MCP SERVER | NO CODE | CLOUD HOSTED

CARTO MCP for AI Agents

Analyzing Service Areas and Optimal Driving Routes

CARTO lets your AI agents run spatial SQL directly against cloud data warehouses (BigQuery, Snowflake). It handles everything from geocoding thousands of addresses to calculating complex service areas and optimal driving routes.

F Quality Score 3.6/100

spatial-analytics | gis | spatial-sql | location-intelligence | data-warehousing | geospatial-data

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

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.

CARTO MCP

10 tools available
Cloud-hosted on Vinkius

This MCP gives your agent direct access to CARTO's full suite of cloud-native spatial analytics. You don’t need to touch the GIS interface or write complex API calls for every task. Instead, you simply ask your AI client to perform actions like 'Find all stores within a 15-minute drive of downtown.'

It translates that request into the necessary geospatial commands: running quick SQL queries on your warehouse data, batch processing hundreds of addresses, or generating accurate travel isoline boundaries.

The power comes from being able to orchestrate these complex steps conversationally. Whether you're importing a massive CSV file with coordinates or comparing two datasets using spatial joins, your agent handles the heavy lifting. Since it connects via Vinkius, you access all of this capability—from data ingestion to final map generation—through one single connection point.

Core Capabilities

01 — Run Spatial SQL Queries

Execute quick analytical queries or long-running batch transformations directly against your BigQuery, Snowflake, or Redshift data warehouse.

02 — Batch Geocode Addresses

Convert large lists of unstructured street addresses into precise latitude and longitude coordinates efficiently in a single request.

03 — Calculate Service Areas (Isolines)

Generate polygons showing the area reachable by car or foot within a specified travel time or distance from any given point.

04 — Find Optimal Driving Routes

Determine the shortest, most efficient driving path between two points, including total distance and estimated duration.

05 — Import External Data Files

Upload massive external datasets—like CSV or GeoJSON files—from a public URL and load them into your connected data warehouse for analysis.

One Click on Vinkius — From Prompt to Execution

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

01 First, subscribe to the CARTO MCP on Vinkius and provide your active CARTO Organization credentials and API Key.

02 Next, give your AI client permission to run spatial commands against your data warehouse—whether that's running a simple query or submitting an async job.

03 Finally, ask your agent for the analysis you need. It executes the required steps (like geocoding addresses or calculating routes) and hands you the structured, usable result.

The bottom line is: you talk to your AI client like a colleague, and it handles all the complex geospatial data plumbing.

Built For

This MCP is essential for any professional who relies on location intelligence or large-scale spatial datasets. If your job involves finding optimal routes, analyzing service territories, or correlating physical addresses with database records, this is for you.

Logistics Strategist

Uses the MCP to calculate 15-minute delivery isolines around new warehouse sites and cross-references those zones against population density data.

Data Scientist / GIS Analyst

Feeds a batch of customer addresses into the system and uses spatial SQL to join the resulting coordinates with transaction history for deep pattern analysis.

Real Estate Developer

Asks their agent to find optimal locations by calculating catchment zones around potential sites, ensuring they meet minimum population requirements.

What Changes When You Connect

01 The `geocode_batch_addresses` tool lets you process thousands of customer addresses in one go, saving hours of manual data cleanup.

02 You can run complex queries using `execute_sql_query` to quickly join spatial datasets with standard business metrics without leaving the chat window.

03 Calculating service boundaries is simple: use `calculate_isoline` to instantly graph all areas within a 15-minute drive of a store site for planning.

04 Handling large data transfers? Just point your agent at an external file URL, and `import_external_file` loads it directly into your warehouse.

05 The asynchronous job tools (`create_async_sql_job` , `poll_async_job_status`) handle massive ETL tasks that would time out if run in a simple query.

Real-World Applications

Planning New Store Locations

A retail strategist needs to test five potential store sites. They ask their agent to use `calculate_isoline` for a 10-minute drive time around each site, letting them visualize the full catchment area and select the most viable spot.

Auditing Customer Data Quality

A data scientist receives a spreadsheet of 5,000 addresses. They use `geocode_batch_addresses` to validate every single entry against real-world coordinates before running any expensive analytics queries.

Running Large-Scale Data Migrations

A data engineer needs to run a complex spatial join across two massive tables. Instead of timing out, they use `create_async_sql_job` and then monitor progress with the related polling tools.

Calculating Supply Chain Efficiency

The operations team asks their agent to calculate the fastest driving path between a regional distribution center and three client sites using `calculate_route`, optimizing fuel consumption for the fleet.

Patterns to Avoid

Running complex joins manually

✗ AVOID

Trying to build a massive spatial join query piece by piece in the GUI, risking timeouts or forgetting necessary filters.

✓ INSTEAD

Instead, let your agent execute the entire workflow using `execute_sql_query` or submitting it as a long-running job with `create_async_sql_job`.

Handling addresses one by one

✗ AVOID

Manually entering dozens of street addresses into a geocoder, which is slow and prone to human error.

✓ INSTEAD

Always use the specialized `geocode_batch_addresses` tool. It handles bulk data efficiently in a single request.

Ignoring job status checks

✗ AVOID

Submitting an asynchronous query and then waiting indefinitely, unsure if it succeeded or failed silently.

✓ INSTEAD

Submit the job with `create_async_sql_job` and then use `poll_async_job_status` to check its state until you get a definitive success or failure message.

