MCP SERVER | NO CODE | CLOUD HOSTED

# Marqeta MCP

## Control card issuance, funding, and user lifecycles.

Marqeta MCP manages your entire card issuing infrastructure through natural conversation. Your agent can provision physical or virtual cards, create new business accounts, manage user lifecycle states, control funding sources via ACH and GPA orders, and run payment simulations for complex fintech workflows.

A+ Quality Score 98.33/100

card-issuing | fintech-api | virtual-cards | payment-infrastructure | gpa-orders

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

SCAN TO VIEW

# The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

# Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

# Marqeta MCP

31 tools available
Cloud-hosted on Vinkius

Managing a modern payments platform involves dozens of moving parts—user provisioning, fund transfers, card status updates, and compliance checks. Instead of navigating multiple dashboards or writing custom API calls every time you need to onboard a new client or issue a batch of cards, this MCP handles it all. Your agent speaks the language of payment infrastructure, allowing you to manage everything from creating a core business entity to funding an account with a simple prompt. When you connect this specialized capability through Vinkius, you get one single access point to execute complex financial actions without writing any code. It's about getting the job done, fast and accurately.

## Core Capabilities

**01 — Provisioning User Accounts**

Create new business entities or individual users needed for card program enrollment.

**02 — Issuing and Managing Cards**

Generate physical or virtual payment cards, and define the specific product rules that govern how those cards can be spent.

**03 — Funding Accounts & Compliance**

Move money into General Purpose Accounts (GPA) using ACH funding sources and perform necessary KYC verification checks.

**04 — Monitoring Account Status**

Retrieve the current balances, card details, or full history for any user or business token.

**05 — Simulating Financial Transactions**

Test complex payment flows by simulating authorizations, clearings, and reversals before making live transactions.

## One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to the Marqeta MCP and provide your required Marqeta Application Token, Admin Access Token, and Base URL (Sandbox or Production).

02 Your agent connects these credentials to the Vinkius ecosystem, granting it access to the full range of card issuing tools.

03 You simply tell your agent what you need—for example, 'Fund user X' or 'Issue a new virtual card for Y'—and it executes the necessary sequence of actions.

The bottom line is that you get to manage complex financial workflows through plain conversation, eliminating manual API calls and dashboard switching.

## Built For

This MCP is essential for anyone working in digital payments who needs to move beyond simple data retrieval. It targets the Payments Operations Manager who's tired of manually calling multiple APIs just to onboard a single user, or the FinTech Developer building complex payment logic that requires programmatic card and funding management.

**Payments Operations Manager**

Handles new client onboarding and account maintenance. They use this MCP to execute KYC checks and create initial business accounts without needing developer intervention.

**FinTech Developer**

Builds payment processing logic into applications. They rely on the MCP to programmatically issue cards, simulate transactions, and manage card product definitions.

**Product Manager (Payments)**

Defines new features like funding mechanisms or user state changes. They use it to test complex flows by executing GPA orders and creating specific funding sources.

## What Changes When You Connect

01 Automate full onboarding cycles. Instead of manually creating accounts and then running KYC checks, your agent handles the entire process by executing 'create_business' followed immediately by 'perform_kyc', saving dozens of clicks.

02 Simplify card provisioning. You can define a new payment structure using 'create_card_product' once, and then instantly issue multiple physical or virtual cards using 'create_card' without repeating the setup steps.

03 Handle complex funding logic. Need to top off an account? Your agent manages it by first checking balances with 'get_balances', and then initiating a transfer via 'create_gpa_order'.

04 Test payment flows safely. Before going live, you can use tools like 'simulate_authorization' or 'simulate_clearing' to test transaction logic without spending real money.

05 Manage account status changes instantly. Instead of logging into the UI and clicking a dropdown, your agent executes transitions—like using 'create_user_transition' or 'create_card_transition'—to change states immediately.

