MCP SERVER | NO CODE | CLOUD HOSTED

Fasting Window Resolver MCP for AI Agents

Pinpoint Intermittent Fasting Schedules and Meal Timing

The Fasting Window Resolver MCP handles intermittent fasting schedules with precision. It lets your AI client instantly tell you if you're currently eating or fasting, map out your entire daily schedule, and calculate the exact time remaining until your next meal boundary.

A+ Quality Score 100/100

fasting | intermittent-fasting | health-tracker | scheduling | wellness

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JULY 2026
vinkius.com/mcp/fasting-window-resolver

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.

Fasting Window Resolver MCP

3 tools available
Cloud-hosted on Vinkius

Managing an intermittent fast requires more than just counting hours; it demands knowing exactly where you are in relation to your scheduled windows. This MCP provides precise tools for tracking complex feeding and fasting cycles.

It lets you ask simple questions like, 'Am I safe to eat right now?' or 'When does my fast end?' The system checks your current time against your personalized schedule, giving you a definitive answer instantly. You can also get the full rundown of all four boundary timestamps for any given day.

If you use this MCP through Vinkius, your AI client connects to thousands of other services in one place. It means you don't have to jump between multiple health trackers or spreadsheets just to manage your diet schedule. You keep everything centralized and get reliable timing data right where you need it.

Core Capabilities

01 — Determine current fasting/feeding status

Instantly checks if the user is currently within their scheduled eating window or deep in a fast.

02 — View complete daily schedule boundaries

Retrieves all four critical timestamps for the day, including transitions that cross midnight.

03 — Calculate time until next meal event

Provides a precise countdown of how much time is left before any specified target meal or window begins.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/fasting-window-resolver** — connect your AI agent in three steps.

01 You first define your fasting parameters, such as the start and length of your feeding window.

02 Your AI client runs a query against this MCP to analyze your current time relative to those boundaries.

03 It returns an actionable status—for example, 'You are currently in a fast' or 'Dinner starts in 45 minutes.'

The bottom line is that you stop guessing about your schedule and start acting based on real-time data.

Built For

Biohackers, health coaches, and wellness enthusiasts who rely on precise timing for their daily routines. If your diet or fitness plan hinges on hitting specific meal windows, this MCP is essential.

Nutritionist/Health Coach

Uses it to advise clients on when they are currently in a safe window to eat, ensuring adherence to complex intermittent fasting protocols.

Biohacker/Athlete

Tracks precise time remaining until their next nutrient intake or meal to optimize energy levels and muscle recovery.

Wellness Enthusiast

Keeps a clear, reliable record of their fasting cycles, especially when schedules cross time zones or midnight boundaries.

What Changes When You Connect

01 Stop guessing about your diet. Using `get_current_state_details` immediately tells you if you're in a feeding or fasting window, so you never risk breaking your fast accidentally.

02 Manage complex schedules that cross midnight. The tool uses `get_window_boundaries` to map out every boundary for the day, giving you total clarity on your eating limits.

03 Optimize meal timing with precision. `calculate_duration_until_event` gives you a clear countdown, so you know exactly how much time is left until that pre-workout shake or evening snack.

04 Saves you from manual calculation. Instead of using multiple spreadsheets to track boundaries and transitions, your agent handles it all in one step.

05 Reduces cognitive load. You get definitive answers on status and timing instantly, freeing up mental energy for actually following your routine.

Real-World Applications

Trying to follow a complex 16/8 schedule

A user asks their agent, 'Am I good to eat now?' The agent calls `get_current_state_details` and replies: 'You are currently in your feeding window. You've got about three hours left.' This eliminates the guesswork when tracking complex protocols.

Planning for a travel day with time zone shifts

A coach asks their agent to map out boundaries for an 8-hour eating window starting at noon local time. The tool uses `get_window_boundaries` and correctly shows the full schedule, even accounting for midnight transitions in different time zones.

Needing a quick reminder before a workout

A user needs to know when their pre-workout meal window closes. They ask how long they have until 4:00 PM, and the agent uses `calculate_duration_until_event` to give them an accurate countdown.

Checking a full week's schedule at once

The user asks for all scheduled boundaries. The MCP runs through multiple days using `get_window_boundaries`, providing the necessary data points for meal prep and planning.

Patterns to Avoid

Treating it like a simple timer

✗ AVOID

Just setting an alarm for '8 hours'. This fails because it doesn't account for the specific start time, or transitions across midnight.

✓ INSTEAD

You must use `get_window_boundaries` first. This tool maps out all four precise boundary timestamps needed to build a correct schedule, regardless of where the day starts.

