MCP
Server ID	`019d756e-9a1b-70b3-9ec7-3704563b91d3`
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/checkoutcom-1.

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