MCP SERVER | NO CODE | CLOUD HOSTED

Net Balance Minimizer MCP for AI Agents

Automating Group Expense and Debt Settlement

The Net Balance Minimizer resolves group debts by calculating who owes what and generating a minimal settlement plan. It takes complex shared expenses—like trips or team dinners—and figures out the fewest transactions needed to clear everyone's accounts back to zero.

A+ Quality Score 100/100

debt-settlement | expense-splitting | group-expenses | transaction-minimization | financial-tools

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JULY 2026
vinkius.com/mcp/net-balance-minimizer

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.

Net Balance Minimizer MCP

3 tools available
Cloud-hosted on Vinkius

Managing shared finances can be a mess. When a group pays for things together, tracking receipts and figuring out who owes whom turns into an accounting headache. This MCP solves that by acting as a specialized calculation engine for group spending. You feed it the raw data—all the expenses and who paid what—and the tool analyzes the entire financial picture. It doesn't just tell you if someone is over or under; it uses advanced math to generate the most efficient path back to zero balance, reducing weeks of mess into a few clicks. When you connect this MCP through Vinkius, your AI agent handles all the heavy lifting, creating a clear, actionable transaction list for everyone involved.

Core Capabilities

01 — Calculate individual net balances

The tool determines exactly how much money every participant needs to receive or pay out based on shared expenses.

02 — Summarize total group spending

It aggregates all the raw expense entries, giving you a single summary of the total amount spent by the group.

03 — Generate optimized payment transactions

This is the core function: it creates a precise list of transfers that settles every debt in the fewest possible steps.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/net-balance-minimizer** — connect your AI agent in three steps.

01 First, you input all raw financial data—the expenses paid and who participated. The MCP first summarizes total group spending.

02 Next, it calculates each person's net balance to pinpoint exactly who is owed money and who needs to pay up.

03 Finally, the tool generates a streamlined list of transactions that clears every debt with minimum transfers, giving you the settlement plan.

The bottom line is your AI agent takes messy expense lists and spits out a simple, actionable payment schedule.

Built For

Anyone who has ever paid for dinner with friends or organized a group trip knows the pain: the spreadsheet mess. This MCP helps event planners, small businesses managing shared vendor costs, and travel groups settle up without arguing over receipts.

Group Trip Organizer

They input all hotel payments and meal receipts into the agent. The tool then generates a payment plan, telling everyone exactly who pays whom to cover costs.

Small Team Lead

The lead collects vendor invoices for shared tools or equipment rentals. They run the MCP to get a minimal transaction list, ensuring the team's finances are settled quickly and fairly.

Freelance Project Coordinator

They track joint costs like software subscriptions or materials purchased by different contractors. The tool calculates the precise split and minimizes administrative effort.

What Changes When You Connect

01 Stop spending hours tracking receipts. By running the `expense_summary` tool, your agent instantly gets a clean total of all group money spent.

02 Eliminate payment disputes. The MCP calculates individual net balances using the `net_balances` tool, so everyone knows their exact financial standing without manual calculation.

03 Cut down on bank transfers. Instead of generating ten separate payments, the `settlement_transactions` tool creates a minimal list of transactions, saving time and effort for all parties.

04 Go beyond simple splitting. This MCP handles complex debt scenarios, not just dividing costs equally, but settling based on actual contributions.

05 Gain immediate clarity. You move from 'who owes who' chaos to a single, clean settlement plan that your agent delivers instantly.

Real-World Applications

The post-trip accounting disaster

Four friends take a weekend trip. One person booked the hotel, another paid for gas, and two people covered meals. Instead of creating a massive spreadsheet, they ask their agent to use the MCP tools. The system processes all expenses, calculates net balances, and spits out three simple transfers: John pays Sarah $50, Mike pays David $20, etc.

Organizing family reunions

A large extended family pays for venue rentals and catered meals. The organizer feeds all receipts into the MCP. It calculates everyone's share and uses `settlement_transactions` to create a clean payment schedule, ending arguments over who paid what.

Shared vendor costs for small business

A consulting team jointly purchases a specialized software license. They feed the expenses into their agent. The MCP summarizes the total cost and runs `net_balances`, showing that each person owes exactly $150, and then generates the final payment plan.

Patterns to Avoid

Calculating balances manually

✗ AVOID

Trying to use multiple spreadsheets or basic calculator functions to track dozens of receipts and payments. This process is slow, error-prone, and requires constant manual cross-referencing.

✓ INSTEAD

Feed the data into your agent and let it run `net_balances` first. Then, immediately follow up by calling `settlement_transactions`. This two-step process guarantees a correct, minimal payment plan.

Ignoring transaction efficiency

✗ AVOID

Accepting an arbitrary settlement list that requires many small payments (e.g., five people paying $10 each). This maximizes administrative overhead and friction.

✓ INSTEAD

Always use the dedicated `settlement_transactions` tool after calculating balances. It's designed to group debts into the fewest possible transfers, saving everyone time.

