MCP SERVER | NO CODE | CLOUD HOSTED

OptimoRoute MCP

Orchestrate complex delivery planning from chat.

OptimoRoute MCP handles complete logistics planning from your AI client. Use it to create new delivery orders, plan optimized multi-day routes for multiple drivers, track vehicle GPS in real time, and verify proof of delivery—all without logging into a separate dashboard.

A+ Quality Score 100/100

route-optimization | delivery-management | gps-tracking | fleet-management | proof-of-delivery

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

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.

OptimoRoute MCP

10 tools available
Cloud-hosted on Vinkius

Need to manage complex deliveries? This MCP lets you run your entire route planning process from a single conversation. You can generate new delivery orders, specifying addresses, required service times, or even capacity limits. Once the orders are ready, you tell the agent to queue up an optimization job for specific dates and drivers. The system handles the heavy lifting, allowing you to check the solver status or abort complicated computations if needed. When routes finalize, you can download a full manifest showing every optimized stop assigned to each vehicle. Beyond planning, it monitors operations: get real-time coordinates for any driver in your fleet using the agent, and pull detailed completion records, including photos and signatures, when verifying proof of delivery. Because this MCP lives on Vinkius, connecting to thousands of other services, you can finally keep all your operational data flowing through one central AI client.

Core Capabilities

01 — Create new orders

Adds structured delivery orders to the system with necessary details like addresses and time windows.

02 — Run route planning jobs

Schedules complex routing computations for specific dates and groups of drivers, allowing you to monitor progress or cancel them early.

03 — Track live vehicle location

Pulls immediate GPS coordinates for any driver, giving you a real-time view of the fleet's movements.

04 — Generate final route manifests

Downloads structured reports detailing every optimized stop and assigned sequence for all drivers involved.

05 — Verify delivery completion records

Retrieves the full record of a completed drop-off, including signatures, photos, and notes.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your OptimoRoute API Key.

02 Use your AI client to issue commands, like asking the agent to create a new delivery order or queue up an optimization job for tomorrow's routes.

03 The agent calls the necessary tools, processes the data through OptimoRoute, and returns results—like status updates, manifest downloads, or live GPS coordinates.

The bottom line is you manage entire logistics operations using natural chat commands instead of clicking through multiple web dashboards.

Built For

This is for the dispatch manager who needs to generate routes and track drivers on the fly, or the ops analyst who has to audit hundreds of delivery proofs at the end of the day. You need a single source of truth that doesn't require you to switch tabs.

Dispatch Manager

Creating urgent orders and monitoring real-time driver positions during active delivery windows.

Logistics Planner

Queueing large route optimization jobs for next week and downloading the final manifests without ever opening the main OptimoRoute website.

Operations Analyst

Pulling proof of delivery data and checking scheduling history to ensure compliance reports are accurate for billing.

What Changes When You Connect

01 Don't waste time navigating dashboards. You can use `create_route_order` to add new stops and then immediately call `queue_route_optimization` —all in one conversation.

02 Need to know where the fleet is? Instead of calling dispatch, ask your agent for real-time coordinates using `get_live_driver_gps`. It gives you immediate visibility into every vehicle's location.

03 The audit process gets faster. You can automatically gather proof of delivery details by running `get_order_pod`, pulling signatures and photos without manual data entry.

04 Managing driver capacity is simple. Use `update_driver_shifts` to set working hours or load limits, ensuring the planned route respects actual operational rules.

05 If a plan goes wrong, you don't get stuck. You can check progress using `check_planning_status`, and if it gets stuck, use `abort_route_planning` immediately.

Real-World Applications

The urgent last-minute addition

A client calls needing a drop-off in a new neighborhood. Instead of logging into the system and manually inputting everything, you just tell your agent: 'Create a delivery order for 14 Oak Lane with a 3-hour window.' The agent uses `create_route_order` to add it, then suggests running a minor optimization using `queue_route_optimization`.

Daily operational audit

At the end of the day, you need proof that all 50 deliveries were signed off. You ask your agent to pull the completion details for every order via `get_order_pod`. It gathers all signatures and photos into one report for compliance.

Optimizing a multi-day schedule

You're planning next week's runs. You tell your agent: 'Optimize routes for the North Zone using drivers D1, D2, and D3 for May 15th.' The agent uses `queue_route_optimization`, and when it finishes, you download the full manifest with `download_manifest_routes`.

Mid-day troubleshooting

Driver D2 calls saying he's lost. Instead of having him backtrack manually, you ask your agent for his current location using `get_live_driver_gps`. You see where he is and immediately adjust the route schedule by calling `get_order_scheduling`.

Patterns to Avoid

Treating it like a simple map query

✗ AVOID

Trying to figure out if one driver can make two stops tomorrow just by asking 'Can I deliver here and there?' This ignores time windows, capacity, and route sequencing.

✓ INSTEAD

You must first use `create_route_order` for all addresses. Then, you tell the agent to run a full optimization using `queue_route_optimization`. The system handles the complex scheduling logic.

Running optimizations without constraints

✗ AVOID

Telling the agent: 'Optimize routes now.' This will generate an unusable plan because it ignores that Driver D1 can't work past 5 PM or carry more than 2,000 lbs.

