

MCP SERVER

NO CODE

CLOUD HOSTED

Navitia MCP

Master European Transit Planning & Tracking

Navitia gives your agent complete control over European public transit data, from Paris to Lyon. Plan complex trips combining buses, metros, trains, walking, and cycling across multiple countries. Check real-time departures, monitor service disruptions, or generate accessibility maps showing reachable areas within minutes.

A+ Quality Score 100/100

journey-planning

real-time-transit

multimodal-transport

isochrone-mapping

public-transportation



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.

Navitia MCP

11 tools available

Cloud-hosted on Vinkius

Need to plan a trip across France? This MCP connects your AI client to Navitia's European transit data. Instead of jumping between different operator websites or sifting through raw feeds, your agent handles the whole thing naturally. You can ask it to figure out the best way from Gare du Nord to the Eiffel Tower—combining metro lines, buses, and even suggesting a scenic walk segment. It doesn't just plan the route; it tracks real-time arrivals, warns you about strikes or maintenance closures across the network, and tells you exactly what stops are near your current location. Vinkius hosts this MCP, letting any AI agent use its full suite of tools to act like a dedicated European travel planner for anything from urban planning analysis to simple daily commutes. It handles all major modes—metro, tram, bus, rail, and more—giving you one source of truth for getting around the continent.

Core Capabilities

01 — Plan multi-modal trips

It calculates complete itineraries combining trains, metros, buses, walking, cycling, and cars between any two points.

03 — Monitor service disruptions

The MCP checks for active alerts, strikes, maintenance works, or other operational notices across entire transit networks.

05 — Find nearby stops and lines

You can quickly discover all transit stations, bus stops, or metro points near any coordinates or address.

02 — Check real-time departures

Your agent gets immediate updates on when the next train or bus leaves a specific stop, including delay status and platform numbers.

04 — Analyze geographic accessibility

It generates detailed maps showing every area you can reach within a specific time limit from a given starting point.

One Click on Vinkius — From Prompt to Execution

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

- 01 Subscribe to this MCP on Vinkius and enter your unique Navitia API key.
- 02 Your AI client uses the natural language prompt—for example, 'What's the best way from X to Y?'
- 03 The agent calls the necessary tools (like `plan_journey`) and returns a detailed, actionable itinerary with times, transfers, and disruptions.

The bottom line is, you don't need to understand GTFS feeds or operator APIs; your AI client handles all that complexity for you.

Built For

Anyone who has to coordinate movement across European cities—whether it's a tourist, a logistics manager, or an urban planner. You're tired of opening three different transit websites just to figure out one route.

Travel Agent

Building complex itineraries for clients that span multiple modes and require real-time disruption alerts.

Urban Planner

Analyzing a city's infrastructure by generating isochrone maps or comparing network coverage areas to assess accessibility gaps.

Logistics Coordinator

Planning multi-stage deliveries that rely on public transport for last-mile movement, needing precise departure and arrival times.

What Changes When You Connect

- 01 Stop guessing routes. Use `plan_journey` to get accurate, multi-modal itineraries combining buses, metros, trains, and even bikes in one single output.

-
- 02 Never show up late again. The MCP checks for active service disruptions using `get_disruptions`, letting you know instantly if there's a strike or maintenance work on your planned route.

 - 03 Beyond just routes, use `get_isochrone` to analyze city growth or real estate potential by mapping out exactly what area is reachable within 30 minutes.

 - 04 Need local context? Use `search_places` and `get_nearby_stops` to quickly identify the nearest operational station or stop point from any coordinates.

 - 05 It handles all European operators. Instead of needing knowledge of RATP, SNCF, and TCL separately, this MCP manages them all under one roof.
-

Real-World Applications

A tourist needs to navigate a complex city center trip.

The agent uses `plan_journey` with the start and end coordinates. The resulting itinerary details which combination of metro lines and buses is fastest, while also flagging any minor delays found via `get_arrivals`.

A logistics team needs to confirm service reliability before dispatching personnel.

