MCP SERVER | NO CODE | CLOUD HOSTED

Upper Route Planner MCP

Orchestrate Complex Delivery Logistics by Conversation

Upper Route Planner connects your AI agent to sophisticated delivery logistics tools. It lets you manage entire fleets and complex routes by optimizing schedules, monitoring drivers in real-time, and provisioning new stops instantly using natural conversation.

A+ Quality Score 100/100

route-optimization | delivery-management | fleet-tracking | logistics | api-integration | scheduling

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/upper-route-planner

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.

Upper Route Planner MCP

6 tools available
Cloud-hosted on Vinkius

Managing a fleet of deliveries is usually a headache involving multiple spreadsheets, status dashboards, and endless calls. This MCP changes that. You connect your agent to gain full control over high-fidelity route orchestration through simple conversation. Instead of clicking into separate systems to check driver assignments or verify stop metadata, you just ask your AI client for the information. It can list all optimized routes, tell you which drivers are assigned where, and even generate new delivery tasks with precise time windows automatically. If you're looking for a central hub to manage this complexity, Vinkius hosts thousands of specialized connectors, making Upper Route Planner one place to orchestrate your entire operation. Your agent handles the heavy lifting: it can pull detailed information about specific stops or check the overall API status, so you stay on top of field progress without leaving your workspace.

Core Capabilities

01 — List all delivery routes

Retrieve a comprehensive list of optimized delivery routes and their associated status metadata.

02 — Get specific stop details

Pull complete, granular information for any single delivery stop or task location.

03 — Create new tasks

Generate and provision entirely new high-fidelity delivery tasks, including time windows and customer data.

04 — Manage drivers and staff

List active delivery drivers and access current resource allocation details for your field workforce.

05 — Monitor API status

Verify the account's operational connection health and track total route orchestration volume.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/upper-route-planner** — connect your AI agent in three steps.

01 Subscribe to this MCP on Vinkius and retrieve your unique API Token from your Upper dashboard settings.

02 Connect your agent using the token. This gives your AI client access to all of Upper's logistics tools.

03 Start talking to your agent. You can ask it to list routes, create tasks, or check stop details via natural language commands.

The bottom line is you get a single conversational layer that lets your AI act as an expert logistics coordinator for your entire fleet.

Built For

Anyone whose job involves moving things from Point A to Point B. This MCP is built for the Operations Lead who spends too much time switching between tracking dashboards and spreadsheets, or the Dispatcher tired of manual route verification.

Logistics Manager

You use this to pull full status reports on your routes and monitor driver progress without manually checking multiple web portals.

Operations Lead

You verify complex delivery metadata, manage task priorities across regions, and ensure the field operations stay healthy by querying the system directly.

Dispatcher

You analyze technical route efficiency or need to quickly provision a batch of new tasks for tomorrow's manifest using simple AI prompts.

What Changes When You Connect

01 Stop manually checking statuses. You can instantly check the status of a stop using get_upper_stop_details, getting proof of delivery data in seconds.

02 Never lose track of personnel. List all drivers with list_upper_drivers and know exactly which resource is assigned to which route right now.

03 Provision tasks on demand. Use create_upper_delivery_task to generate new jobs instantly, complete with precise time windows for the driver.

04 See the big picture fast. List upper routes gives you an overview of your entire day's manifest without clicking into dozens of separate dashboards.

05 Deep context retrieval: Need details on a specific stop? get_upper_stop_details pulls every piece of necessary metadata, keeping everything perfectly aligned for that parcel.

Real-World Applications

Handling an unexpected delay

A driver calls and reports they're stuck at a specific location. Instead of calling dispatch, you ask your agent to check the status using get_upper_stop_details for that stop ID. The agent pulls up the full metadata instantly, allowing you to reroute the remaining deliveries immediately.

Building out tomorrow's manifest

You just signed a new client and need 50 delivery tasks added by morning. You use create_upper_delivery_task multiple times in one session, providing all necessary addresses and time windows until the entire job is scheduled.

Reviewing weekly performance

The manager needs to know how many routes were completed last week. You ask your agent to list upper routes for a specific date range, giving you an immediate overview of completion volume and average trip time.

Verifying system health before launch

Before running the entire day's schedule, you use check_upper_status. The agent confirms your API connection is healthy and ready for the full wave of route orchestration volume.

Patterns to Avoid

Treating it like a simple lookup tool

✗ AVOID

Asking the AI to just 'tell me where John Doe is.' This only provides basic location data and ignores critical operational details.

✓ INSTEAD

To get full context, you must ask for specific stop metadata using get_upper_stop_details or list upper routes. These tools provide the necessary high-fidelity data points.

