

MCP SERVER

NO CODE

CLOUD HOSTED

Chore Fairness Balancer for equitable distribution.

Stop burnout by balancing workloads against historical contributions.

Chore Fairness Balancer is a distribution engine that prevents burnout by balancing current workloads against historical contributions. It uses effort points to weight tasks and respects member availability and exclusions. Your AI client can use this to generate optimal schedules, check for logical conflicts in your setup, and predict how changes to chores or availability will impact the overall fairness of the group.

A+ Quality Score 100/100

chore

fairness

scheduling

workload

automation



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.

Chore Fairness Balancer MCP

4 tools available

Cloud-hosted on Vinkius

Managing household or group chores often leads to resentment when one person ends up doing more than their fair share. This MCP solves that by treating labor as a measurable metric. Instead of just listing tasks, it uses effort points to weight each chore based on how much work it actually takes.

When you connect this to your AI client, you can stop guessing who should do what. The engine looks at how much work everyone has done in the past to ensure the current week's load is balanced. It also respects specific constraints, like when someone is unavailable or if certain members are excluded from specific tasks. You can run simulations to see if adding a new recurring chore will tip the scales too far for one person, or use it to verify that your chore configuration doesn't have any logical gaps that would leave a task unassigned. It turns chore scheduling from a source of conflict into a data-driven process.

Core Capabilities

01 — Effort-based weighting

Your agent assigns points to tasks to reflect actual labor required.

02 — Historical balancing

The MCP adjusts current assignments based on past contributions to prevent burnout.

03 — Constraint validation

Your AI client checks that all chores can be covered by available members.

04 — Fairness forecasting

The tool predicts how schedule changes will affect the group's workload balance.

05 — Workload snapshots

Your agent can pull summaries of how much work each person is carrying.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/en/ai-agent-connect/chore-fairness-balancer — connect your AI agent in three steps.

- 01
- Connect the MCP to Claude, Cursor, or Windsurf via Vinkius.
- 02
- Provide your list of chores and effort points to your agent.
- 03
- Define member availability and any specific exclusions.
- 04
- Ask your agent to calculate the optimal assignments.
- 05
- Review the workload summary or fairness impact to confirm the plan.

Connect the MCP to your preferred client and let your agent handle the math.

Built For

This MCP is for anyone managing shared living spaces or group responsibilities where workload equity is a priority.

Household Managers

People looking to automate fair chore rotation for families or roommates.

Community Organizers

Individuals managing shared duties in dorms, co-living spaces, or small organizations.

Project Coordinators

Users who want to apply workload balancing logic to small team tasks.

What Changes When You Connect

- 01
- Reduces burnout by weighting tasks with effort points.
- 02
- Prevents resentment by accounting for historical contributions.
- 03
- Eliminates scheduling conflicts by validating member availability.

04 Provides data-driven insights into group labor distribution.

Real-World Applications

Roommate Chore Rotation

Automatically assign cleaning and maintenance tasks so no one roommate carries the load.

Family Task Management

Balance household chores among family members based on their specific availability.

Co-living Workload Audits

Use workload summaries to see if the current chore distribution is actually fair.

New Task Planning

Simulate the impact of adding a new weekly chore before you officially assign it.

Patterns to Avoid

Ignoring historical labor data

❌ AVOID

You assign chores based only on who is free this week, ignoring that one person did everything last month.

✓ INSTEAD

Use the MCP to account for past contributions so the workload stays even over time.

Adding tasks blindly

❌ AVOID

You add a new weekly deep clean to the list without checking if anyone actually has the time to do it.

✓ INSTEAD

Run a simulation first using ``simulate_fairness_impact`` to see how that new chore shifts the balance.

Assuming the schedule is valid

❌ AVOID

You set up a complex rotation and realize halfway through the week that two people are assigned the same task.

✓ INSTEAD

Run ``validate_chore_config`` to catch logical gaps and assignment conflicts before they happen.

The Right Fit

Connect this MCP if you manage a household, dorm, or small group where workload equity is a priority. Do not use it if you are managing a solo project or a large corporation. The decision comes down to whether you have multiple people sharing a pool of responsibilities.

Chore Fairness Balancer Toolset

These tools handle everything from generating balanced schedules to simulating how new tasks change your group's workload.

#	TOOL	DESCRIPTION
01	calculate_assignments	This tool generates the most balanced chore distribution for a specific timeframe.
02	get_member_workload_summary	Get a snapshot of how labor is currently distributed across your entire group.
03	simulate_fairness_impact	Use this to predict how adding new chores or changing member availability will change the fairness score.
04	validate_chore_config	This tool checks your chore list and member constraints to ensure there are no logical conflicts.

See It in Action

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

U Assign the weekly chores for our household.



The chores have been assigned: Alice is responsible for Cleaning (5 pts) and Maintenance (3 pts), while Bob is responsible for Logistics (4 pts). The current fairness score is 0.95.

U Check if our current chore configuration is valid.



The configuration is valid. All chores have at least one eligible member based on current availability and exclusions.

U How much work is Charlie doing this week?



Charlie has 8 effort points assigned this period, with a historical effort total of 45 points and a burden ratio of 0.18.

Frequently Asked Questions

01 How does the MCP decide who does which chore?

It uses effort points to weight the difficulty of tasks and balances these against the historical contributions and current availability of each member.

02 Can I use this with Claude or Cursor?

Yes, you can connect this MCP to any MCP-compatible client including Claude, Cursor, and Windsurf.

03 What happens if someone is unavailable for a chore?

The MCP respects member availability and exclusions when calculating assignments to ensure tasks are only given to eligible people.

04 How do I know if the chore list is missing something?

You can use the `validate_chore_config` tool to check if your chores and member constraints are logically sound.

05 Does this MCP host the data?

Vinkius hosts and manages the MCP, so it is ready to use immediately after you connect your client.

06 How does the system ensure fairness?

The system uses effort points for each task and compares current assignments against the historical effort totals of each member to maintain a stable fairness score.

07 Can I prevent certain people from doing specific tasks?

Yes, you can define exclusions in the member list to ensure specific members are not assigned tasks they are unable or unwilling to perform.

08 What happens if a chore cannot be assigned?

If a chore cannot be assigned due to availability or exclusion constraints, it will be listed in the missed constraints section of the assignment result.

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>"chore-fairness-balancer": { "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

Chore Fairness Balancer 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 Chore Fairness Balancer. 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	October 2026
MCP Server	Chore Fairness Balancer MCP
Server ID	01a0f748-5856-70ca-b552-7c6297e68625
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/chore-fairness-balancer.