The agent checks `get_disruptions` for the entire region and then uses `get_lines` to identify which specific transport lines are running smoothly, saving hours of manual cross-referencing.

An urban planner needs to assess a new development site.

The agent calls `get_isochrone` for the proposed address, showing exactly which residential neighborhoods are within a 45-minute transit radius. This data helps validate market potential.

A user is lost near a major hub and needs immediate alternatives.

The agent calls `get_nearby_stops` using the current GPS coordinates. It presents three options: walking to a nearby bus stop, taking the metro, or finding an alternative line with minimal transfers.

Patterns to Avoid

Comparing different API documentation.

X AVOID

Trying to manually piece together data from the RATP developer portal and the SNCF feed because you think one is better than the other. This takes hours of parsing XML/JSON.

✓ INSTEAD

Use this MCP's `plan_journey` tool. It handles all major European operators, providing a unified itinerary regardless of which company owns the line.

Ignoring service alerts.

X AVOID

Planning a trip using only scheduled times found via `get_stop_schedule` and getting stuck because there was an unscheduled strike or maintenance closure.

✓ INSTEAD

Always check `get_disruptions` first. If it flags any major issues, adjust your plans before relying on fixed timetables.

Searching by vague location names.

X AVOID

Trying to figure out if a destination is covered because you only know the general area name, leading to incorrect or incomplete results.

✓ INSTEAD

Start by calling `get_coverage`. This confirms which metropolitan areas are supported before running any line search or journey plan.

The Right Fit

Use this MCP if your core need is comprehensive European transit intelligence. If you must compare different modes (bus to train, metro to bike), use the `'plan_journey'` tool. Don't use it if your goal is simply checking a static timetable; while `get_stop_schedule` works, `plan_journey` offers more context and transfer options. Conversely, don't rely on this solely for general mapping or local point-of-interest data—use `search_places` for that instead. This MCP shines when you need real-time status combined with multi-modal route planning and deep network analysis.

Getting around Europe feels like juggling a dozen different websites.

Today, if you're trying to plan any major trip across French or European cities, you open three tabs: one for the metro, one for the regional train, and one for buses. You copy-paste stops into each site, manually check schedules, then cross-reference them all to see if they run on weekends, dealing with conflicting data points every single time.

With this MCP, you just ask your agent: 'How do I get from Point A to Point B?' Your AI client handles the entire process. It checks real-time departures, combines metro lines with regional rail, and even suggests a walk segment—giving you one simple, reliable itinerary.

Navitia MCP gives you total operational visibility.

Manual planning only shows the 'ideal' route. You miss critical context like knowing if there's an active strike on a major line, or what alternative routes are available right now because of construction work.

Now, your agent checks `get_disruptions` first. It doesn't just give you directions; it gives you confidence that the journey can actually happen.

Navitia: 11 Tools for Transit Operations

These eleven tools allow your agent to perform everything from finding nearby stations to calculating complex multi-modal routes across Europe.

#	TOOL	DESCRIPTION
01	<code>get_arrivals</code>	Retrieves a list of incoming services at a stop with scheduled and real-time expected arrival times.
02	<code>get_coverage</code>	Lists all metropolitan areas and cities the transit data covers, letting you know what regions are accessible through the MCP.
03	<code>get_departures</code>	Retrieves a list of outgoing services from a stop with scheduled and real-time departure times.
04	<code>get_disruptions</code>	Pulls active service alerts, including strikes, maintenance works, or incidents affecting specific lines across the region.
05	<code>get_isochrone</code>	Generates a geographic map showing all areas reachable from a point within a set time limit.
06	<code>plan_journey</code>	Calculates the best multi-modal route between two points, including trains, buses, walking, and transfers.
07	<code>get_lines</code>	Lists all operational transit lines in a region, providing details like their mode (metro, bus) and operator affiliation.
08	<code>get_nearby_stops</code>	Finds and lists specific transit stops or stations near a given geographic coordinate.
09	<code>get_networks</code>	Provides a list of all major transit operating companies and network authorities in the covered region.
10	<code>search_places</code>	Searches for specific types of locations, like stations or addresses, using an autocomplete style search.
11	<code>get_stop_schedule</code>	Pulls the full timetable for a single stop, detailing service frequency and operating hours across weekdays and weekends.

