MCP SERVER | NO CODE | CLOUD HOSTED

Uber MCP

Estimate fares and automate trip logistics.

Uber MCP automates travel logistics by giving your AI agent real-time access to ride data. You can estimate fares for any route, track past trips for expense reports, determine available vehicle types at a location, and manage saved addresses—all without opening the Uber app.

A+ Quality Score 100/100

ride-sharing | price-estimation | trip-planning | location-services | logistics | api-integration

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

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.

Uber MCP

9 tools available
Cloud-hosted on Vinkius

This connector lets your AI client handle complex trip planning and cost analysis directly within your workflow. Instead of manually checking multiple websites or calling customer service, you simply ask your agent to figure out travel logistics. You can instantly compare prices for different ride types, check how long it'll take to get a car, and even review detailed records of past journeys to nail down expense reports. By connecting via Vinkius, your AI client gains access to this robust set of tools, letting you manage everything from setting up favorite locations to getting real-time price estimates before booking anything.

Core Capabilities

01 — Estimate Ride Costs

Get instant comparisons of fares across multiple ride types for a specific route.

02 — Manage Locations

Save and recall favorite addresses, like 'Home' or 'Work', using simple aliases.

03 — Check Trip History

Pull detailed records of past trips, including exact dates, distances, and costs.

04 — Identify Available Options

Determine which specific vehicle products (like UberX or Comfort) are running in a given area.

One Click on Vinkius — From Prompt to Execution

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

01 First, connect your AI client using your Uber developer credentials.

02 Next, give your agent a natural language prompt, such as 'What's the cheapest way to get me downtown?'

03 Finally, your agent uses the necessary tools to return specific data points like price estimates and available products.

The bottom line is that you speak naturally to your AI client, and it handles all the complex API calls needed to plan or analyze a trip for you.

Built For

This MCP is essential for frequent travelers, expense managers, and operations teams who spend time coordinating business trips. If you're tired of switching between booking sites or manually calculating costs from receipts, this connector gets the job done.

Executive Assistant

Uses the agent to quickly calculate fare estimates and check available pickup times for multiple potential meeting locations.

Project Manager

Retrieves detailed trip history records after a project phase ends, compiling necessary data points for expense reports.

Field Operations Coordinator

Checks location autocomplete and available ride products in real-time when scheduling multiple team movements across a city.

What Changes When You Connect

01 Get immediate cost comparisons. Instead of guessing, you can use `get_price_estimate` or the more detailed `get_ride_estimate` to know exactly what a ride will cost before you commit.

02 Stop typing addresses repeatedly. You can save and manage favorite places using `add_saved_place`, and then reference them instantly with `get_saved_places` without needing coordinates.

03 Never guess availability again. Use `get_products` to see every single ride option—from basic UberX to premium services—that are actually running at your pickup spot.

04 Streamline expense tracking. Running `get_trip_history` pulls all necessary data points in one go, giving you a clean summary of dates, products, and spending totals for accounting.

05 Improve accuracy when booking. If you're unsure where to drop someone off, running `get_place_autocomplete` helps pinpoint the exact coordinates needed for flawless ride requests.

Real-World Applications

Planning a multi-stop business day

A user needs to take three different rides throughout the afternoon. Instead of manually checking prices for each segment, they ask their agent to get price estimates across all products for the full route. The agent uses `get_price_estimate` and then advises which combination minimizes cost or time.

Analyzing corporate travel spending

The finance team needs a report on employee travel costs from last month. They ask their agent to pull the trip history, using `get_trip_history`, and compile a total spend summary for review, eliminating manual data entry.

Getting ready for an unfamiliar city

A traveler lands in a new area and needs directions. They use their agent to check the available ride products at their hotel's coordinates using `get_products`, immediately knowing if premium cars or basic rides are the best fit.

Quickly booking from a known spot

A user frequently visits a client who doesn't have a standard address. They ask their agent to save the location using `add_saved_place` and then reference it later with `get_saved_places` for fast, accurate bookings.

Patterns to Avoid

Forgetting specific products

✗ AVOID

The agent only checks the standard 'UberX' price when comparing routes, missing out on faster or cheaper options like 'Comfort' or 'Black'.

✓ INSTEAD

Don't rely on one product. Always check available ride products using `get_products` first, then compare costs across all types with `get_price_estimate`.

Manually typing coordinates

✗ AVOID

Entering raw latitude and longitude into a prompt is tedious, prone to typos, and slows down the entire workflow.

