

MCP SERVER

NO CODE

CLOUD HOSTED

Overlapping Utility Splitter MCP for AI Agents

Fairly divide utility bills and roommate expenses when tenants move out mid-cycle.

Overlapping Utility Splitter helps you divide utility bills fairly when a roommate or tenant moves out halfway through a billing cycle. It handles the math for you, calculating exactly how much each person owes based on the days they actually lived in the property. It's perfect for avoiding awkward conversations about shared expenses and ensuring everyone pays their fair share.

A+ Quality Score 100/100

billing

utility-split

expense-sharing

calculator

finance-tools



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.

Overlapping Utility Splitter MCP

3 tools available

Cloud-hosted on Vinkius

Dealing with shared bills gets messy the moment someone moves out mid-month. You're left trying to do manual math to figure out how many days the old roommate stayed versus how many days the new one was there, all while trying to keep the peace. This MCP takes that headache off your plate. Instead of squinting at a paper bill and using a calculator, you can just tell your agent the total cost and the move-out dates. It handles the daily rate conversions and the final split automatically. It's a handy tool for property managers or roommates who want a fair, data-backed way to settle up. You can find this and thousands of other tools in the Vinkius catalog to keep your property operations organized without the manual grunt work.

Core Capabilities

01 — Calculate daily utility rates

Find out the daily cost of a utility based on a total bill and the length of the billing cycle.

02 — Split bills between residents

Divide a total utility cost between two people based on the specific dates they occupied the home.

03 — Verify residency dates

Check if the move-out dates provided are logically sound and fall within the current billing window.

04 — Determine departing resident shares

Get the exact dollar amount an outgoing tenant owes for their portion of the cycle.

05 — Automate pro-rated math

Convert total monthly costs into daily rates to ensure accurate expense sharing.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/mcp/overlapping-utility-splitter — connect your AI agent in three steps.

- 01 Provide the total bill amount and the duration of the billing cycle.
- 02 Input the specific dates each resident lived in the property.
- 03 Get a breakdown of the exact costs for each person.

The bottom line is you get a precise, defensible breakdown of shared costs in seconds.

Built For

This is for landlords, property managers, and roommates who are tired of the who owes what friction. It's for anyone who needs to settle shared expenses accurately without doing the manual math themselves.

Property Manager

Handles tenant move-outs and ensures final bills are split correctly between outgoing and incoming residents.

Roommate

Settles monthly utilities fairly when someone moves out mid-cycle to avoid awkward fairness arguments.

Real Estate Coordinator

Manages shared living expenses for multi-unit properties and needs consistent billing logic.

What Changes When You Connect

- 01 Stop doing manual math for split bills by using `calculate_split_amounts` to get instant totals for everyone.
- 02 Avoid disputes with roommates because `get_daily_cost_rate` provides a clear, data-backed daily price for the home.

-
- 03 Prevent input errors with `validate_residency_period` which catches dates that fall outside the billing cycle before you commit.

 - 04 Save time on tenant move-outs by letting your agent handle the residency math automatically every time a lease ends.

 - 05 Keep property management records accurate by using consistent logic for every shared expense across your portfolio.
-

Real-World Applications

Mid-month roommate move-out

A roommate moves out on day 14 of a 30-day cycle. Ask your agent to split a \$200 electric bill using `calculate_split_amounts`.

Landlord pro-rating water bills

A landlord needs to settle a water bill for a tenant who left mid-month. Use the tool to get the exact daily rate first.

Validating move-out dates

A property manager wants to verify if a move-out date is valid before finalizing a lease end using `validate_residency_period`.

Shared internet split

You're splitting a shared internet bill where the new person joined 10 days into the month and need a fair breakdown.

Patterns to Avoid

Guessing the split

✗ AVOID

Just split it 50/50 since they were here most of the time.

✓ INSTEAD

Use `calculate_split_amounts` to get the exact dollar amount based on actual days lived in the property.

Entering dates outside the cycle

✗ AVOID

Trying to enter a move-out date for day 35 of a 30-day bill.

✓ INSTEAD

Use `validate_residency_period` to ensure the dates are logically sound before calculating.

Forgetting to check the daily rate

X AVOID

Dividing the bill by the number of people instead of the number of days.

✓ INSTEAD

Use `get_daily_cost_rate` first to establish the daily cost before doing any splitting.

