

M C P S E R V E R

N O C O D E

C L O U D H O S T E D

Household Bill Splitter MCP

Keep your shared household expenses organized and fair.

Household Bill Splitter MCP handles the math and tracking for shared living expenses. Your AI client uses this to identify residents, calculate individual shares for specific bills, log payments, and monitor who owes what. It removes the friction of manual expense tracking by letting your agent manage the ledger directly.

A+ Quality Score 100/100

bills

expenses

residents

splitting

accounting



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

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.

Household Bill Splitter MCP

4 tools available

Cloud-hosted on Vinkius

Managing a shared household shouldn't feel like a part-time accounting job. This MCP gives your AI client the ability to act as a household bookkeeper. Instead of manually calculating splits or chasing roommates for money, you just tell your agent what was spent.

You can use this to pull a list of everyone living in the house, figure out exactly what each person owes for a new utility bill, and record payments as they happen. The MCP keeps a running tally of everyone's financial standing, so you always know if someone is in the red or has a credit. It's a direct way to turn your AI agent into a tool that manages the actual math of shared living, ensuring everyone pays their fair share without the constant back-and-forth.

Core Capabilities

01 — Resident Management

Your agent identifies who is part of the household to ensure accurate splits.

02 — Automated Math

The MCP calculates equal shares for any bill amount you provide.

03 — Debt Tracking

Your agent monitors resident balances to see who owes money or has credits.

04 — Expense Logging

The tool records bill payments and updates the household ledger immediately.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/en/ai-agent-connect/household-bill-splitter — connect your AI agent in three steps.

- 01 Connect your MCP-compatible client to Vinkius.
- 02 Identify your household and residents using the provided tools.
- 03 Input bill amounts to calculate equal splits.
- 04 Record payments to update the ledger and resident balances.

Get up and running by connecting your client to the Vinkius hosted MCP.

Built For

This is for anyone managing shared living spaces who wants to automate the tedious math of splitting costs.

Roommates

Hand off bill calculations and debt tracking to an AI agent.

Property Managers

Use the tool to track resident payments and outstanding balances.

Heads of Household

Delegate the task of recording expenses and monitoring credits to your AI client.

What Changes When You Connect

- 01 Eliminates manual math errors during bill splitting.
- 02 Maintains a real-time ledger of resident debts and credits.
- 03 Centralizes household expense tracking within your AI chat interface.

Real-World Applications

Monthly Utility Splitting

Tell your agent the total for the water bill, and it will calculate the equal share for every resident.

Rent Tracking

Record monthly rent payments to keep an accurate history of what has been paid.

Debt Monitoring

Check the current financial standing of all residents to see who needs to settle up.

New Resident Onboarding

Use the resident list to ensure all current occupants are accounted for in the next bill cycle.

Patterns to Avoid

Manual Math Errors

❌ AVOID

You try to divide a \$142.50 utility bill by four people using your AI client's general knowledge.

✓ INSTEAD

Let the MCP handle the precision. Use ``calculate_equal_split`` to get the exact amount each person owes.

Incomplete Resident Data

❌ AVOID

You tell your agent to split a bill without checking if a roommate moved out last week.

✓ INSTEAD

Always run ``get_resident_list`` first to ensure you are only charging people currently living in the house.

Ignoring Debt Totals

❌ AVOID

You record a payment but forget to check if that person still owes money from the previous month.

✓ INSTEAD

Check the current financial standing of everyone by using ``get_resident_balances`` before you assume the ledger is clear.

The Right Fit

Connect this MCP if you manage a shared household and want to stop manually calculating debts. Do not connect it if you live alone or prefer using a spreadsheet. The decision comes down to whether you want your AI agent to act as your primary bookkeeper.

Household Bill Splitter 4 Tools

These tools handle the math, resident tracking, and ledger updates for shared living expenses.

#	TOOL	DESCRIPTION
01	get_resident_list	This tool pulls the current list of people registered in your household.
02	calculate_equal_split	Use this to find out the exact amount each resident owes for a specific bill.
03	get_resident_balances	This tool provides a summary of the current financial standing for every person in the house.
04	record_bill_payment	This tool logs a processed bill and assigns the corresponding debt to the residents.

See It in Action

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

U Who are the residents in household ID 123?



The residents in household 123 are Alice and Bob.

U How much should each person pay for a \$150.50 electricity bill in household 123?



Each resident owes \$75.25.

U Record a rent payment of \$2000 for household 123.



The rent payment of \$2000 has been successfully recorded for household 123.

Frequently Asked Questions

01 How do I split a bill with this MCP?

You can ask your AI client to use the `calculate_equal_split` tool after providing the bill amount.

02 Can I see who owes money?

Yes, use the `get_resident_balances` tool to see the financial standing of every resident.

03 Which AI clients can I use?

You can use any MCP-compatible client like Claude, Cursor, or Windsurf.

04 Do I need to host this myself?

No, Vinkius hosts and manages the MCP for you, so it is ready to use immediately.

05 How are debts assigned?

When you use the `record_bill_payment` tool, the debt is automatically assigned to the residents.

06 How are the bills divided?

The total bill amount is divided equally among all registered residents in the household.

07 How can I see who owes money?

You can use the ``get_resident_balances`` tool to see the current financial standing and outstanding debts for every resident.

08 Can I record different types of bills?

Yes, you can use ``record_bill_payment`` to log any expense, such as rent, electricity, or one-time repairs, by providing a description.

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 <code>"household-bill-splitter": { "url": "..." }</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

Household Bill Splitter 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 Household Bill Splitter. 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	September 2026
MCP Server	Household Bill Splitter MCP
Server ID	01a0cf4b-2dc1-7176-89a5-2242d7c75fdf
Platform	Vinkius Cloud for AI Agents
Endpoint	<code>https://edge.vinkius.com/{token}/mcp</code>

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/en/ai-agent-connect/household-bill-splitter.