Calculating duration manually

✗ AVOID

Using a simple clock calculation like '12:00 PM to 4:00 PM is 4 hours.' This ignores time zones and specific event targets.

✓ INSTEAD

Use `calculate_duration_until_event`. It takes the target date/time into account, giving you an accurate duration even if the boundary crosses a day change.

Ignoring current status

✗ AVOID

Assuming that because it's 'after noon,' they can eat. This overlooks whether they are actually in their designated feeding window.

✓ INSTEAD

Always call `get_current_state_details` first. This tool is the quickest way to confirm your precise physiological state relative to your established schedule.

The Right Fit

Use this MCP if you need absolute certainty about your eating and fasting schedules, especially when those cycles cross midnight or involve varying time zones. If you're tracking intermittent fasting for health reasons, this is the definitive tool. Don't use it if you just need a generic timer; that won't work because it lacks boundary awareness. Likewise, don't rely on it if your diet plan doesn't follow rigid, timed windows—if your eating habits are flexible, a simple calendar reminder will suffice. This MCP is for precision biohacking.

Fasting Window Resolver: Precision Intermittent Fasting Scheduling

Today, tracking IF means juggling apps, spreadsheets, and sticky notes. You manually calculate boundaries, checking if 8 hours starting at 4 PM is actually safe when the clock rolls over to a new day. It's tedious; you spend more time figuring out *when* you can eat than actually eating.

With this MCP, your agent handles all that math instantly. Simply ask for your boundaries, and it returns the full schedule, correctly accounting for midnight transitions so you know exactly when your fast starts and ends.

Fasting Window Resolver: Real-Time Meal Timing Clarity

The biggest pain point is the uncertainty. You're in a window, but do you have enough time for your workout and post-workout meal? Manually checking the remaining minutes against a fluctuating clock is frustrating.

Now, you just ask. The MCP uses `calculate_duration_until_event` to give you the definitive countdown. It’s instant confirmation that lets you plan around your boundaries instead of constantly worrying about them.

Fasting Window Resolver: 3 Tools for Meal Timing Clarity

These tools allow your agent to check your current status, view full daily schedules, and count down precisely until any scheduled meal event.

#	TOOL	DESCRIPTION
01	`get_current_state_details`	Checks your status right now and tells you how much time is left until the next change in your fasting cycle.
02	`calculate_duration_until_event`	Determines the exact remaining minutes or hours until any specific target date and time is reached.
03	`get_window_boundaries`	Pulls up the full schedule for a given day, showing all four necessary boundary timestamps.

See It in Action

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

U I'm doing 18/6, starting my fast at 8 PM. What are my windows for tomorrow?

Intermittent Fasting Schedule

- **Fast Starts:** Tonight, 8:00 PM (Ends Tomorrow)
- **Feeding Window Opens:** Tomorrow, 12:00 PM
- **Fast Ends:** Tomorrow, 8:00 AM

This schedule covers all boundaries for the next two days. Keep that fast going!

U It's 9:30 AM. Am I in a safe window to have breakfast?

Current Status Check

You are currently **fasting**. Your fast started at 8 PM last night and doesn't end until 12:00 PM today. You still have 2 hours and 30 minutes left to stay clean.

U How long do I have until my evening meal starts?

Meal Countdown

Your next scheduled event is Dinner.

⏰ **Time Remaining:** 4 hours, 15 minutes.
🗓️ **Target Time:** Today at 6:00 PM.

Frequently Asked Questions

01 **How does the Fasting Window Resolver handle time zones?**

It handles them automatically. You just need to provide your local time zone, and the MCP calculates all boundaries correctly, even when your schedule crosses multiple time zone lines.

02 **Is this enough for tracking general healthy eating habits?**

While it's designed for IF, its core function is precise scheduling. It tells you exactly if and when a window opens or closes, giving you the necessary structure for any timed routine.

03 **What happens if my schedule changes mid-fast?**

You simply update your start/end times with your agent. The Fasting Window Resolver immediately recalculates everything, updating all boundaries and current status without you having to do the math yourself.

04 **Can I use Fasting Window Resolver for meal planning?**

Yes. By mapping out your complete daily schedule using `get_window_boundaries`, you gain a clear visual plan, making it easy to book meals and stick to your boundaries.

05 **Does the Fasting Window Resolver only work for 16/8 schedules?**

No. You can set any window length or duration, from 4 hours up to 24 hours. It adapts to whatever specific protocol you follow.

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 `"fasting-window-resolver": { "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

Fasting Window Resolver 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 Fasting Window Resolver. 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	Fasting Window Resolver MCP
Server ID	`019f2a94-913f-703d-92b6-037113c9c91e`
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/fasting-window-resolver.