MCP SERVER | NO CODE | CLOUD HOSTED

Couples Shared Account Engine MCP for AI Agents

Automating Fair Financial Splits and Proportional Expense Sharing

The Couples Shared Account Engine calculates fair and proportional expense sharing between partners based on their respective incomes. It determines who owes what for joint bills, tracks surpluses after shared costs, and figures out the exact transfer amounts needed to keep household accounts balanced.

A+ Quality Score 100/100

finance | budgeting | couples | expenses | automation | shared-accounts

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JULY 2026
vinkius.com/mcp/couples-shared-account-engine

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.

Couples Shared Account Engine MCP

4 tools available
Cloud-hosted on Vinkius

When you and a partner share bills, figuring out who owes what can be a headache that leads to arguments. This engine cuts through the complex math of proportional finance. It ensures every person contributes their fair share relative to their earnings, so nobody feels unfairly burdened by joint expenses. You use this MCP to calculate initial contribution weights based on income, then determine exact dollar amounts owed for specific shared costs. If someone pays upfront, the engine calculates the precise transfer needed to settle the balance. It even checks the financial stability of the partnership after all obligations are met. Since Vinkius hosts this MCP, your AI client connects once and gains immediate access to this complex finance tool alongside thousands of others.

Core Capabilities

01 — Determine contribution weight by income

It calculates what percentage of the total household finances each partner should contribute based on their individual earnings.

02 — Calculate specific obligations for shared costs

The tool determines the exact dollar amount each partner is responsible for contributing to joint expenses.

03 — Figure out necessary account transfers

It calculates the precise money transfer required between partners to fully reconcile a set of shared bills.

04 — Check remaining financial balance

You can evaluate the remaining surplus or deficit for each partner after all joint obligations and personal costs are accounted for.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/couples-shared-account-engine** — connect your AI agent in three steps.

01 First, feed the engine your partners' incomes. This calculates the financial weight of each person to determine their fair contribution proportion.

02 Next, input all shared expenses and individual spending. The tool then uses these figures to calculate exactly how much money each partner owes for the joint costs.

03 Finally, run a reconciliation check. It tells you the precise transfer amount one partner needs to send to another to settle up completely.

The bottom line is that it automates the messy arithmetic of shared finances, giving you clear dollar amounts and proportions instead of guesswork.

Built For

This MCP helps anyone managing joint accounts or complex household budgets. It's essential for couples, co-signers, or roommates who need an objective, mathematical way to settle shared bills without conflict.

Household Manager

Uses the engine whenever a large joint expense comes up—like mortgage payments or appliance purchases—to immediately figure out proportional splits.

Financial Planner (Personal)

Models different income scenarios to advise clients on optimal financial contribution strategies for shared living situations.

Roommate/Co-signer

Needs a quick, unbiased calculation of who owes what when utility bills or rent are paid by only one person.

What Changes When You Connect

01 Eliminate arguments over money. The engine provides objective, math-backed splits instead of relying on gut feelings or who paid the card.

02 Determine precise contributions. Use `calculate_individual_obligations` to know exactly how much each partner owes for groceries or utilities.

03 Manage cash flow instantly. Running `calculate_reconciliation_transfer` tells you the exact transfer needed, so you can settle up in one transaction.

04 Check financial stability easily. `calculate_partner_surplus` shows if your shared finances are actually healthy after all bills are covered.

05 `calculate_income_proportions` quickly gives you the percentage weight of each person's income, setting a fair baseline for all shared decisions.

Real-World Applications

The initial budget split

You and your partner just moved in together. You ask your agent to run the incomes through `calculate_income_proportions` so you know what percentage of shared costs each of you should generally cover.

Reconciling a large upfront payment

One partner paid $3,000 for annual insurance while the bills totaled $2,500. You run `calculate_reconciliation_transfer` to get the precise amount the other person needs to send over immediately.

Figuring out bill responsibility

You just got a list of shared costs—utilities, food, etc. Your agent uses `calculate_individual_obligations` to tell you exactly how much money each person needs to pull out for the month.

Reviewing joint financial health

After paying all the bills and covering personal savings goals, your agent runs `calculate_partner_surplus` so you can see if the shared account has enough left over to cover an emergency fund.

Patterns to Avoid

Assuming a 50/50 split

✗ AVOID

Telling your agent, 'Just assume we both pay half,' when dealing with expenses. This ignores the reality of different incomes and isn't mathematically accurate for proportional sharing.

✓ INSTEAD

Instead, use `calculate_income_proportions` first to set a fair weighted baseline based on actual income disparity. Then, use that proportion to calculate obligations using `calculate_individual_obligations`.

Ignoring overpayments

✗ AVOID

If one person pays $150 more than their share for utilities, simply asking the agent to 'split it' fails. You need a specific transfer calculation.

✓ INSTEAD

Use `calculate_reconciliation_transfer`. This tool handles the complexity of figuring out how much money needs to move between accounts to correct an overpayment.

Mixing up obligations and surplus

✗ AVOID

Using one function to determine what's owed for bills and another function to check remaining funds is inefficient. You need a clear workflow.

