MCP SERVER | NO CODE | CLOUD HOSTED

Brankas MCP for AI Agents

Manage payments and financial data across Southeast Asia's open finance ecosystem

Brankas connects your AI client to Southeast Asia's open finance network. Your agent can handle complex money movements and data retrieval, managing everything from accepting direct payments and initiating payouts to aggregating real-time bank statements and balances.

F Quality Score 3.6/100

open-finance | api-integration | disbursements | account-to-account | financial-data | southeast-asia

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

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.

Brankas MCP

8 tools available
Cloud-hosted on Vinkius

Need to manage financial operations across multiple banks in Southeast Asia? This MCP gives your AI client the power to run full payment cycles through Brankas. Instead of switching between dashboards or manually calling APIs, you talk to your agent, and it handles the money moves. You can initiate payments by creating checkout sessions for instant deposits, and then automatically send payouts using both inter-bank and intra-bank transfers. Beyond moving cash, your agent retrieves bank account balances, tracks transaction history, and pulls identity data from consented users. Because Vinkius is the premier catalog for these kinds of integrations, you connect once to access all this financial utility through any compatible client.

It simplifies complex finance tasks into natural conversation.

Core Capabilities

01 — Accepting Direct Payments

Your agent can create checkout sessions to accept instant account-to-account payments and track the status of incoming funds.

02 — Initiating Payouts

The MCP allows your agent to send payouts automatically, supporting both inter-bank and intra-bank transfers, and monitors their progress until completion.

03 — Retrieving Financial Balances

Your AI client pulls real-time bank account balances linked to the connected accounts on demand.

04 — Accessing Transaction History and Identity Data

You can retrieve full bank statement data, view transaction history, and pull identity verification details from consented users.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to the Brankas MCP on Vinkius.

02 Provide your specific Brankas API Key and select your operating environment (sandbox or production).

03 Use any compatible AI client to begin managing payments, payouts, and financial data via natural conversation.

The bottom line is, you connect the credentials once; your agent handles all the complexity from there.

Built For

Fintech developers, operations managers, and compliance officers need this MCP. If your job involves moving money or verifying financial data across different banks in Southeast Asia, you'll want this. It eliminates the manual cross-referencing between payment gateways and internal systems.

Fintech Developer

Building automated payout features that require coordinating transfers, status checks, and data retrieval using your agent.

Financial Operations Manager

Executing complex cash flow operations, such as running payroll or vendor disbursements, by automating inter-bank payouts.

Compliance Analyst

Consolidating user data and financial statements for auditing purposes, ensuring all transactions are tracked from initiation to completion.

What Changes When You Connect

01 Instant cash flow management: Use `create_checkout` to immediately accept account-to-account funds without manual intervention.

02 Full payout automation: Automate disbursements by initiating transfers using both `inter_bank_transfer` and `intra_bank_transfer` tools.

03 Real-time financial oversight: Get accurate, up-to-the-minute fund status checks via the `get_balance` tool, giving instant visibility into cash positions.

04 Deep audit trail access: Pull comprehensive transaction data using `get_statement`, providing a full record for compliance and auditing purposes.

05 Payment tracking assurance: Verify every money movement—incoming payments with `get_transaction` or payouts with `get_transfer_status`.

Real-World Applications

Processing Vendor Payments at Scale

A logistics company needs to pay 50 vendors daily. Instead of logging into multiple banking portals, the agent uses `inter_bank_transfer` and `intra_bank_transfer` tools sequentially to execute all required payouts based on a single spreadsheet input.

Onboarding New Merchants

A payment processor needs to verify a new merchant's financial standing. The agent first uses `get_identities` and then runs `get_balance` to confirm the merchant's identity and current available funds before activating services.

Handling Refunds and Chargebacks

When a customer initiates a refund, the agent doesn't just process it; it uses `create_checkout` to track the incoming payment status (`get_transaction`) and confirms the original funds were received before initiating the return payout.

Quarterly Compliance Auditing

An accountant needs a complete picture of account activity. The agent pulls `get_statement` data for the last quarter, aggregates it with identity records (`get_identities`), and presents a single, cohesive report.

Patterns to Avoid

Assuming all payments are instant.

✗ AVOID

Relying only on payment initiation without checking the status. For example, assuming an inter-bank payout succeeded just because it was submitted hours ago.

✓ INSTEAD

Always confirm transfer completion. Use `get_transfer_status` to monitor payouts and use `get_transaction` for incoming payments until they show a final 'SUCCESS' state.

Using only one banking system.

✗ AVOID

Only checking balances or statements from the primary bank account, missing funds held in secondary linked accounts used for different payment streams.

✓ INSTEAD

Use `get_balance` to check all linked accounts and ensure you pull comprehensive data using `get_statement` across every necessary source.

Over-relying on manual data entry.

✗ AVOID

Manually copying transaction IDs or account numbers from emails into a payment system, which introduces human error into the payout process.