See It in Action

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

U Plan a trip from Gare du Nord to the Eiffel Tower using public transit in Paris.



Planning multimodal journey with Navitia... I found 3 options: Option 1 (fastest): Metro Line 4 from Gare du Nord to Saint-Michel, then RER C to Champ de Mars-Tour Eiffel — 28 minutes total, 1 transfer, fare ~€2.15 with t+ ticket, departure at 09:42, arrival at 10:10. Option 2 (direct): Bus 42 from Gare du Nord to Champ de Mars — 42 minutes, no transfers, fare €2.15, departure at 09:38, arrival at 10:20. Option 3 (scenic walk): Metro Line 5 to Place d'Italie, then Line 6 (aerial) to Bir-Hakeim, 8-minute walk to Eiffel Tower — 35 minutes, 1 transfer, elevated metro segment with panoramic views. All options show normal service with no disruptions. Would you like step-by-step walking directions for any option?

U Show me all metro departures from Chatelet station in the next 20 minutes.



Searching for Chatelet metro stops... Found multiple platforms. Fetching real-time departures: Line 1 (direction La Defense — departing in 3 minutes, on-time), Line 1 (direction Vincennes — 7 minutes, on-time), Line 4 (direction Bagneux-Lucie Aubrac — 4 minutes, on-time), Line 4 (direction Porte de Clignancourt — 9 minutes, delayed 2 minutes), Line 7 (direction Villejuif-Louis Aragon — 6 minutes, on-time), Line 7 (direction La Courneuve-8 Mai 1945 — 11 minutes, on-time), Line 11 (direction Rosny-Bois-Perrier — 8 minutes, on-time), and Line 14 (direction Aeroport d'Orly — 12 minutes, on-time). All services operated by RATP. No active disruptions on any lines. Would you like detailed route information for any specific line?

U What areas can I reach within 45 minutes by public transit from Lyon Part-Dieu station?



Generating isochrone from Lyon Part-Dieu with 45-minute radius (2700 seconds)... The reachable area covers most of Lyon metropolitan area including: City center (Presqu'île) — 15 minutes by Metro A, Villeurbanne — 10 minutes by Metro A or Tram T1, Confluence district — 20 minutes by Tram T1, Gerland — 15 minutes by Metro B, Vaise — 20 minutes by Metro D, Caluire-et-Cuire — 25 minutes by Bus C, Ecully — 30 minutes by Tram T6, Bron — 30 minutes by Tram T3, Venissieux — 25 minutes by Metro D, and Oullins — 15 minutes by Metro B. Total accessible area spans approximately 12km radius. Would you like the full GeoJSON polygon for mapping, or details about accessibility to any specific neighborhood?

Frequently Asked Questions

01 How does Navitia MCP help with real-time tracking?

It uses `get_arrivals` and `get_departures` to pull live data, showing if a train is running on time or delayed. It's essential for passenger pickup coordination.

02 Can I use Navitia MCP to plan journeys across different countries?

Yes, the MCP supports multimodal trips across major European networks and operators like SNCF and RATP in a single request.

03 What is the difference between `get_nearby_stops` and `search_places` using Navitia MCP?

`search_places` finds general locations, stations, or addresses by name. `get_nearby_stops` specifically lists official transit stops and points of interest around a given coordinate.

04 Does the MCP only handle trains and metros?

No, it handles everything—buses, trams, regional rails, metro lines, walking paths, cycling routes, and even car segments for full journey planning.

05 How accurate is the service disruption data in Navitia MCP?







The MCP aggregates active alerts from multiple operators, covering strikes, maintenance, and weather impacts across all modes of transport.

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>"navitia": { "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

Navitia 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](https://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 Navitia. 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	Navitia MCP
Server ID	019d75db-4b9d-73bf-ba74-b12b2092e502
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/navitia.