MCP SERVER | NO CODE | CLOUD HOSTED

Fair Chore Rotation Algorithm MCP for AI Agents

Creating perfectly balanced group task schedules and managing household responsibilities

The Fair Chore Rotation Algorithm is an intelligent scheduling system designed for groups who need to distribute tasks evenly and fairly. It automatically manages complex rotations, ensuring that no participant gets stuck doing the same chore back-to-back or consistently receiving the heaviest load. This MCP uses deterministic algorithms to generate perfectly balanced schedules, making household management simple and equitable.

A+ Quality Score 100/100

rotation | scheduling | fairness | algorithm | tasks

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JULY 2026
vinkius.com/mcp/fair-chore-rotation-algorithm

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.

Fair Chore Rotation Algorithm MCP

3 tools available
Cloud-hosted on Vinkius

Managing group responsibilities can feel like a spreadsheet nightmare. You're constantly worried about fairness—did Sarah do Laundry four times while Tom only did Dishes? The Fair Chore Rotation Algorithm fixes that by creating structured, cyclic schedules for any set of participants and tasks. It uses advanced scheduling logic to build out perfect grids, guaranteeing balance over time.

Instead of manually tracking assignments week after week, you simply connect your AI client through Vinkius. Your agent then handles the complexity. You can ask it to generate a full schedule using the `generate_rotation_grid` tool, check if the current plan violates rules with `validate_rotation_integrity`, or measure exactly how often each person is assigned tasks using `calculate_task_frequency`. This system ensures everyone shares the load equally, keeping arguments out of your routine and making household management predictable.

Core Capabilities

01 — Generate a full chore schedule

The MCP creates a complete day-by-day assignment plan for all participants based on provided tasks.

02 — Check for scheduling rules violations

It verifies that no person is scheduled to perform the same task two days in a row, preventing repetitive assignments.

03 — Measure assignment equality across the group

You can calculate and compare how many times each specific chore was assigned among all participants.

04 — Adjust schedules based on participant count

The system adapts its rotation logic when you change the number of people or tasks involved in the group.

05 — Maintain long-term rotational balance

It builds cycles that ensure every person eventually gets a chance at every task in an equal sequence.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/fair-chore-rotation-algorithm** — connect your AI agent in three steps.

01 First, you tell your AI client the roster of participants and all the available chores. This sets the parameters for the rotation.

02 Next, the MCP runs the scheduling logic to build the grid, ensuring that assignments are balanced over time and no person repeats a task immediately.

03 Finally, you receive the complete, validated schedule, allowing you to distribute responsibilities without worrying about fairness or repetition.

The bottom line is, it turns complex chore management into a simple input/output process, giving you an instant, equitable plan.

Built For

This MCP serves household managers and small team leads who struggle with keeping track of shared responsibilities. If your group uses spreadsheets or physical charts that quickly become outdated, this is for you. It solves the constant friction point of figuring out 'whose turn was it?'

Household Coordinator

Uses this MCP to generate monthly chore rotations and verify they are perfectly fair across all family members.

Small Team Lead (Non-HR)

Manages shared, non-critical team tasks (like equipment setup or meeting prep) and needs a way to ensure equitable distribution of effort among volunteers.

What Changes When You Connect

01 Guaranteed Fairness: Never worry about one person doing all the heavy lifting again. Use `calculate_task_frequency` to prove that assignments are equally distributed.

02 Zero Repetition Risk: The MCP validates every assignment, so you know immediately when a chore is correctly rotated away from a participant using `validate_rotation_integrity`.

03 Effortless Planning: Stop drawing grids by hand. Use the `generate_rotation_grid` tool to create months of balanced chores in seconds.

04 Predictable Routine: By generating a deterministic schedule, you take guesswork out of household life and build consistency into your routine.

05 Audit Trail: You can always audit past schedules, checking task frequency to ensure accountability for every chore assigned.

Real-World Applications

The family needs a new rotation cycle

A parent asks their agent to generate a six-week schedule for four kids and five chores. The agent uses `generate_rotation_grid` and delivers a complete, balanced plan that keeps everyone busy without repeating tasks.

Checking for fairness after an ad-hoc week

The household coordinator asks the agent to check task frequency across the last month's assignments. The agent uses `calculate_task_frequency` and immediately identifies that one person was assigned dishes three times more than anyone else.

Validating a partial schedule draft

A user manually enters a few days of chores into a temporary list. They ask the agent to validate it, using `validate_rotation_integrity`, which alerts them that one person is assigned laundry on consecutive days.

Patterns to Avoid

Using simple round-robin lists

✗ AVOID

Creating a schedule where you just list tasks: Day 1: A, B, C. Day 2: C, A, B. This often fails to account for task type or individual capacity.