✓ INSTEAD

Let your agent handle it. Use `inter_bank_transfer` or `intra_bank_transfer` and let the MCP manage the data flow directly from the source.

The Right Fit

Use this MCP if your core business logic requires coordinated, multi-stage financial operations: accepting funds, verifying identities, checking balances, *and* sending payouts. If you only need to read simple data (like fetching a single balance), other basic data connectors might suffice. However, if the workflow involves moving money—whether it's an immediate checkout or a delayed payout—you must use Brankas because it handles both sides of the transaction lifecycle. Don't use this MCP if your needs are limited to simple CRM updates; you need its payment tools for any flow that changes who owns the cash.

Brankas Payments: Automating Incoming Funds and Disbursements

In traditional finance ops, handling incoming payments means jumping between multiple bank portals to confirm funds have cleared. Sending payouts is worse; you often need separate workflows for inter-bank versus intra-bank transfers, which requires manual status checks and error resolution.

With the Brankas MCP, your agent handles this entire flow naturally. You can tell it, 'Send $500 to Vendor X.' It manages both the initiation of the transfer (using `inter_bank_transfer`) and then automatically monitors its success until you get confirmation that the funds landed.

Brankas Financial Data: Centralizing Statements and Balances

Today, getting a full picture of a client's finances means downloading separate CSV reports for bank statements and then manually cross-referencing those dates with identity verification records. It's slow, prone to versioning errors, and requires hours of spreadsheet cleanup.

Now your agent pulls all that data instantly. You ask it for the latest statement (`get_statement`), and it combines that transaction history with the current `get_balance` figure into one clean output. You get a single source of truth.

Brankas MCP: 8 Tools for Financial Data & Payments in SEA

Use these tools to create checkouts, get balances, pull statements, send inter-bank payouts, and manage the full lifecycle of financial transactions.

#	TOOL	DESCRIPTION
01	`create_checkout`	Creates a new session to accept direct payments into the connected account.
02	`get_balance`	Retrieves the current balance for a linked bank account.
03	`get_identities`	Gets stored identity data associated with the connected user.
04	`get_statement`	Pulls detailed transaction history and statement data for a linked account.
05	`get_transaction`	Checks the current status of an incoming direct payment transaction.
06	`get_transfer_status`	Retrieves the real-time status of a previously initiated disbursement transfer (payout).
07	`inter_bank_transfer`	Sends money out via an inter-bank payout, linking two separate financial institutions.
08	`intra_bank_transfer`	Sends money out within a single bank's network (intra-bank payout).

See It in Action

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

U I need to pay my team's quarterly bonuses out immediately.

Payout Summary: Q2 Bonuses

- **Total Payout:** 15,000 IDR
- **Method Breakdown:**

* Inter-Bank (External): 12,000 IDR (Status: SUCCESS)
* Intra-Bank (Internal): 3,000 IDR (Status: PENDING)

- The transfer for the remaining 3,000 IDR is scheduled and should complete within the hour. You can check its status using the transaction reference provided above.

U Can you confirm if my checkout payment from yesterday went through?

Payment Status Check: txn_abc123

- **Amount:** 500,000 PHP
- **Status: COMPLETED** ✅
- **Timestamp:** 2026-06-12 14:30 UTC
- The funds are confirmed and the transaction ID is valid for records. No further action needed.

U What was my total cash balance and what were my last few transactions?

Account Snapshot

- **Current Balance:** 8,520,000 IDR
- **Last 3 Transactions (from statement):**

1. Deposit: +10,000,000 IDR (Source: Payroll)
2. Withdrawal: -500,000 IDR (Source: Vendor Payment)
3. Transfer: -300,000 IDR (Destination: Tax Authority)

Frequently Asked Questions

01 **How does the Brankas MCP handle complex payouts to multiple banks?**

The Brankas MCP manages all payout types automatically. Whether you need an inter-bank or intra-bank transfer, your agent initiates it and monitors the status until funds are confirmed deposited.

02 **Can I use this with my existing accounting software?**

Yes, by having the AI client pull transaction data via `get_statement`, you feed clean, structured financial reports directly into your accounting platform for easy reconciliation and auditing.

03 **Is Brankas MCP only for Southeast Asian payments?**

The functionality is focused on open finance operations across the region. It provides deep integration with local bank systems necessary for accurate payment processing in this market.

04 **What if a payout fails? Can the Brankas MCP help me track it?**

Absolutely. The MCP lets your agent check the transfer status using specific tools, confirming if a payout is pending or failed and providing visibility into why.

05 **Does Brankas MCP let my AI client accept payments from customers?**

Yes, you can create checkout sessions to accept instant account-to-account payments. Your agent tracks the transaction ID until those funds are fully received and cleared.

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 `"brankas": { "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

Brankas 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 Brankas. 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	Brankas MCP
Server ID	`019d7562-a02f-737e-b831-d288ee122d3c`
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/brankas.