## Real-World Applications

**Onboarding a New Merchant Account**

A Payments Operations Manager needs to bring a new merchant online. They ask their agent to first 'create_business', then run 'perform_kyc' on the entity, and finally 'get_balances' to verify initial funding readiness.

**Re-issuing Cards for Deactivated Users**

A developer needs to replace a lost card. They ask their agent to first identify all cards belonging to the user using 'list_cards_by_user', and then execute 'create_card' with the correct product rules.

**Executing Scheduled Funding Runs**

The finance team needs to ensure funds are ready for a large batch of users. They instruct their agent to check balances using 'get_balances', and then use 'create_gpa_order' for every account that falls below the minimum threshold.

**Debugging Payment Failures**

A developer encounters a payment failure in testing. They ask their agent to retrieve the specific card details using 'get_card', and then use 'simulate_reversal' to understand what went wrong without impacting live funds.

## Patterns to Avoid

### Assuming a user is ready for funding

✗ AVOID

Sending an ACH order immediately without verifying the account status or available balance. This wastes time and might fail due to outdated tokens.

✓ INSTEAD

Always check the current state first. Use 'get_user' or 'get_balances' before attempting a transfer with 'create_gpa_order'. If you need to change the user status, use 'create_user_transition'.

### Issuing cards without rules

✗ AVOID

Trying to create a card using only basic details. This often fails because payment systems require predefined product parameters (e.g., spending limits).

✓ INSTEAD

First, define the entire system structure by calling 'create_card_product'. Then, use that resulting product ID when you call 'create_card' to ensure it adheres to defined rules.

### Manually changing user status in multiple places

✗ AVOID

When a user relationship ends, manually updating the status across different backend systems. This is slow and highly prone to human error.

✓ INSTEAD

Use the specialized transition tools. Simply ask your agent to 'create_user_transition' or 'create_card_transition'. The MCP handles the state change consistently.

## The Right Fit

Use this Marqeta MCP if your job revolves around executing specific, high-stakes financial actions: issuing payments, provisioning accounts, managing compliance status, or moving funds. Think card lifecycle management in its entirety. However, don't use it if you only need to send a simple text message (use a messaging tool) or pull general, unstructured data from an unrelated

source. If your goal is merely to read documentation or summarize policy documents, this MCP won't help. You must be able to map the task directly to a financial action like 'create_gpa_order,' 'get_card,' or 'perform_kyc.'

## Managing card programs today means clicking through half a dozen dashboards.

Lately, setting up a new payment flow feels like juggling. You have to go check the user's status in one tab, define the product rules in another, then run KYC verification on a third screen, and finally initiate the funding order by copying tokens from four different places. The sheer amount of context switching slows down everything.

With this MCP, you talk to your agent like talking to a colleague who knows the system inside and out. Instead of clicking through five tabs and copy-pasting three tokens, you simply prompt: 'Onboard user X.' You get the whole sequence executed—from verifying identity with 'perform_kyc' to provisioning the card itself.

## Marqeta MCP gives you total control over funding and account status.

The biggest manual pain points are the repetitive checks. You spend time verifying if a user is active enough to receive funds, or checking if a card has been properly transitioned from a temporary state before activation. These manual audits eat up valuable operational time.

Now, you can ask your agent to check balances using 'get_balances' and simultaneously verify the card status with 'get_card.' It’s not just about getting data; it’s about executing complex checks across multiple tools in a single conversation.

# Marqeta: 20 Tools for Payment Control

These tools let you manage every step of card issuance, from creating business entities to funding accounts and simulating complex payment transactions.