✓ INSTEAD

Always start by updating driver capacity and shifts using `update_driver_shifts`. Then run the optimization using `queue_route_optimization` to make sure constraints are baked into the plan.

Manually copying data after completion

✗ AVOID

Having to go through 10 different email attachments, one for each driver, just to get signatures and photos for audit purposes.

✓ INSTEAD

After delivery is complete, ask the agent to run `get_order_pod`. It pulls all required verification details into a single structured report.

The Right Fit

Use this MCP if your core pain point is orchestrating multi-stop, multi-day logistics planning across multiple field workers. If you need to create orders, run complex optimization jobs based on capacity and time windows, and then track the result in real-time, this tool handles the full lifecycle from start to finish.

Don't use it if your only need is simple single-leg tracking (e.g., 'What is the nearest gas station?'). For that, a basic map service or location lookup tool is enough. If you just need to read historical reports without planning anything new, an accounting ledger MCP might be better. But for any workflow involving creation, calculation, and monitoring of physical routes, this OptimoRoute MCP is necessary.

The current system requires constant context switching.

Right now, planning a major delivery run means opening the dedicated logistics dashboard. You input the orders there. Then, you have to open a separate tracking tool just to see if the drivers left on time. If something changes—a cancellation or a new stop—you have to manually delete old records and re-upload everything into the right tab.

With this MCP, you don't touch any dashboards. You talk to your agent about the plan: 'We need to add three stops next week.' The agent uses `create_route_order` to handle that data change, then queues up a new optimization job with `queue_route_optimization`. It handles all the messy database work and just gives you a simple confirmation.

OptimoRoute MCP delivers full control over your deliveries.

The biggest time sink is reconciling data. You manually check if the planned route matches what was actually delivered, cross-referencing manifest downloads with physical proof of delivery photos and signatures from different sources.

Now, you ask your agent to pull everything together using `get_order_pod`. The MCP pulls the completion details directly, matching them against the original orders. You get a clean, single report that proves exactly what happened on site.

OptimoRoute: 10 Tools for Logistics Management

These tools let you manage every part of the logistics process, from creating initial orders to tracking live driver GPS coordinates and finalizing proof of delivery.

#	TOOL	DESCRIPTION
01	`get_order_pod`	Retrieves the official record verifying that a specific delivery was successfully completed.
02	`create_route_order`	Adds a new, bounded delivery stop or order to be included in future route calculations.
03	`delete_stale_order`	Removes old or inactive orders from the system so they don't mess up current routing plans.
04	`get_live_driver_gps`	Gets the most recent, real-time GPS location data for any driver in your assigned fleet.
05	`download_manifest_routes`	Generates a structured file listing all optimized stops and the sequence of travel routes for the drivers.
06	`check_planning_status`	Checks if a scheduled route optimization job is finished, or if it's still running in the background.
07	`get_order_scheduling`	Looks up the specific estimated time of arrival (ETA) for an order within the overall planned schedule.
08	`queue_route_optimization`	Starts a new, large-scale route optimization job by feeding in solver logic and required parameters.
09	`abort_route_planning`	Stops a running or stuck route planning process that needs to be canceled immediately.
10	`update_driver_shifts`	Updates the working hours and operational constraints, like maximum load capacity, for individual drivers.

See It in Action

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

U Create a delivery order for 200 Main St, Boston with a 2-hour time window from 9 AM to 11 AM.

Order ORD-4521 created successfully. Destination: 200 Main St, Boston, MA. Service time: 15 minutes. Time window: 09:00–11:00. The order is queued and ready for the next route optimization run. Want me to start a route optimization for today?

U Start route optimization for tomorrow with drivers D1, D2, and D3.

Route optimization queued. Planning ID: PLN-7789. Solver is running with 3 drivers and 42 pending orders for April 1. Estimated completion: ~45 seconds. Want me to check the status in a moment?

U Where is driver D2 right now?

Driver D2 is currently at 42.3601, -71.0589 (near Faneuil Hall, Boston). Last ping: 12 seconds ago. Speed: 28 km/h. Currently en route to stop 4 of 8 (ETA: 5 minutes). Want me to pull the full route manifest for D2?

Frequently Asked Questions

01 **How do I start route planning with OptimoRoute MCP?**

You first need to subscribe and enter your API Key into the agent. Then, you can use `create_route_order` to input all necessary stops before running the optimization via `queue_route_optimization`.

02 **What is the difference between using OptimoRoute MCP for tracking versus planning?**

Planning uses tools like `queue_route_optimization` to create a future route. Tracking uses `get_live_driver_gps` to report current, real-time location data during an active run.

03 Can OptimoRoute MCP handle cancelled stops?

Yes. If a stop is no longer needed, you use `delete_stale_order`. This keeps your routing calculations clean and prevents the agent from trying to route an invalid address.

04 How do I get all the final delivery proof using OptimoRoute MCP?

Use the `get_order_pod` tool. It automatically collects structured verification data, including photos and signatures, confirming successful delivery for audit purposes.

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 `"optimoroute": { "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

OptimoRoute 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 OptimoRoute. 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	OptimoRoute MCP
Server ID	`019d75ea-c4fe-7205-93ea-4fb7cacec012`
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/optimoroute.