The Right Fit

Use this MCP if your core business problem revolves around location: calculating distances, determining service areas, or

analyzing geographical patterns. You need it when raw coordinates are the key ingredient for decision-making.

Don't use this if you just need to read a simple list of users (a general database tool works fine) or if your data already has clean, pre-calculated geometry fields that never change. If all you do is look at static tables without needing spatial joins or location intelligence, you don't need the full power of CARTO.

If the task involves mapping physical locations to business outcomes—like finding a delivery zone (using `calculate_isoline`) or checking if an address exists (using `geocode_address`)—this is your tool.

CARTO MCP for AI Agents: Solving Complex Geospatial SQL Queries

Today, running a spatial query means jumping between tabs. You copy the raw data into a spreadsheet, you run the geocoding in one service, and then you paste those coordinates back into your main database just to execute the final join.

With this MCP, all that happens conversationally. Your agent executes the necessary steps—like running an initial `execute_sql_query` against your warehouse—and returns a clean, finished data set ready for immediate use.

CARTO MCP for AI Agents: Mastering Location Intelligence and Routing

Before this connector, calculating service boundaries or optimal routes required dedicated GIS software and expert knowledge to write the right API calls. It was a multi-step process that always felt brittle.

Now, you simply ask for it. Whether you need a 15-minute drive isoline using `calculate_isoline` or the best path between two points using `calculate_route`, the system handles the complex spatial geometry instantly.

CARTO: 10 Tools for Spatial SQL and Location Intelligence

Use these tools to perform everything from bulk geocoding addresses to running large-scale asynchronous spatial data transformations against your warehouse.

#	TOOL	DESCRIPTION
01	calculate_isoline	Generates a polygon showing the reachable area from a point based on travel time or distance thresholds.
02	calculate_route	Finds the best driving path between two coordinates and returns the total distance in meters and estimated duration in seconds.
03	create_async_sql_job	Submits a large, complex SQL query as an asynchronous job that runs in your data warehouse for heavy computations that take minutes to complete.
04	execute_sql_query	Runs immediate analytical queries against your connected data warehouse using standard spatial functions.
05	geocode_address	Converts a single text address string into precise latitude and longitude coordinates.
06	geocode_batch_addresses	Processes multiple addresses at once, converting entire lists of street names into geo-coded coordinates efficiently.
07	get_import_status	Checks if a previously started data import job is finished and reports the status or any errors that occurred.
08	import_external_file	Loads an external file, like GeoJSON or CSV, into your data warehouse by pointing to its public URL.
09	list_map_datasets	Retrieves a list of all existing map datasets and tables within your CARTO organization for review.
10	poll_async_job_status	Periodically checks the status of an asynchronous SQL job, confirming if it's pending, running, or completed.

See It in Action

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

U I'm opening a new branch. Can you calculate the 15-minute drive area around this proposed location and cross-reference it with available demographic data?

Isoline Analysis Complete

Based on current traffic models, the optimal service zone is generated as follows:

- **Coverage Area:** Approximately 3.2 square miles.
- **Key Populations Included:** >15,000 residents within a 10-minute radius.
- **High Traffic Corridors:** The main routes are covered by the path geometry below.

[GeoJSON Polygon Data Here]

This boundary shows your potential reach and confirms high population density in the southwest quadrant.

U We got a list of 500 store locations. Can you validate all those addresses using real-world coordinates?

Batch Geocoding Results

Successfully processed 500 entries. Here's a sample:

ADDRESS	LAT	LON	STATUS
123 Main St, City A	40.71	-74.01	OK
789 Oak Ave, City B	34.05	-118.25	OK
PO Box 100, Unknown	N/A	N/A	INVALID

Note: The batch process flagged 5 addresses as invalid for manual review.

U We need to run an advanced query that joins our sales data with the census boundary polygons. Can you set it up?

Async SQL Job Initiated

I've submitted your complex spatial join request. The job ID is `job_xyz123`.

- **Status:** Pending (Running)
- **Estimated Time:** 8-10 minutes.

You can check the status anytime using this ID. I will notify you when the results table (`sales_with_census`) is available.

Frequently Asked Questions

01 **How does CARTO MCP help me find optimal routes between sites?**

It calculates the most efficient driving path, giving you not just a line on a map, but precise total distance in meters and estimated time in seconds. This is useful for optimizing your delivery fleet.

02 **Is CARTO MCP good for handling huge lists of addresses?**

Yes. You don't have to geocode them one by one. The batch processing tools allow you to send entire spreadsheets of addresses and get back validated coordinates for every single entry.

03 **What if my data is in a different cloud warehouse, not BigQuery?**

The CARTO MCP supports querying your data directly from Snowflake, Redshift, or PostgreSQL. It works with the spatial features built into those databases.

04 **Can I use this tool for planning new store coverage areas?**

Absolutely. You can calculate service isolines—the actual physical area you'll reach within a set driving time—allowing you to select sites that cover the right population density.

05 **Is CARTO MCP better than writing custom API scripts?**

Yes, it's much easier. Instead of managing multiple API calls for geocoding, routing, and SQL jobs, you just ask your agent to perform the entire multi-step process conversationally.

06 **Does CARTO MCP handle data I need to load from an external file?**

It does. You can provide a publicly accessible URL for any standard format—CSV, GeoJSON, or Shapefile—and the tool will automatically ingest and prepare it in your warehouse.

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 `"carto": { "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

CARTO 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 CARTO. 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	CARTO MCP
Server ID	`019d7569-ffe7-733e-b720-e128eea9f6a3`
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/carto.