| # | TOOL | DESCRIPTION |
|---|---|---|
| 01 | `create_ach_funding_source` | Establishes a new source for Automated Clearing House (ACH) funds. |
| 02 | `create_auth_control` | Sets up specific controls to manage transaction authorizations. |
| 03 | `create_business` | Registers and creates a new business entity within Marqeta. |
| 04 | `create_card_product` | Defines the rules and parameters for an entire type of payment card that can be issued. |
| 05 | `create_card` | Generates a single, specific physical or virtual card for use. |
| 06 | `create_card_transition` | Moves an existing card to a new operational status (e.g., from Active to Suspended). |
| 07 | `create_direct_deposit_transition` | Updates the status of a direct deposit record. |
| 08 | `create_gpa_order` | Initiates an order to transfer funds into a user's General Purpose Account (GPA). |
| 09 | `create_payment_card_funding_source` | Sets up a source specifically for funding payment cards. |
| 10 | `create_pin_control_token` | Creates a secure token used to manage and control card PINs. |
| 11 | `create_program_funding_source` | Sets up funding sources for an entire payment program. |
| 12 | `create_user` | Registers and creates a new user account holder in the system. |
| 13 | `create_user_transition` | Changes a user's overall status (e.g., Active to Closed). |
| 14 | `create_velocity_control` | Implements rules that limit how frequently an account can be used. |
| 15 | `create_webhook` | Creates a webhook endpoint to receive automated notifications from Marqeta. |

| # | TOOL | DESCRIPTION |
|---|---|---|
| 16 | get_balances | Retrieves the current financial balance for either a user or a business account. |
| 17 | get_business | Fetches all details about an existing Marqeta business using its unique token. |
| 18 | get_card | Retrieves specific information and status for a single card by its token. |
| 19 | get_direct_deposit | Fetches the details of a recorded direct deposit transaction. |
| 20 | get_user | Retrieves all data associated with a specific user account token. |
| 21 | list_cards_by_user | List all cards for a specific user |
| 22 | list_users | List Marqeta users |
| 23 | perform_kyc | Perform KYC verification for a user |
| 24 | ping_webhook | Ping a webhook |
| 25 | provision_apple_pay | Provision a card for Apple Pay |
| 26 | provision_google_pay | Provision a card for Google Pay |
| 27 | reveal_pin | Reveal a card PIN |
| 28 | simulate_authorization | Simulate an authorization |
| 29 | simulate_clearing | Simulate a clearing transaction |
| 30 | simulate_reversal | Simulate a reversal |
| 31 | update_pin | Set or update a card PIN |

# See It in Action

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

**U** List the last 10 users registered in my Marqeta environment.

I've retrieved the users. Here are the most recent 10 account holders, including 'John Doe' (Token: user_abc123) and 'Jane Smith' (Token: user_xyz789).

**U** Issue a new virtual card for user token 'user_888' using card product 'prod_999'.

Processing card issuance... Success! A new virtual card has been created for user_888. The card token is 'card_555666' and it is currently active.

**U** Fund $50.00 USD to user 'u_789' from funding source 'fs_000'.

Creating GPA order... The funding of $50.00 USD to user u_789 was successful. Order token: gpa_order_12345.

# Frequently Asked Questions

01 **How do I list all users using the Marqeta MCP?**

You use the 'list_users' tool to retrieve a list of all registered user tokens. This is useful for bulk operations or auditing who has access.

02 **What does 'create_gpa_order' do in Marqeta MCP?**

The 'create_gpa_order' tool initiates a funding transaction, moving money into an account holder's General Purpose Account (GPA). This is the primary way to fund user accounts.

03 **Can I add Apple Pay support using Marqeta MCP?**

Yes. You use 'provision_apple_pay' to link a card token to Apple Pay, making the physical or virtual card usable in that ecosystem.

04 **How does 'simulate_authorization' work with this MCP?**

This tool lets you test how a payment will behave without actually moving money. It simulates an authorization request so you can validate your application logic safely before going live.

05 **I need to change a user from Active to Suspended; which tool do I use?**

You should use the 'create_user_transition' tool. This ensures that the state change is logged correctly and applied system-wide, preventing operational errors.

# 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 `"marqeta": { "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

# Marqeta 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 Marqeta. 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 | Marqeta MCP |
| Server ID | `019e38bd-5a62-734b-b823-dd939e44523b` |
| 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/marqeta.