✓ INSTEAD

Use `get_place_autocomplete` first. It ensures you have valid, structured addresses that your agent can use for accurate planning.

Confusing time with cost

✗ AVOID

Thinking the fastest product is always the most expensive or vice versa; making assumptions without checking real-time data.

✓ INSTEAD

Compare both speed and price. Use `get_time_estimate` to see pickup times, then use `get_ride_estimate` for that specific product's cost.

The Right Fit

Use this MCP if your primary need is logistics coordination: estimating fares, managing locations, or analyzing travel history. You should connect it when you have a sequence of actions—like 'Compare the cost and time from my saved home location to the airport.' If all you need is basic chat functionality that doesn't involve real-world data (e.g., planning a fictional trip), don't use this. For simple, single-point lookups on public information, an open knowledge base or general search tool might suffice. However, if

the decision hinges on *current* cost, *available* services, or *past* records, this MCP is required.

Dealing with travel requires constant cross-referencing and guesswork.

Right now, planning a simple business trip means bouncing between multiple platforms. You check one site for the price, then another to see how long it takes, and finally open a spreadsheet to record the details. It's clicking through tabs, copy-pasting addresses into different forms, and constantly refreshing pages just to verify availability.

With this MCP, your agent handles all that cross-referencing automatically. You simply ask: 'What is the fastest way from my office to the client site?' The agent doesn't just guess; it uses its tools to check available products, estimate prices, and even confirm the pickup time, giving you a single, actionable answer.

Uber MCP gives you reliable trip data instantly.

The specific manual steps that vanish are checking if your planned route is covered by certain products and manually recording the destination coordinates. You no longer have to worry about whether an address is valid or what services are actually running at a given intersection.

You get immediate, comprehensive data streams for location planning and cost analysis. It's not just about booking; it's about having complete, verified intelligence on every stage of the journey.

Uber MCP: 9 Tools Available

Use these tools to get real-time price comparisons, check historical travel data, and manage all your favorite destinations within the Uber ecosystem.

#	TOOL	DESCRIPTION
01	`add_saved_place`	Saves a new address to your profile, giving it an alias like 'Client Office' for later quick reference.
02	`get_trip_history`	Retrieves a list of past rides, including the start/end points, product used, distance traveled, and final price.
03	`get_user_profile`	Confirms which Uber account your agent is currently logged into for accurate data retrieval.
04	`get_place_autocomplete`	Suggests valid addresses and locations as you type, helping confirm exact pickup or dropoff coordinates.
05	`get_price_estimate`	Calculates a cost estimate for a trip between two specific points using any available ride product.
06	`get_products`	Lists all possible Uber services—like Black or Comfort—available at a given latitude and longitude.
07	`get_ride_estimate`	Provides detailed, product-specific pricing before you book the trip, making comparisons easier.
08	`get_saved_places`	Lists all the custom and saved locations attached to your account for easy reference in planning routes.
09	`get_time_estimate`	Checks how long it will take for a vehicle to arrive at your location, helping you choose the fastest service.

See It in Action

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

U Estimate the price for an UberX from my home to the airport at 3pm tomorrow

I'll get price estimates for all Uber products between your locations right away.

U Show me my last 10 Uber trips with total spending

I'll pull your recent trip history and compile the spending summary.

U What Uber products are available at my current location and how fast can they pick me up?

I'll check available products and pickup times at your location.

Frequently Asked Questions

01 **How do I use Uber MCP to estimate a price?**

Use `get_price_estimate` and provide the starting and ending locations. This tool compares costs across different ride types so you can pick the best option for your budget.

02 **Can I find out my past trip spending using Uber MCP?**

Yes, use `get_trip_history`. It pulls detailed records of every previous journey, including product used, distance, and price, making expense reporting simple.

03 **Does Uber MCP help me find saved addresses?**

Absolutely. Run `get_saved_places` to list all your custom or default locations. You can then use these aliases in future requests instead of typing out the full address every time.

04 **What if I don’t know the exact coordinates?**

Don't worry. Start by using `get_place_autocomplete`. This tool suggests valid place descriptions and structured addresses, helping you narrow down the perfect pickup or dropoff spot.

05 **Is Uber MCP only for ride booking?**

No. It's also great for general planning. You can use `get_products` to see what services are available at a location, even if you aren't ready to book yet.

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 `"uber": { "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

Uber 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 Uber. 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	Uber MCP
Server ID	`019d7617-b788-733c-929f-795ea3bd18e9`
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/uber.