Missing total cost context

✗ AVOID

Only focusing on individual payments without knowing the overall spending picture. This leads to disputes over which expenses were even included in the calculation.

✓ INSTEAD

Start by summarizing all data using `expense_summary`. This gives you a verifiable, single source of truth for the total cost before running any balance calculations.

The Right Fit

Use this MCP if your core problem is debt settlement among a defined group of people based on shared expenses. If you just need to split costs equally (e.g., dividing a bill by four), basic math works, but this tool provides the necessary nuance for complex scenarios involving uneven payments and varying contributions. Don't use it if you are trying to manage inventory or track assets; that requires a different type of database connection. If your goal is simply to list all expenses without knowing who pays what, using only `expense_summary` is enough. But if you need the final 'who pays whom' answer, this MCP is mandatory.

Net Balance Minimizer: Solving Group Debt Settlement with Finance

Right now, settling shared finances means opening multiple documents, cross-referencing who paid what for the hotel versus the groceries. You spend time tracking payments and manually adjusting numbers in a spreadsheet just to figure out if you're owed twenty dollars or thirty. It’s tedious, frustrating accounting work.

With this MCP, your agent handles the whole mess. Give it the receipts, and it instantly runs the calculations needed to find net balances and generate a single, perfect payment list. You get crystal clear financial accountability without touching an equation.

Net Balance Minimizer: Streamlining Expense Splitting in Group Finances

The manual process involves multiple steps: first, summing up every single expense item; second, calculating each person's share of that total; and finally, figuring out how to combine those shares into a few manageable payments. This is pure administrative drag.

This MCP automates the entire sequence. It moves you past simple splitting by determining the minimum number of transactions required for settlement. You don't just get an answer; you get the most efficient way to pay it.

3 Tools for Group Expense and Debt Settlement Finance

These tools allow your agent to calculate net balances, summarize all expenditures, and generate optimized payment plans.

#	TOOL	DESCRIPTION
01	`net_balances`	Calculates the final financial balance for each participant in a group.
02	`expense_summary`	Aggregates and summarizes all recorded expenditures from the group.
03	`settlement_transactions`	Generates the final, optimized list of transfers needed to settle all outstanding debts.

See It in Action

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

U We paid $600 total for a weekend trip with four people, but Jane covered the car rental and Tom covered dinner. Figure out who pays whom.

Settlement Plan Generated

Based on the expenses, here is how you can settle up:

1. **Tom transfers $75 to Sarah.** (Covers his overpayment for meals)
2. **Mike transfers $50 to Jane.** (Balances out car rental costs)
3. **Sarah transfers $25 to David.** (Final adjustment).

Total transactions required: 3.

U Summarize these expenses from the last month's team outing: [{'payer_name': 'Alex', 'amount': 400, 'split_between': ['A', 'B']}], [{'payer_name': 'Ben', 'amount': 200, 'split_between': ['C', 'D']}]

Group Spending Summary

- **Total Expenditure:** $600.00
- **Number of Entries:** 2
- **Breakdown:**

* Alex paid: $400 (Split between A & B)
* Ben paid: $200 (Split between C & D)

This figure provides the total amount that needs to be settled among the group members.

U We have these current balances: Sarah is owed $13.50, Tom owes $18.00, and David owes $7.50. Give me the payment plan.

Minimal Payment Schedule

To settle all accounts with three transfers:

- Tom sends **$13.50** to Sarah.
- David sends **$4.50** to Sarah.
- Tom sends remaining **$3.50** to David.

Frequently Asked Questions

01 **How does the Net Balance Minimizer MCP handle complex shared expenses?**

It takes all your receipts and figures out who owes what, regardless of how many people are involved. It doesn't just calculate if someone is over or under; it designs a complete financial map for you.

02 **Is the Net Balance Minimizer MCP better than just splitting the total bill equally?**

Yes, significantly. Basic equal splitting fails when contributions are uneven—for example, if one person paid $800 and another only paid $50. This MCP accounts for actual payments, not just an even split.

03 **What kind of data does the Net Balance Minimizer MCP need to run?**

You need a list of expenses, including who paid the money and which participants shared that cost. The tool needs both the payer name and all involved parties for accurate results.

04 **Does this MCP provide an actual payment schedule or just balances?**

It provides both. First, it calculates individual net balances; second, it generates a transaction list that tells you exactly who sends money to whom, minimizing the number of payments required.

05 **Can I use Net Balance Minimizer MCP for small business team costs?**

Absolutely. If your team pays jointly for software or supplies, this MCP calculates the precise split and generates a clean payment schedule, eliminating messy internal accounting.

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 `"net-balance-minimizer": { "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

Net Balance Minimizer 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 Net Balance Minimizer. 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	July 2026
MCP Server	Net Balance Minimizer MCP
Server ID	`019f1fe2-2804-7023-aab7-ef341affab5e`
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/net-balance-minimizer.