MCP SERVER | NO CODE | CLOUD HOSTED

Mollie MCP

Manage European Payments via Chat Command

Mollie MCP connects your AI agent directly to European payment processing. It lets you manage everything from creating new payments and tracking complex refunds to listing customer subscriptions—all through natural conversation. Handle iDEAL, Bancontact, credit cards, and local methods without opening a dashboard.

A+ Quality Score 100/100

payment-gateway | recurring-billing | refund-management | fintech | checkout | transaction-tracking

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

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.

Mollie MCP

7 tools available
Cloud-hosted on Vinkius

Manage all your European payments using just your voice or chat window. This MCP connects directly to your Mollie account, giving your AI client the power to perform complex financial operations typically reserved for developer dashboards. You can tell it to list recent transactions and get full status details on every payment. Need to process a refund? Your agent handles that too. Beyond payments, you can pull up customer lists or view which subscriptions are active for any user. All this happens without needing to copy an API key or navigate multiple web portals. Vinkius makes sure your AI client connects once and gets access to the full power of payment management in one place.

Core Capabilities

01 — Process payments

Initiate new transactions, specifying amounts and descriptions for immediate processing.

02 — Check transaction status

Retrieve full details on any payment, confirming its current state (e.g., paid, open, expired).

03 — List customer data

Pull a list of registered users to manage recurring billing cycles.

04 — Manage refunds

View all historical and pending refunds processed through the system.

05 — Track subscriptions

Get a detailed view of active or cancelled recurring billing plans for specific customers.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and enter your Mollie API Key in the Vinkius platform.

02 Tell your AI client exactly what you need, like 'List all refunds from last month' or 'Create a payment for €75'.

03 Your agent calls the necessary tool, sends the data to Mollie, and returns the result—whether it's a list of customers or confirmation that a payment was created.

The bottom line is you get financial operations done through conversation instead of clicking through dashboards.

Built For

E-commerce managers and finance ops staff who spend too much time jumping between payment portals to reconcile sales or issue refunds. If your job involves more than five clicks to verify a transaction, you need this.

E-commerce Operations Manager

Uses this MCP to automatically track payments and list customer subscriptions after a major sale or product launch.

Finance/Bookkeeper

Retrieves reports on all refunds processed over the last quarter and checks enabled payment methods for reconciliation.

Developer (Non-coding)

Needs to test or verify complex payment flows, like creating a payment and immediately checking its status without writing boilerplate code.

What Changes When You Connect

01 Automate reconciliation: Instead of digging through payment dashboards, use the list_refunds tool to quickly gather all refund data for month-end closing.

02 Handle customer billing changes instantly: Use list_customer_subscriptions to check a user's plan status and then create_payment if an upgrade is needed.

03 Cross-reference payments: You can list_payments to get general transaction volume, then use get_payment_details on specific IDs to find the exact failure point.

04 Verify setup quickly: Need to know what payment options you support? The list_payment_methods tool shows all enabled methods (iDEAL, Bancontact, etc.) in seconds.

05 Simplify customer onboarding: Start by listing_customers, then use create_payment to generate a trial or paid transaction for them.

Real-World Applications

Handling failed payments

A user needs to know why a payment failed. They ask their agent to get_payment_details using the transaction ID, and the MCP returns the exact error code from Mollie.

Processing annual renewals

The finance team runs into old customer records. They use list_customers to find a user, then call list_customer_subscriptions to see which plan is expiring, and finally create_payment for the renewal amount.

Auditing refund trails

An auditor needs proof of all money returned last quarter. They ask the agent to list_refunds, getting a clean, paginated report they can use immediately.

Checking payment method availability

A developer is building checkout logic for a new country and needs to know which local rails are ready. They run list_payment_methods to confirm iDEAL or Bancontact support.

Patterns to Avoid

Using multiple APIs

✗ AVOID

Writing separate scripts for payments, then another set of calls just for refunds and customer data. This creates complex maintenance overhead.

✓ INSTEAD

Connect this single MCP to your AI client. You can chain the tools—like using list_payments first, and then passing the resulting IDs to list_refunds—all within one natural language prompt.

Manual data lookup

✗ AVOID

Having to manually log into the payment gateway dashboard, search for a customer ID, copy the subscription status, and paste it into a spreadsheet.

✓ INSTEAD

Use list_customers followed by list_customer_subscriptions. Your agent handles the lookups and presents the data directly in the chat window.

Ignoring payment type specifics

✗ AVOID

Assuming all payments are credit card based, when they might actually be iDEAL or Bancontact. This causes workflow failure.

