MCP SERVER | NO CODE | CLOUD HOSTED

Overpass (OpenStreetMap) MCP

Query global location data with conversational AI.

Overpass (OpenStreetMap) lets you query the world's free geographic database directly through your AI client. You can search for anything from restaurants and hospitals to EV charging stations and specific retail tags across any global area, needing zero API keys or complex coding.

A+ Quality Score 98.33/100

openstreetmap | geographic-data | spatial-query | location-intelligence | mapping

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/overpass-openstreetmap

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.

Overpass (OpenStreetMap) MCP

16 tools available
Cloud-hosted on Vinkius

Need to know what's actually on the ground in a city? This MCP connects your agent to OpenStreetMap, querying massive amounts of public geographic data. You don't need to write complicated queries; you just ask for what you need —like finding all hospitals within a specific neighborhood or locating every coffee shop near a university campus. The result is structured data containing names, addresses, and details for everything found.

Whether you’re planning an urban development project, building a location-aware app, or just trying to find the nearest sushi place in Tokyo, this tool gives you instant access. Since Vinkius hosts this MCP, you connect once from your preferred AI client (Claude, Cursor, etc.) and gain immediate access to querying the world's most comprehensive map data source.

Core Capabilities

01 — Find specific amenities by type

You can search for common points of interest like restaurants, schools, or pharmacies within a defined geographical area.

02 — Track infrastructure and services

Locate critical utilities such as EV charging stations, fuel pumps, ATMs, and public parking spots across large regions.

03 — Deep dive into regional data

Execute custom queries using Overpass QL syntax to retrieve highly specific OSM features not covered by standard searches.

04 — Analyze proximity to a point

Find amenities or resources near a known GPS coordinate without needing to define an entire search box.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius and connect your preferred AI client.

02 Tell your agent exactly what you need, providing necessary geographic boundaries or coordinates (e.g., 'Find all hospitals between these two points').

03 The tool runs the query against OpenStreetMap and returns structured JSON data containing names, addresses, details, and locations of everything found.

The bottom line is that you get clean, actionable location intelligence without writing a single API call yourself.

Built For

Anyone whose job depends on knowing where things are. This includes urban planners analyzing land use patterns, developers building location-based apps, and field researchers needing real-time data verification.

Urban Planner

Determines optimal locations for new infrastructure by querying density of existing amenities like parks or schools across an entire district.

Software Developer

Builds location-aware prototypes that need to pull real-time POI data, such as finding all nearby restaurants based on user coordinates.

Travel & Logistics Coordinator

Plans routes and services by checking the availability of specific resources like ATMs or gas stations along a predetermined path.

What Changes When You Connect

01 Don't waste time on manual map lookups. Instead of clicking through separate map layers for restaurants, you simply ask your agent to 'Find all Italian restaurants near 40.75 N.' and get a direct list from the `search_restaurants` tool.

02 It handles complex geography without code. You don't need to manually calculate bounding boxes for every search; tools like `search_nearby_amenities` let you pinpoint resources relative to any given coordinate.

03 The data is incredibly rich and structured. When you find a hotel using the `search_hotels` tool, you get star ratings, addresses, and phone numbers—not just a vague dot on a map.

04 You can analyze entire regions at once. If you need an infrastructure report for a neighborhood, running a custom query via `custom_query` aggregates data points like parks, schools, and commercial zones simultaneously.

05 It covers all the essentials of daily life. Need to check for medical services? Use `search_hospitals`. Looking for car repairs? Check `search_shops` with specific filters. It's a one-stop geographic intelligence tool.

Real-World Applications

Analyzing neighborhood readiness

An urban planner needs to assess if a new development site has enough infrastructure. They use the MCP to run `search_amenities` and gather counts for hospitals, schools, and pharmacies within the proposed boundaries, confirming it meets zoning requirements.

Planning an emergency response

A coordinator needs to know where medical help is located. They run `search_hospitals` for a specific sector, prioritizing facilities based on their listed specialties and proximity details.

Building a local guide app

A developer needs to populate their prototype with POIs. They use `search_nearby_amenities` repeatedly around a central point to gather data on nearby banks and ATMs, then feed that structured list into the app's database.

Patterns to Avoid

Treating it like general knowledge

✗ AVOID

Asking the agent, 'What is the best place to eat in New York?' The model might give a list of famous restaurants but won't provide structured data with addresses or phone numbers.

✓ INSTEAD

Be specific and use location context. Instead, ask: 'Using `search_restaurants`, find all Italian places between these coordinates.' This forces the tool to return actionable, verifiable data.

Ignoring required parameters

✗ AVOID

Running a query without defining a bounding box or proximity center. The MCP won't know where to look and will fail or return uselessly broad data.

✓ INSTEAD

Always provide the coordinates or the specific area of interest. If you need an entire district, use `search_by_tag` with known boundaries; otherwise, specify a BBox for targeted searches.

Overloading one query

✗ AVOID

Trying to find everything—restaurants, schools, and parks—in one massive `custom_query`. The resulting JSON will be huge and difficult to parse.

✓ INSTEAD

Break it up. Use separate tools like `search_restaurants` then run `search_parks` in a second turn. This keeps the data clean and focused.

The Right Fit

Use this MCP if your job requires verifiable, structured geographic data about physical points of interest. If you need to know 'How many hospitals are within 1km of X?' or 'What are the addresses of all gas stations in Y?', this is what you use. It's perfect for developers building location-aware services or planners needing census-level data.