The Right Fit

Use this MCP if you need to calculate the exact split of a utility bill when occupancy changes during a single billing period. It's perfect for roommates, landlords, and property managers who want to remove the emotion and manual math from expense sharing. Don't use this if you are trying to split a bill among three or more people, as it is specifically designed for two residents. It's also not for tracking ongoing monthly payments; it's for solving the specific problem of overlapping residency in a single cycle. If you need a full accounting suite for multiple properties, you'll want a more robust property management tool instead.

Overlapping Utility Splitter for Fair Roommate Bill Splitting

Most people handle roommate move-outs by grabbing a napkin and trying to do rough math. You end up guessing how many days the old roommate stayed, trying to figure out the daily rate of the water bill, and hoping the other person doesn't get annoyed by the final number. It's a headache that leads to unnecessary friction.

This MCP changes that by automating the entire calculation. You just tell your agent the total bill and the dates people lived there. It pulls the daily rate, validates the dates, and gives you a final number you can actually show to your roommate. You get a clean, defensible split without the manual work.

Overlapping Utility Splitter for Property Management Billing

Property managers often have to deal with pro-rated utilities when a tenant leaves mid-lease. This usually involves opening a spreadsheet, typing in dates, and manually calculating the daily cost to ensure the outgoing tenant pays their fair share.

With this MCP, that process disappears. Your agent handles the residency validation and the cost distribution in one go. It turns a tedious administrative task into a simple conversation with your AI client.

Overlapping Utility Splitter 3-Tool Roommate Billing

Use these tools to calculate daily rates, split utility costs between residents, and validate move-out dates automatically.

#	TOOL	DESCRIPTION
01	<code>get_daily_cost_rate</code>	Finds the daily price of a utility based on the total bill and cycle length. This helps establish a baseline for daily usage.
02	<code>calculate_split_amounts</code>	Divides a total bill between two people based on their specific days in the home. It creates a clear breakdown of who pays what.
03	<code>validate_residency_period</code>	Checks if the move-out dates provided actually make sense within the billing window. This prevents errors in the final calculation.

See It in Action

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

U The electric bill is \$150 for a 30-day cycle. Person A moved out on day 12. How much should they pay?



Person A owes \$60.00, and Person B owes \$90.00.

U What is the daily rate for a \$300 water bill over 31 days?



\$9.68 per day.

U Is it valid to say someone moved out on day 35 of a 30-day cycle?



No, that is not a valid residency period as the move-out day exceeds the billing cycle duration.

Frequently Asked Questions

01 How does the Overlapping Utility Splitter help with roommate disputes?

It provides a mathematically accurate way to split bills based on how many days each person actually lived in the home. This removes guesswork and helps keep things fair.

02 Can I use Overlapping Utility Splitter for property management?

Yes, it's great for landlords who need to pro-rate utility costs for tenants moving out mid-cycle. It ensures the outgoing tenant pays for exactly what they used.

03 Does the Overlapping Utility Splitter handle more than two people?

This specific MCP is designed to split costs between two residents. For larger groups, you'd need to calculate the daily rate first and then divide that among the participants.

04 How accurate is the split calculation?

It uses the exact daily cost rate of the utility. By converting the total bill into a daily amount, it ensures the split is based on the actual duration of stay.

05 What happens if I enter a move-out date that is wrong?

The MCP includes a check to make sure your dates actually fit inside the billing cycle. If you enter a date that's too late, it will tell you the residency period is invalid.

06 Can I use Overlapping Utility Splitter for other types of bills?

It's built for utilities like water, electric, or gas. It works best when there's a clear daily cost associated with the total bill amount.

07 How does the split calculation work?

It uses ``calculate_split_amounts`` to find a daily rate and allocates costs based on how many days each person resided in the property.

08 Can I verify if my move-out date is valid?

Yes, use the ``validate_residency_period`` tool to check if the departure day falls within the billing cycle.

09 What inputs are needed for the split?

You need the total bill amount, the day person A moved out, and the total days in the billing cycle.

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>"overlapping-utility-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

Overlapping Utility 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 Overlapping Utility 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	July 2026
MCP Server	Overlapping Utility Splitter MCP
Server ID	019f384c-adb2-73f0-9c70-6b50c0b01c9e
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/mcp/overlapping-utility-splitter.