MCP SERVER | NO CODE | CLOUD HOSTED

FareHarbor MCP

Handle live booking inventory via chat.

FareHarbor MCP lets you manage and process tour and activity bookings directly from your AI agent's conversation window. Use this to find authorized operators, check real-time availability across specific dates or date ranges, fetch detailed pricing for time slots, and finalize new customer reservations using natural language prompts.

A+ Quality Score 100/100

booking-engine | tour-management | activity-reservations | inventory-availability | online-payments | travel-tech

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/fareharbor

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

FareHarbor MCP

11 tools available
Cloud-hosted on Vinkius

This connector gives your AI client full control over the entire tour booking lifecycle. Forget opening a web browser, navigating multiple forms, or copying IDs into separate spreadsheets. You simply talk to your agent about travel plans, and it handles the rest.

Need to know which companies operate in an area? Ask it. Want to check if there are spots left for whale watching next Tuesday? It gets that live inventory data instantly. When you find a slot, you can confirm a booking—including customer details and ticket types—all within the chat flow. Because we host this on Vinkius, your agent doesn't need to know how FareHarbor works; it just needs to know how to talk to the catalog. It lets you monitor existing reservations by UUID or list all recent bookings for an operator without ever leaving your communication tool.

Core Capabilities

01 — Discovering Operators

List every authorized tour company and activity provider available in the network.

02 — Checking Availability

Fetch real-time inventory for specific tours, checking slots by a single date or across an entire range of dates.

03 — Retrieving Item Details

Get comprehensive information on any specific tour or activity item, including general details and pricing structures.

04 — Managing Reservations

List recent bookings for an operator or pull the detailed status of a reservation using its unique ID (UUID).

05 — Creating Bookings

Finalize new reservations, providing all necessary customer information and ticket types in one transaction.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/fareharbor** — connect your AI agent in three steps.

01 First, subscribe to this MCP and provide your FareHarbor App Key and User Key credentials.

02 Next, connect your AI agent or client (like Cursor or Claude) through the Vinkius platform using those keys.

03 Finally, ask your agent a natural language question—for example, 'What's available for hiking next week?'—and it executes the necessary checks and actions.

The bottom line is you talk to your AI client, and it uses this MCP to run all the complex booking logic in the background.

Built For

This connector is built for travel agencies, Online Travel Agents (OTAs), and operations teams who deal with high volumes of live inventory data. If your job involves checking availability or confirming bookings across multiple external systems, this MCP saves hours of manual switching.

Travel Affiliate Manager

Uses the MCP to quickly find and confirm activities for clients in chat tools without ever leaving the messaging platform.

OTA Developer

Integrates live booking data streams into concierge chatbots or internal ticketing systems using AI-powered workflows.

Operations Coordinator

Monitors and adjusts the status of existing client bookings, checking UUID details or listing all recent reservations when issues pop up.

What Changes When You Connect

01 You don't have to manually check availability. By asking your agent, it uses `list_availabilities_by_date` or `list_availabilities_by_range` to give you instant capacity data for any tour item.

02 Stop switching between tabs when confirming a sale. You can submit a new reservation, including customer details and ticket types, in one go using the `create_booking` tool.

03 Need to check if an operator is authorized? Instead of browsing lists, just ask your agent, and it calls `list_companies` to give you the full roster instantly.

04 When a client asks about a reservation they made last month, your agent can look up that status using `get_booking`, requiring only the UUID for fast confirmation.

05 The MCP gives you access to crucial operational data—like all authorized companies or lodging locations via `list_lodgings`—without needing developer-level API calls.

Real-World Applications

Handling last-minute itinerary changes

A client cancels their original booking. The agent uses `get_booking` to confirm the cancellation details, then immediately checks for alternatives using `list_availabilities_by_range` and presents three suitable options back to the user.

Onboarding a new travel partner

A team member needs to know which companies are available in a region. They prompt the agent, which uses `list_companies`. The team member can then see all authorized operators and proceed with booking items from one of them.

Automating concierge responses

A user asks for tomorrow's activities. The agent first calls `list_items` to find available tours, then uses `get_availability` on the top result to confirm pricing and remaining spots before presenting a booking link.

Auditing past sales

A manager needs to verify if an account has processed bookings for 'Island Adventures.' They prompt the agent, which calls `list_bookings` to pull up recent activity and confirm revenue streams.

Patterns to Avoid

Treating it like a simple lookup tool

✗ AVOID

The user asks only for 'availability' but doesn't specify the date or item. The agent fails because it lacks context.

✓ INSTEAD

Always include enough detail, such as 'Check availability for the Sunset Cruise on May 15th.' This allows the agent to correctly call `list_availabilities_by_date`.

Manually cross-referencing bookings

✗ AVOID

Copying a booking ID from an email, pasting it into a website form, and waiting for confirmation.