Don't use it if your goal is purely conversational or abstract. If you just need to know 'How do I get to Times Square?' (which requires real-time traffic/transit routing) or 'What are the general trends in global retail spending?', this tool won't help. For those tasks, a different type of data source or agent is needed.

Manually checking maps for services is a total waste of time.

Right now, figuring out resources in an area means juggling multiple browser tabs: one for Google Maps to check general locations, another for local business directories to get phone numbers, and maybe a third sheet just to track coordinates. You're copy-pasting addresses and manually cross-referencing data points until you finally piece together a usable picture of the neighborhood.

With this MCP, that manual process disappears. You ask your agent once—for example, 'I need every pharmacy and school within these boundaries.' The tool executes all those checks against OpenStreetMap, compiling detailed addresses, specialties, and contact info into one clean data packet you can use immediately.

Overpass (OpenStreetMap) MCP delivers structured geographic data.

The biggest manual step that goes away is the tedious cross-referencing of disparate map layers. You never have to switch between a general search for shops and a targeted query for ATMs again. All the intelligence lives in one place.

Now, you treat geographic data like any other API endpoint—it's direct, fast, and structured. It’s not just showing dots on a map; it gives you the raw, actionable records that power real applications.

Overpass (OpenStreetMap) with 16 Tools

These tools let you execute every type of spatial query imaginable—from simple nearby searches to complex custom database lookups—all through your AI agent.

#	TOOL	DESCRIPTION
01	custom_query	Allows you to execute any custom Overpass QL query against the entire OpenStreetMap database.
02	search_amenities	Searches for general amenities like restaurants, schools, and hospitals within a specified bounding box.
03	search_atms	Retrieves detailed information on ATM locations, including the bank operator and availability status, within a given area.
04	search_by_name	Finds specific OSM features by name while also allowing you to filter results with full details like phone numbers or websites.
05	search_by_tag	Narrows down search results by filtering elements based on specific OpenStreetMap key/value tags in a bounding box.
06	search_charging_stations	Locates and gathers details about EV charging stations, including connector types and charging speeds, within a specified area.
07	search_fuel_stations	Finds gas/fuel stations in a bounding box, providing information on fuel types offered and operating hours.
08	search_hospitals	Searches for hospitals and clinics within a defined area, returning details like specialties and emergency service contacts.
09	search_hotels	Locates hotels in a bounding box, providing key details such as star ratings and contact information.
10	search_nearby	Finds general OSM elements near a single GPS point without needing to define an entire search area.
11	search_nearby_amenities	Searches for specific amenities like banks or schools near a location, returning names and addresses.
12	search_parks	Identifies parks and green spaces within a bounding box, providing details on area size and features like playgrounds.
13	search_pharmacies	Locates pharmacies in a specified region, giving opening hours, phone numbers, and dispensing information.

#	TOOL	DESCRIPTION
14	search_restaurants	Searches for restaurants within an area, optionally filtering by cuisine type like Italian or Mexican.
15	search_schools	Finds schools in a bounding box, providing details on student capacity and contact information.
16	search_shops	Searches for various shops, including supermarkets or clothing stores, within a defined geographical area.

See It in Action

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

U Find all restaurants in Lower Manhattan.

Found 250+ restaurants in Lower Manhattan (bbox: 40.70,-74.02,40.72,-73.98). Includes Italian, Chinese, Japanese, American, Mexican and many more cuisines. Names, addresses, phone numbers and opening hours included.

U Find ATMs within 500m of Times Square (40.7580, -73.9855).

Found 18 ATMs within 500m of Times Square. Operators include Chase, Bank of America, Citibank, Wells Fargo and independent ATM networks. Locations include buildings, stores and subway stations.

U Find EV charging stations in downtown San Francisco.

Found 45 EV charging stations in downtown SF. Includes Tesla Superchargers, ChargePoint, EVgo and Blink stations. Connector types, charging speeds and access info provided.

Frequently Asked Questions

01 **How do I find nearby restaurants using Overpass (OpenStreetMap) MCP?**

You use the `search_nearby_amenities` tool and specify 'restaurant' as the amenity type. You only need to provide your current location or coordinates, and the tool handles the rest.

02 **Can I find specific shops using Overpass (OpenStreetMap) MCP?**

Yes, you can use `search_shops` and filter by common types like 'electronics' or 'bakery.' It returns details for those shops within your specified area.

03 **Is this good for finding medical facilities? Overpass (OpenStreetMap) MCP?**

Absolutely. Use `search_hospitals` to find major care centers, or use `search_pharmacies` if you need local dispensing information and opening hours.

04 **What if I want data that isn't listed in the search_amenities tool?**

For specialized searches, always default to the `custom_query` tool. This lets you write specific Overpass QL syntax to pull out any unique OpenStreetMap feature.

05 **Can I find EV charging stations for free? Using Overpass (OpenStreetMap) MCP?**

Yes, the `search_charging_stations` tool queries this public data. It provides names, addresses, and crucial details like connector types within your bounding box.

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 `"overpass-openstreetmap": { "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

Overpass (OpenStreetMap) 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 Overpass (OpenStreetMap). 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	Overpass (OpenStreetMap) MCP
Server ID	`019d8469-8c5a-7007-9d00-7303f488f7c0`
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/overpass-openstreetmap.