

MCP SERVER

NO CODE

CLOUD HOSTED

Delivery Route Stop Sequencer MCP

Turn complex delivery windows into precise, actionable timelines.

Delivery Route Stop Sequencer MCP solves the math behind delivery logistics. It takes your travel times, service durations, and delivery windows to build efficient stop orders. Your AI client can use this to check if a route is actually possible or to figure out exactly how much buffer time you have before a driver runs late.

A+ Quality Score 100/100

delivery

route-optimization

logistics

scheduling

travel-time



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

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.

Delivery Route Stop Sequencer MCP

4 tools available

Cloud-hosted on Vinkius

You can stop guessing the best order for your deliveries. This MCP acts as an optimization engine that handles the heavy lifting of route sequencing. Instead of manually checking if a driver can make it from point A to point B within a specific window, you let your AI client run the numbers. It uses a travel-time matrix and your specific service durations to build a full timeline. You can test different route ideas to see if they are physically possible or use it to analyze how well a driver performed at a specific stop. It's built to handle the constraints that make real-world logistics difficult, like tight time windows and varying service times at each location.

Core Capabilities

01 — Route Sequencing

Your agent uses this to determine the best order for a list of stops.

03 — Buffer Estimation

Your agent calculates the time remaining before a delay impacts the next stop.

02 — Constraint Validation

Your agent checks if a proposed route respects all time windows and service times.

04 — Performance Analysis

Your agent pulls metrics to see how efficient a specific stop was.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/en/ai-agent-connect/delivery-route-stop-sequencer — connect your AI agent in three steps.

- 01

Connect your preferred MCP-compatible client to Vinkius.
- 02

Provide your stop data, travel times, and time windows to your AI client.
- 03

Ask your agent to calculate the sequence or validate a route.
- 04

Receive the optimized timeline or feasibility report directly in your chat.

Connect your AI client to Vinkius to start running optimization math immediately.

Built For

Logistics professionals and dispatchers use this to automate the math behind route planning.

Dispatchers

Hand off stop lists and time windows to the AI to generate valid routes.

Fleet Managers

Use the efficiency metrics to review driver performance and route accuracy.

Logistics Analysts

Provide travel matrices to the AI to test different routing scenarios.

What Changes When You Connect

- 01

Calculates full timelines based on travel-time matrices.
- 02

Verifies route feasibility against strict time windows.
- 03

Identifies timing risks through buffer calculations.

04 Provides specific performance metrics for individual stops.

Real-World Applications

Daily Route Planning

Generate a sequence of stops that respects every customer's delivery window.

Route Feasibility Testing

Check if a driver can actually make a specific set of stops without being late.

Driver Performance Review

Analyze stop efficiency to see where time is being lost during a route.

Risk Management

Calculate how much extra time is available in a route to account for traffic or delays.

Patterns to Avoid

Manual sequence guessing

❌ AVOID

You try to eyeball a list of 20 stops and manually check if the driver can make it between them. It takes forever and you'll likely miss a time window.

✓ INSTEAD

Let your agent run the numbers using `calculate_optimal_sequence` to build a complete, mathematically sound timeline.

Ignoring buffer margins

❌ AVOID

You build a route that hits every window perfectly but leaves zero room for traffic or a slow unloading process.

✓ INSTEAD

Use `estimate_route_window_buffer` to see exactly how much wiggle room you have before a delay ruins the schedule.

Assuming all routes are feasible

❌ AVOID

You send a driver a list of stops that looks good on a map but violates several strict customer time windows.

✓ INSTEAD

Run the proposed order through `validate_route_constraints` to confirm it actually works with your service durations.

The Right Fit

Logistics professionals and dispatchers should connect this MCP if they manage complex routes with strict time windows. If your routes are simple point-to-point trips with no constraints, you don't need this. The deciding factor is whether you need to mathematically prove a route is possible before a driver leaves the depot.

Delivery Route Stop Sequencer: 4 Specialized Tools

These tools handle the heavy math of route sequencing, constraint validation, and performance tracking.

#	TOOL	DESCRIPTION
01	calculate_optimal_sequence	This tool finds the most efficient order for your stops and builds a complete route timeline.
02	estimate_route_window_buffer	This tool calculates how much flexibility you have in a route before a stop becomes late.
03	get_stop_efficiency_metrics	This tool analyzes how a specific stop performed within a completed route.
04	validate_route_constraints	Use this to check if a specific sequence of stops is actually possible given your time windows and service durations.

See It in Action

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

- U** Calculate the best sequence for these stops: travel times `[[0,10,20],[10,0,15],[20,15,0]]`, stops: `[{'id':1,'windowStart':0,'windowEnd':30,'serviceTime':5}, {'id':2,'windowStart':10,'windowEnd':40,'serviceTime':5}]`, depotIndex: 0, startTime: 0.



The optimal sequence is Stop 1 followed by Stop 2. Stop 1 arrives at 10 and departs at 15. Stop 2 arrives at 30 and departs at 35.

- U** Is this route valid? Travel times `[[0,5],[5,0]]`, stops: `[{'id':1,'windowStart':0,'windowEnd':10,'serviceTime':5}, {'id':2,'windowStart':2,'windowEnd':8,'serviceTime':5}]`, sequence: `[0,1]`, startTime: 0.



The route is valid. Stop 1 is reached at 5 and Stop 2 is reached at 10.

- U** What is the efficiency of a stop that arrived at 15 with a 5 minute travel time and 10 minute service duration?



The stop has an idle ratio of 0.0 and a travel ratio of 0.33.

Frequently Asked Questions

01 What data does this MCP need to calculate a route?

You need to provide a travel-time matrix, a list of stops with their service durations and time windows, a starting time, and the depot index.

02 Can I use this to check if a driver will be late?

Yes. You can use the tool to estimate the route window buffer, which shows how much flexibility exists before a stop becomes late.

03 How does it handle service times?

The engine includes the service duration at each stop when calculating the arrival and departure times for the entire timeline.

04 Which AI clients can use this MCP?

Any MCP-compatible client like Claude, Cursor, or Windsurf can use this once connected via Vinkius.

05 Does this MCP host the data?

Vinkius hosts the MCP and manages the execution, so you just need to provide the data to your AI client.

06 How do I find the best route for my drivers?

You can use the ``calculate_optimal_sequence`` tool by providing the travel time matrix, stop requirements, and depot start time to get the most efficient sequence.

07 Can I check if a route is valid before sending it to a driver?

Yes, the ``validate_route_constraints`` tool allows you to verify if a specific sequence of stops adheres to all delivery window and service duration constraints.

08 How can I see how much delay risk exists for a stop?

Use the ``estimate_route_window_buffer`` tool to calculate the buffer time and risk level for any specific stop in your route.

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 <code>"delivery-route-stop-sequencer": { "url": "..." }</code>
 Windsurf	MCP Settings → <code>mcp_settings.json</code> → 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

Delivery Route Stop Sequencer 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 Delivery Route Stop Sequencer. 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	October 2026
MCP Server	Delivery Route Stop Sequencer MCP
Server ID	01a0ed26-1704-70f1-ab04-55a8cbd3203b
Platform	Vinkius Cloud for AI Agents
Endpoint	<code>https://edge.vinkius.com/{token}/mcp</code>

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/en/ai-agent-connect/delivery-route-stop-sequencer.