

M C P S E R V E R

N O C O D E

C L O U D H O S T E D

Amsterdam City Status & Incidents MCP

Real-time infrastructure and public service data for your AI agent.

Amsterdam City Status & Incidents MCP provides direct, keyless access to the City of Amsterdam's DSO open-data platform. Your AI client can pull live updates on electricity and gas outages, active road construction, citizen-reported issues, and the status of public street lights and clocks. It connects your agent to the city's live incident feeds without needing manual API credentials.

A+ Quality Score 100/100

amsterdam

outages

road-works

citizen-reports

street-lights

smart-city



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.

Amsterdam City Status & Incidents MCP

5 tools available

Cloud-hosted on Vinkius

You can pull live data from Amsterdam's public service registries directly into your AI workflow. This MCP acts as a bridge to the city's DSO open-data platform, giving your agent the ability to monitor local disruptions as they happen. Instead of manually checking city dashboards, you can ask your AI client to find specific outages or report statuses in real time.

If you are building monitoring tools or local dashboards, this MCP provides the raw data needed to track utility failures, road closures, and neighborhood-level citizen reports. You can filter by postcode, neighborhood, or specific incident types to get precise information. Whether you need to know why a street light is out or where a new roadwork project is starting, the data is available through a few direct commands. It covers everything from energy grid status to the operational state of public clocks and lighting across the city.

Core Capabilities

01 — Utility Monitoring

Your agent tracks electricity and gas outages by status and energy type.

02 — Infrastructure Tracking

The AI monitors road construction and repair projects across the city.

03 — Public Service Auditing

Your client checks the operational status of street lights and public clocks.

04 — Citizen Feedback Analysis

The agent searches through registered citizen incident reports by theme or neighborhood.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/en/ai-agent-connect/amsterdam-city-status-incidents — connect your AI agent in three steps.

- 01 Connect your MCP-compatible client to Vinkius.
- 02 Select the Amsterdam City Status & Incidents MCP.
- 03 Ask your AI agent to retrieve specific city data using natural language.
- 04 The agent calls the relevant tool to pull live data from the city's platform.
- 05 Your agent processes the results and presents the information to you.

Connecting to this MCP gives your AI client immediate access to Amsterdam's live data feeds.

Built For

This MCP is built for professionals managing city operations or building automated monitoring tools.

City Operations Agents

They use the data to monitor real-time disruptions in public infrastructure.

Incident Monitors

They automate alerts for power outages or road closures.

Dashboard Developers

They pull live city data to populate local disruption maps and displays.

What Changes When You Connect

- 01 Eliminates the need for manual API key management and credential handling.

-
- | | |
|----|--|
| 02 | Provides direct access to the Amsterdam DSO open-data platform. |
| 03 | Enables precise filtering by postcode, neighborhood, or incident type. |
| 04 | Delivers real-time updates on utility and street-level disruptions. |
-

Real-World Applications

Emergency Response Planning

An agent monitors power outages to identify areas requiring immediate attention.

Traffic Management

A tool tracks active road works to help predict or report local traffic delays.

Neighborhood Watch Automation

Users monitor citizen reports to stay informed about local issues like water or waste.

Public Asset Maintenance

A system checks the status of street lights to identify broken fixtures for repair crews.

Patterns to Avoid

Broad, unguided searches

❌ AVOID

Asking your agent to find all city problems in Amsterdam at once.

✓ INSTEAD

Use specific filters like neighborhood or postcode with `search_citizen_reports` to get useful, localized results.

Manual dashboard monitoring

❌ AVOID

Refreshing the city's public web portal every hour to check for new outages.

✓ INSTEAD

Automate your checks by having your agent run `get_power_outages` to pull the latest electricity and gas status updates.

Vague infrastructure queries

❌ AVOID

Asking if the streets are okay in a specific district.

✓ INSTEAD

Be precise by using `list_road_works` to check for specific construction notices or project statuses.

The Right Fit

Connect this MCP if you build automated monitoring tools or dashboards for city operations. Do not connect it if you only need general city information without a need for real-time, actionable data. The decision depends on whether your workflow requires live updates from the Amsterdam DSO platform.

Amsterdam City Status & Incidents 5 Tools

This MCP provides direct access to Amsterdam's DSO open-data platform. Use these tools to pull real-time reports on utilities, road works, and citizen-reported issues.

#	TOOL	DESCRIPTION
01	get_power_outages	Get electricity and gas outage reports in Amsterdam. It provides status, energy type, cause, and affected areas via postcodes or polygons.
02	list_road_works	List road works and construction notices on Amsterdam streets. You can filter by work type, status, or specific project names.
03	search_citizen_reports	Search citizen trouble reports registered with the city. Filter results by theme, category, neighborhood, or external status.
04	list_street_light_status	List public street light fixtures and their current status. It includes object numbers, service status, and geographic coordinates.
05	list_public_clocks	List public clocks and their operational status. This tool provides object numbers, status codes, and location data.

See It in Action

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

U Show the open electricity outages right now.



The current open electricity outages include reports for the following areas: [List of affected postcodes, energy types, and start times].

U Which road works are currently in execution?



Active road works include: [List of project names, work types, and location descriptions].

U How many open citizen reports are there for the 'water' theme?



There are [Number] open reports under the water theme, including issues in [Neighborhoods] categorized under [Categories].

Frequently Asked Questions

01 Do I need an API key to use this MCP?

No. This MCP provides keyless access to the public Amsterdam DSO open-data platform.

02 Which AI clients can I use with this?

You can use this with any MCP-compatible client, including Claude, Cursor, Windsurf, and VS Code.

03 Can I filter outages by specific neighborhoods?

Yes. You can filter power outages and citizen reports by postcode, neighborhood, or specific area polygons.

04 Does this include data on street lights?

Yes. The MCP includes tools to list the status and location of public street lights and public clocks.

05 Is the data real-time?

The data comes directly from the city's live incident and public service status feeds.

06 Do I need an API key?

No. The stroomstoringen, wior, meldingen and storingsmeldingen_ovl_en_klokken services of the city's DSO open-data platform are public (OPENBAAR) and work without credentials.

07 How fresh is the data?

Outages and road works are updated by the city's own incident systems on a regular basis and are a near-real-time view of public incidents; citizen reports reflect the register at its sync interval. Treat status values as the official Dutch terms and map them to your own states.

08 Why are the status values in Dutch?

The city publishes the data in Dutch; for example outage status uses values like 'opgelost' (resolved) and road works main status uses 'In uitvoering' (in execution). The tools return the raw values — translate or map them in your agent rather than filtering on a guessed English term.

09 Can I filter outages by postal code?

No. The affected-postcodes field on outage records is a semicolon-joined string, so exact-match filtering by postal code is not meaningful. Filter by status, energy type, outage type or affected area instead, and narrow client-side if you need a specific postal code.

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>"amsterdam-city-status-incidents": { "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

Amsterdam City Status & Incidents 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 Amsterdam City Status & Incidents. 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	September 2026
MCP Server	Amsterdam City Status & Incidents MCP
Server ID	01a0d9c8-335b-7398-9606-e9d3f66d7dff
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/amsterdam-city-status-incidents.