

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

Accessible Ride Request Protocol MCP

Handle complex mobility transit logistics without the manual errors.

Accessible Ride Request Protocol MCP handles the heavy lifting for specialized transportation. It checks if mobility equipment fits in specific vehicles, builds custom pickup checklists, and generates complete ride documentation. Your AI client uses this to ensure every transit request meets strict physical requirements and safety protocols.

A+ Quality Score 100/100

mobility

ride-request

logistics

accessibility

transport



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.

Accessible Ride Request Protocol MCP

4 tools available

Cloud-hosted on Vinkius

You can stop guessing if a wheelchair will fit in a specific van or if a driver has the right equipment. This MCP acts as a logistics engine for accessible transit. When you're planning trips for users with physical requirements, your agent uses this tool to validate equipment dimensions against vehicle constraints. It doesn't just check sizes; it builds the full documentation needed for a successful ride. You can also generate specific handoff checklists for caregivers and verify that emergency contact chains are logically sound. It's built to turn messy accessibility needs into structured, actionable ride requests that providers can actually use.

Core Capabilities

01 — Equipment Validation

Your agent checks mobility device dimensions against vehicle interior clearances.

02 — Documentation Generation

The MCP produces complete ride request packages based on provider rules.

03 — Safety Auditing

Your agent verifies that emergency contact sequences are complete and logical.

04 — Handoff Planning

The tool builds specific checklists for caregivers to use during pickups.

One Click on Vinkius — From Prompt to Execution

Available at vinkius.com/en/ai-agent-connect/accessible-ride-request-protocol — connect your AI agent in three steps.

- 01 Connect your MCP-compatible client to the Vinkius platform.
- 02 Provide your agent with mobility equipment dimensions or vehicle constraints.
- 03 Run validation tools to check fit or verify emergency contact chains.
- 04 Generate the final ride request documentation or handoff checklists.

Connect your client to Vinkius to start managing mobility logistics immediately.

Built For

This MCP is designed for professionals managing specialized transit and mobility services.

Transit Coordinators

They hand off equipment specs and destination data to generate ride requests.

Care Managers

They use the tool to build handoff checklists for caregivers and clients.

Logistics Dispatchers

They use it to validate vehicle compatibility for mobility devices.

What Changes When You Connect

- 01 Prevents equipment mismatch by checking dimensions before the vehicle arrives.
- 02 Reduces pickup delays with custom handoff checklists for caregivers.

- 03 Ensures safety by auditing emergency contact escalation paths.
- 04 Standardizes ride documentation to meet specific provider requirements.

Real-World Applications

Wheelchair Transport Planning

Verify that a specific manual or power wheelchair fits in a scheduled van.

Emergency Protocol Setup

Check that a user's emergency contact sequence is logically sound for transit safety.

Caregiver Coordination

Create a checklist for a caregiver to ensure a smooth transition during a pickup.

Provider Compliance

Generate ride requests that already meet the specific rules of a transport provider.

Patterns to Avoid

Assuming vehicle compatibility

❌ AVOID

You book a standard van for a passenger with a heavy power wheelchair without checking the interior clearance first.

✓ INSTEAD

Run the equipment dimensions through ``validate_equipment_fit`` to ensure the device actually fits the vehicle type.

Manual documentation errors

❌ AVOID

A dispatcher manually types out a ride request and misses a specific provider requirement, leading to a rejected booking.

✓ INSTEAD

Use ``generate_ride_request`` to produce documentation that automatically validates against provider rules.

Weak emergency protocols

❌ AVOID

You set up a contact list for a high-needs passenger but realize during a delay that the primary contact is unreachable and there is no backup.

✓ INSTEAD

Use `verify\_contingency\_chain` to audit your escalation paths and ensure the fallback logic is sound.

The Right Fit

Transit coordinators and care managers should connect this MCP if they handle specialized mobility requests. If you only manage standard ride-share or non-specialized transport, you don't need this. The deciding factor is whether your trips require validating physical equipment dimensions against vehicle constraints.



4 Specialized Tools for Accessible Transit Logistics

These tools handle everything from physical equipment validation to emergency contact auditing and documentation.

#	TOOL	DESCRIPTION
01	build_handoff_checklist	This tool creates a custom checklist for users or caregivers. It ensures a smooth transition during the pickup process.
02	validate_equipment_fit	This tool checks if specific mobility equipment will physically fit inside a designated vehicle type.
03	generate_ride_request	This tool produces a full suite of ride documentation. It validates all user inputs against specific provider rules first.
04	verify_contingency_chain	This tool audits your emergency contact sequence. It makes sure the fallback list is logical and contains all necessary escalation data.

See It in Action

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

- U** Generate a ride request for a wheelchair user from 123 Maple St to 456 Oak Ave with specific ramp requirements.



Request Brief: Pickup at 123 Maple St, Destination 456 Oak Ave. Equipment: Manual Wheelchair (Width: 26in, Length: 38in). Driver must use high-capacity ramp. Confirmation: Does the vehicle have a minimum 26-inch doorway clearance? Handoff: Confirm driver is at the curb and has the ramp deployed.

- U** Check if a 30-inch wide scooter fits in a vehicle with 28 inches of interior width.



The equipment is not compatible because the scooter width (30 inches) exceeds the vehicle's interior width clearance (28 inches).

- U** Verify my emergency contact list: Alice (555-0101) and Bob (555-0102).



The contingency chain is verified. Sequence: Alice, then Bob. Total contacts: 2.

Frequently Asked Questions

01 How does this MCP ensure a wheelchair will fit in a vehicle?

The `validate_equipment_fit` tool compares the specific dimensions of the mobility equipment against the known constraints of the vehicle type.

02 Can I use this with Claude or Cursor?

Yes, you can connect this MCP to any MCP-compatible client like Claude, Cursor, or Windsurf through Vinkius.

03 What kind of documentation does it produce?

The generate_ride_request tool produces a full suite of ride documentation that has been validated against provider rules.

04 Does it help with emergency planning?

Yes, the verify_contingency_chain tool checks that your fallback contact sequences are logical and contain all necessary data.

05 How do caregivers use this tool?

Caregivers can use the build_handoff_checklist tool to get a customized list of steps for a smooth pickup.

06 How does the tool ensure equipment will fit in the vehicle?

The `validate\_equipment\_fit` tool compares the physical dimensions and weight of the mobility aid against the specific interior clearances and ramp capacities of the vehicle.

07 What is included in the ride request package?

The `generate\_ride\_request` tool produces a request brief for the driver, confirmation questions for the user, a handoff checklist, and a fallback contact sequence.

08 Can I customize the pickup instructions?

Yes, the `build\_handoff\_checklist` tool creates a personalized sequence of actions based on your specific transfer preferences and communication needs.

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>"accessible-ride-request-protocol": { "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

Accessible Ride Request Protocol 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 Accessible Ride Request Protocol. 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	Accessible Ride Request Protocol MCP
Server ID	01a0d239-742a-70a0-aadc-f570a80b5bb1
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/accessible-ride-request-protocol.