✓ INSTEAD

First, run `calculate_individual_obligations` to find the debt. Then, use that resulting amount as an input into `calculate_partner_surplus` to see if there's enough money left over afterward.

The Right Fit

Use this MCP if your household budgeting relies on proportional fairness based on income—that means simple 50/50 splits won't cut it. If you need to know the exact dollar amount each person owes for a shared bill, or how much money needs to move between two accounts to balance things out, this is your tool. Don't use it if you are only tracking spending categories; that's better handled by a simple expense tracker MCP. Also, don't use it just because you need an overall financial status report—for that, look for an overarching budgeting system. This engine focuses purely on the math of *shared cost distribution* and *reconciliation transfers*. You must have income data to start.

Couples Shared Account Engine: Solving Proportional Expense Sharing in Budgeting

Right now, settling up shared bills is a mess of spreadsheets and conflicting memories. You find out the total utilities bill, then you try to figure out how to split it fairly based on who earns what, all while trying to remember who paid the initial down payment last month. It’s tedious, manual math that invites arguments.

With this MCP, your agent handles the whole process. You feed in the income data and the shared costs. The tool calculates the fair proportions automatically, telling you not just how much money is owed, but precisely how much one partner needs to transfer to another to make everything balance.

Couples Shared Account Engine: Streamlining Joint Financial Reconciliation

The biggest time sink is reconciliation. You paid the credit card bill, but your partner paid half of the groceries and you need to figure out if you overpaid or underpaid based on income ratios. It's a complex multi-step calculation that usually requires consulting multiple tabs.

This engine automates the settlement process entirely. After calculating individual obligations for shared costs, it runs the final reconciliation transfer number, giving you one single, actionable dollar amount to settle up and move forward.

Couples Shared Account Engine: 4 Tools for Budgeting and Joint Expenses

Use these tools to automatically determine income proportions, individual spending obligations, required reconciliation transfers, and remaining surplus funds.

#	TOOL	DESCRIPTION
01	`calculate_income_proportions`	Determines the proportional weight of each partner's income compared to the total household income.
02	`calculate_individual_obligations`	Calculates the exact dollar amount each partner is responsible for paying towards shared household expenses.
03	`calculate_partner_surplus`	Evaluates the remaining funds available to each partner after all joint costs and personal expenditures are deducted.
04	`calculate_reconciliation_transfer`	Provides a clear transfer amount needed between partners to balance out discrepancies in how shared bills were paid.

See It in Action

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

U We have $4,000 in total joint bills this month. Partner A earns $80k and Partner B earns $60k. How much does each owe?

📊 **Shared Expense Obligations Report**

Based on your income proportions (A: 66.7%, B: 33.3%), here are the individual obligations for the $4,000 in shared bills:

- **Partner A: $2,666.67**
- **Partner B: $1,333.33**

Please remember to check your total surplus after these payments are made.

U My partner paid $800 for rent and I only owed them $600 based on our proportions. What needs to be transferred?

💰 **Reconciliation Transfer Needed**

The initial payment was: $800.00
Your proportional share should have been: $600.00

To settle the account and balance out, **your partner owes you exactly $200.00.** This amount transfers from your current account to theirs.

U What is our financial standing if we have $15k left after all bills? Assuming A earns $70k and B earns $40k.

✅ **Financial Surplus Check**

After accounting for all shared expenses and personal savings goals, here's the remaining surplus:

- Partner A Surplus: **$5,250.00** (Remaining funds)
- Partner B Surplus: **$4,750.00** (Remaining funds)

The joint account remains stable and healthy.

Frequently Asked Questions

01 **How does the Couples Shared Account Engine handle proportional splits when incomes are vastly different?**

It uses your combined income to determine a mathematical proportion for each partner. For example, if one person earns much more, they contribute a higher percentage of shared costs automatically.

02 **I need help figuring out who owes what after paying rent upfront; can the Couples Shared Account Engine do that?**

Yes, it calculates exactly how much money needs to be transferred between partners. You just input the total bill and who paid it, and it figures out the precise settlement amount.

03 **If we use the engine for shared costs, does it also tell me if our overall finances are stable?**

It checks your financial stability using a surplus calculation. It shows you what remaining funds each person has left after all obligations and personal spending are accounted for.

04 **Can the Couples Shared Account Engine calculate my contribution share if we aren't paying 50/50?**

Absolutely. The engine is designed specifically to move beyond simple splits, calculating contributions based on income ratios so it's always fair and accurate.

05 **What kind of data do I need for the Couples Shared Account Engine to work correctly?**

You primarily need three things: the total incomes of all partners involved, a list of shared expenses, and details on who paid what. The more accurate your inputs, the better the resulting split will be.

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 `"couples-shared-account-engine": { "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

Couples Shared Account Engine 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 Couples Shared Account Engine. 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	Couples Shared Account Engine MCP
Server ID	`019f20c6-85c3-727b-a6c1-63b0765b8050`
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/couples-shared-account-engine.