✓ INSTEAD

Use list_payment_methods to confirm the supported rails first, and then use create_payment while specifying the required method in your request.

The Right Fit

Use this MCP if your primary pain point is managing varied European payment transactions—specifically handling recurring billing, tracking refunds, or creating payments across different methods like iDEAL, Bancontact, and standard cards. It's perfect for e-commerce ops teams who need a unified view of money movement.

Don't use this if you only need to read basic CRM contact details, or if your payment process is entirely internal (no external gateway). For pure data retrieval that doesn't involve transactions, consider a general database MCP instead. If you just need to check if an email address is valid, a simple validation tool does that better.

Reconciliation and refund tracking are manual headaches today.

Right now, when the month ends, your team has to jump through hoops. You log into the payment gateway portal, pull a report for payments, then switch tabs to check refunds. If you need details on a specific customer's recurring charge, you might have to search by email address in one section and cross-reference it with another dashboard. It’s tedious clicking, copying transaction IDs, and manually matching dates.

With this MCP, the same task is conversational. You just tell your agent what you need—for instance, 'Show me all payments that failed last week.' Your AI client calls list_payments and filters the results instantly. The outcome? A clean summary right in the chat window. No dashboards, no manual cross-referencing.

Getting a full picture of your customer billing history.

Before this, tracking subscriptions meant knowing exactly which user ID to search for and then navigating through several linked tabs just to see if they were active or cancelled. Finding the status required multiple lookups across different sections of the portal.

Now, you simply ask your agent about a customer's billing history. The MCP runs list_customer_subscriptions, pulling all that data together. You get the full lifecycle—active, cancelled, next due date—all in one response.

Mollie MCP: 7 Payment Tools

Use these tools to handle every step of your payment lifecycle, from creating a new transaction to auditing old refunds, all through conversational commands.

#	TOOL	DESCRIPTION
01	`create_payment`	Creates a new transaction for money to be paid by the customer.
02	`get_payment_details`	Retrieves specific, granular details about one single payment ID.
03	`list_customers`	Pulls a full list of all registered customers in the Mollie system.
04	`list_payment_methods`	Shows which payment types (like iDEAL or PayPal) are currently enabled for use.
05	`list_payments`	Generates a paginated list of all payments that have occurred.
06	`list_refunds`	Lists every refund transaction processed in the system, helping with reconciliation.
07	`list_customer_subscriptions`	Provides a list of all active and canceled recurring billing plans for one customer ID.

See It in Action

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

U List recent payments and their statuses.

Recent payments: 5. 1) €45.00 — 'Pro Plan Monthly' (Status: Paid ✅, iDEAL). 2) €120.00 — 'Annual License' (Status: Paid ✅, Credit Card). 3) €25.00 — 'Addon Purchase' (Status: Open ⏳). 4) €10.00 — 'Donation' (Status: Expired ❌). 5) €89.00 — 'Enterprise Upgrade' (Status: Paid ✅, Bancontact).

U Create a payment of €50 for a premium upgrade.

Payment created! ✅ ID: tr_7UhSN1zuXS. Amount: €50.00. Description: 'Premium Upgrade'. Checkout URL: https://www.mollie.com/checkout/... Redirect URL: https://myapp.com/success. Status: Open. Waiting for customer to pay.

U Show enabled payment methods and list refunds.

Enabled methods: 6. iDEAL ✅, Credit Card ✅, Bancontact ✅, PayPal ✅, SEPA Direct Debit ✅, Klarna ✅. Refunds (last 30 days): 3. 1) €12.00 (tr_abc, Refunded ✅). 2) €45.00 (tr_def, Processing). 3) €8.50 (tr_ghi, Refunded ✅).

Frequently Asked Questions

01 **How do I use Mollie MCP to check if a payment succeeded?**

Use get_payment_details with the specific transaction ID. This tool returns granular status information, letting you know exactly if the payment was paid, failed, or is still open.

02 **Can I list all available payment methods using Mollie MCP?**

Yes, calling list_payment_methods shows every enabled method your store accepts. This is useful for confirming local rails like iDEAL are active.

03 **What if a customer cancels their subscription? Can I see that with Mollie MCP?**

Use list_customer_subscriptions. It provides the history of billing plans, showing both active and cancelled status for any given user.

04 **Is Mollie MCP only for creating payments?**

No, it's much broader. You can also use tools like list_refunds to manage money that has already left your account for reconciliation purposes.

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 `"mollie-alternative": { "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

Mollie 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 Mollie. 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	Mollie MCP
Server ID	`019dd127-8b49-7374-bed8-e697c7b54c7a`
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/mollie-alternative.