✓ INSTEAD

Just give your agent the UUID. It uses `get_booking` to fetch all status details (paid, pending, etc.) instantly.

Assuming booking success

✗ AVOID

Creating a reservation without verifying if spots are actually open. The booking might fail due to capacity limits.

✓ INSTEAD

Always check the slot first. Use `get_availability` before attempting to call `create_booking`.

The Right Fit

Use this MCP if your core job involves managing real-time, transactional travel inventory—meaning you need to know *if* a spot

is open and then *buy* that spot. It's perfect for integrating booking workflows into chat or code environments.

Don't use this if all you need is static reference data (like a general list of companies). For pure reading, you might only need `list_companies` and `list_items`. If your goal is just to read documentation about the system, using an MCP isn't necessary. You should use it when the outcome must be measurable: listing, checking capacity, or finalizing a purchase.

Managing travel inventory shouldn't require constant context switching.

Today, confirming a trip means jumping between multiple systems. You check the booking platform to see if spots are open; then you go to the payment portal to confirm pricing; and finally, you switch to your CRM just to log the reservation details. It's a tedious copy-paste loop that slows down every sale.

With this MCP, you talk to your agent about the trip, and it handles the entire sequence. It checks the inventory using `list_availabilities_by_range`, confirms the price using `get_item_details`, and completes the reservation with a single prompt. You get back confirmed bookings in plain conversation.

FareHarbor MCP: Confirming Bookings, Effortlessly.

The manual steps that vanish are the API lookups and data validation. Instead of needing to manually find a tour's shortname or its UUID before submitting it, your agent handles all mapping automatically when you request a booking via `create_booking`.

Now, managing complex reservations is just conversation. Your AI client does the heavy lifting so you can focus entirely on the customer experience.

FareHarbor: 11 Booking & Availability Tools

These tools let you programmatically check inventory, manage company data, pull booking statuses, and finalize new reservations using the power of your AI agent.

#	TOOL	DESCRIPTION
01	`create_booking`	Submits a new reservation for a specific tour or activity item.
02	`get_availability`	Retrieves detailed status and capacity for a single, specified time slot.
03	`get_booking`	Pulls the full details of an existing booking using its unique identifier (UUID).
04	`get_item_details`	Fetches all general information about a specific tour or activity item.
05	`get_me`	Confirms and retrieves the current API identity and credentials for the connected user/affiliate account.
06	`list_availabilities_by_date`	Lists all available time slots for an item on one particular date.
07	`list_availabilities_by_range`	Lists available slots for an item when given a start and end date range.
08	`list_bookings`	Retrieves a list of recent bookings made for any company or operator.
09	`list_companies`	Returns an authorized list of all tour companies and operators you can book through.
10	`list_items`	Lists every available tour or activity item offered by a specific company.
11	`list_lodgings`	Retrieves potential lodging and pickup location details for an operator's service area.

See It in Action

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

U List the tour companies I can book for.

I've retrieved your authorized operators. You can book for 'Island Adventures' (shortname: island-adv) and 'Mountain Treks' (shortname: mountain-treks). Which company's items would you like to see?

U Check availability for the Whale Watching tour tomorrow.

Searching availability for tomorrow... I found two slots: 9:00 AM (15 spots left) and 2:00 PM (8 spots left). Would you like to proceed with a booking for any of these?

U Show me the details for booking UUID abc-123.

Retrieving booking... UUID abc-123 is a confirmed reservation for 'John Doe' on the Sunset Cruise. Status is PAID and the customer has been sent their confirmation email.

Frequently Asked Questions

01 **How do I find all authorized tour companies using FareHarbor MCP?**

You call `list_companies`. This tool pulls a full list of every operator in your network, allowing you to know exactly who is available for booking.

02 **Can I check availability across multiple dates with the FareHarbor MCP?**

Yes. Use the `list_availabilities_by_range` tool. This lets you see if a tour has open slots throughout an entire week or month, which is key for planning trips.

03 **Which tool do I use to finalize and pay for a booking?**

You must use `create_booking`. Once the availability checks are done, this tool accepts all required customer and ticket details and processes the final reservation.

04 **What if I only have the UUID? How do I check the status using FareHarbor MCP?**

Use the `get_booking` tool. You provide just the unique UUID, and it returns a detailed record of that booking's current status, whether paid or pending.

05 **Can I see what items a company offers before checking availability?**

Use `list_items`. This tool lists all the available tours under a specific company, letting you decide which item to check for slots later on.

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 `"fareharbor": { "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

FareHarbor 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 FareHarbor. 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	June 2026
MCP Server	FareHarbor MCP
Server ID	`019d7596-a9a0-72ee-a70b-4a8aa31809ed`
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/fareharbor.