✓ INSTEAD

Use this MCP's tools. First, run `generate_rotation_grid` to get a balanced schedule, then use `calculate_task_frequency` to confirm the balance is right.

Manually tracking every single chore

✗ AVOID

Maintaining physical whiteboards or complex spreadsheets that require constant manual updates and risk human error.

✓ INSTEAD

Let your agent handle it. Use `validate_rotation_integrity` to instantly check any existing schedule for rule violations, saving you hours of cross-checking.

Ignoring task weight or difficulty

✗ AVOID

Assuming all chores are equally difficult and distributing them randomly, leading to burnout among some members.

✓ INSTEAD

While the MCP ensures frequency balance, always supplement the schedule by adjusting task weights before running `generate_rotation_grid`.

The Right Fit

Use this if your primary pain point is achieving true equity and avoiding repetitive assignments in a group setting. If you need to guarantee that everyone shares the burden evenly over time, this MCP works perfectly. You rely on the system's ability to calculate task frequency and build complex, non-repeating cycles.

Don't use it if your needs are simple scheduling (e.g., 'Who goes to the store tomorrow?'). For single-event planning, a basic calendar tool is fine. This MCP is for algorithmic group management. If you only need to track who did what and never care about fairness or repetition, then this MCP might be overkill.

Fair Chore Rotation Algorithm: Managing Family Chores Beyond Spreadsheets

Right now, chore assignments usually involve a messy mix of sticky notes, family calendars, and complicated spreadsheets. You end up spending more time arguing about fairness than actually doing chores. You copy-paste lists of names, you cross-reference tasks, and eventually, the whole thing becomes an outdated, unreliable document.

With this MCP, you tell your agent who is involved and what needs to get done. It handles all that complicated math instantly. What you get back isn't just a list; it’s a perfectly balanced, rotation-proof schedule that keeps everyone accountable.

Fair Chore Rotation Algorithm: Verifying Group Task Balance and Equality

The biggest issue is keeping track of who did what. You have to manually count every instance of 'Dishes' or 'Vacuuming' across weeks, hoping you don't miss a day. It’s tedious, time-consuming manual accounting that usually leads to arguments.

Now, your agent uses the system's tools to check this instantly. You get a clear report showing task frequency for every single chore and person involved. The argument is over.

Fair Chore Rotation Algorithm: 3 Tools for Equitable Scheduling

Use these tools to generate, validate, and audit any chore schedule to guarantee maximum fairness and balance among all participants.

#	TOOL	DESCRIPTION
01	`calculate_task_frequency`	This tool measures exactly how many times each specific chore was assigned within the entire group schedule.
02	`generate_rotation_grid`	It builds a full, day-by-day assignment grid showing who does what over a specified period.
03	`validate_rotation_integrity`	You can check the schedule to confirm that no person is assigned the same chore twice in a row.

See It in Action

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

U Generate a 4-week rotation schedule for Mom, Dad, and two kids with Laundry, Dishes, Vacuuming, and Trash.

Generated Rotation Grid (Weeks 1–4)

DAY	PARTICIPANT	CHORE
Wk 1, Mon	Mom	Laundry
Wk 1, Tue	Dad	Dishes
Wk 1, Wed	Kid A	Vacuuming
...	...	...
Wk 4, Sun	Kid B	Trash

Verification: No participant repeats a chore on consecutive days.

U Does the schedule I drafted work? (List provided)

🚨 Integrity Violation Detected!

The current draft is invalid for two reasons:

1. **Violation:** Sarah was assigned Vacuuming on both Day 3 and Day 4.
2. **Balance Issue:** Over the last 7 days, Dishes were assigned 5 times, while Laundry was only assigned 1 time. The load isn't balanced.

U How many times did 'Dishes' get assigned this month?

Task Frequency Report:

CHORE	TOTAL ASSIGNMENTS
Laundry	12
Dishes	**18**
Vacuuming	9
Trash	6

Conclusion: Dishes were assigned significantly more often than any other chore.

Frequently Asked Questions

01 **How does the rotation algorithm prevent consecutive tasks?**

The system uses a daily offset that increments each day, shifting the task index for every participant to ensure no one repeats a chore two days in a row.

02 **Can I verify if my generated schedule is valid?**

Yes, you can use the `validate_rotation_integrity` tool to inspect your schedule for any consecutive assignment violations.

03 **How do I check if tasks are distributed equally?**

Use the `calculate_task_frequency` tool to generate a frequency map of all tasks in your schedule, allowing you to measure distribution equality.

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 `"fair-chore-rotation-algorithm": { "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

Fair Chore Rotation Algorithm 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 Fair Chore Rotation Algorithm. 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	Fair Chore Rotation Algorithm MCP
Server ID	`019f20c7-a3d9-71a6-98e9-13b12c3f8db4`
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/fair-chore-rotation-algorithm.