Over-relying on manual web checks

✗ AVOID

A dispatcher has to open five different browser tabs just to compare the assigned driver against the task priority and the stop's required time window.

✓ INSTEAD

Ask your agent to list_upper_drivers combined with get_upper_stop_details. Your AI client pulls all that complex, cross-referenced information into one answer.

Forgetting to check connectivity

✗ AVOID

Trying to create tasks or pull routes when the API connection is down and getting vague error messages.

✓ INSTEAD

Always run check_upper_status first. This verifies your account-level connectivity before you waste time generating commands that will fail.

The Right Fit

Use this MCP if your primary pain point is managing the *flow* of deliveries, not just reading a single address. You need to know how drivers are assigned, what their current status is, and if the system can handle new tasks reliably. If you only ever need simple mapping or basic tracking (e.g., 'What's the latitude?'), this is overkill; those generic map tools will suffice. However, if your process involves scheduling multiple people, coordinating specific time windows, or monitoring high-volume operations across a whole fleet, then Upper Route Planner is essential because it handles that complex orchestration layer for you. If you only need to view static data, use an analytics tool instead.

The headache of manual logistics tracking

Right now, managing a delivery day means jumping between spreadsheets, the Upper dashboard, and email threads. You copy-paste stop IDs, check driver availability on one screen, and manually update task statuses in another. If you need to know if a route was completed or why it's delayed, you spend minutes clicking through tabs just to verify basic metadata.

With this MCP, your agent handles the cross-referencing. You don't click anything. You simply ask for the status of a stop or list all active routes, and the AI pulls together the verified details from the system in plain English. It’s instant insight without the dashboard overload.

Upper Route Planner with Upper Route Planner MCP

You never have to manually pull driver assignments and compare them against scheduled tasks again. You can use list_upper_drivers combined with get_upper_route_stop, letting your agent perform that complex comparison for you in one query.

It's not just about getting data; it’s about having an operational brain connected to the system. Your AI client acts as a full-time logistics coordinator, giving you true control over the delivery process.

Upper Route Planner: 6 Tools for Logistics

These tools let you list all routes, get granular stop details, add new tasks, track drivers, and verify the system's operational status.

#	TOOL	DESCRIPTION
01	`create_upper_delivery_task`	Adds a brand-new delivery task record into the system.
02	`list_upper_drivers`	Retrieves a list of all active delivery drivers and their current assignments.
03	`get_upper_route_stop`	Fetches specific, detailed information about a single route stop.
04	`list_upper_routes`	Generates a list of all existing optimized delivery routes and their summary details.
05	`get_upper_stop_details`	Accesses complete, high-fidelity metadata for any specific stop location on a route.
06	`check_upper_status`	Verifies the current API connection status of the Upper platform.

See It in Action

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

U List all delivery routes scheduled for today.

I've retrieved your routes. You currently have 8 active high-fidelity routes, including 'North Sector Loop' (Driver: Alex) and 'Downtown Express'. Would you like to see the detailed stop metadata for any of them?

U Create a new delivery task for '123 Tech St' with contact 'John Doe'.

Logistics provisioned! I've successfully generated a new high-fidelity delivery task for 'John Doe' at '123 Tech St'. This record is now live in Upper. Shall I verify the available route technical windows for assignment?

U Check the status of route stop 'stop_456'.

Stop directory orchestrated! Stop 'stop_456' is currently 'COMPLETED' with a recorded high-fidelity proof of delivery. Your API connection is healthy. Shall I retrieve the detailed timing metadata for this stop?

Frequently Asked Questions

01 **How do I start using Upper Route Planner MCP?**

You first subscribe to this MCP on Vinkius and retrieve your API Token from your upper dashboard. Once connected, your agent immediately gains access to all the routing tools.

02 **Can I use Upper Route Planner MCP to add new tasks?**

Yes, you can use create_upper_delivery_task to generate brand-new delivery jobs. You just need to provide the necessary details, like addresses and time windows.

03 **What is the difference between listing routes and getting stop details with Upper Route Planner MCP?**

list_upper_routes gives you a high-level overview of all scheduled trips. get_upper_stop_details drills down to give you every specific piece of metadata for just one single delivery location.

04 **Does Upper Route Planner MCP handle driver assignments?**

Yes, by using list_upper_drivers, your agent pulls the current resource allocation details. This helps you understand who is assigned where across your fleet.

05 **Do I need to use check_upper_status before running routes?**

It's a good habit. Running check_upper_status confirms that the API connection is healthy and ready for high-volume route orchestration, preventing runtime errors.

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 `"upper-route-planner": { "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

Upper Route Planner 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 Upper Route Planner. 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	Upper Route Planner MCP
Server ID	`019dd17e-054f-73e0-ae10-1d3ab87840e1`
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/upper-route-planner.