MCP SERVER | NO CODE | CLOUD HOSTED

Bringg MCP for AI Agents

Manage last-mile delivery tasks and fleet assignments

Bringg MCP gives your AI agent total control over final-mile logistics and dispatch operations. Manage delivery tasks, track entire fleets in real time, and assign drivers using only natural conversation.

F Quality Score 15.83/100

last-mile-delivery | fleet-management | dispatch-automation | delivery-tracking | logistics-optimization | supply-chain

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

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.

Bringg MCP

10 tools available
Cloud-hosted on Vinkius

Need to manage complex last-mile deliveries without switching between five different dashboards? This MCP connects your Bringg account directly to any AI client, letting you handle the full scope of dispatching from a single chat window. You can create new orders on the fly or update old ones; no manual clicks required. If you're using Vinkius, this gives your agent immediate access to all core logistics functions—from checking a customer's history to manually assigning an available driver. It lets you run complex operations like rerouting drivers and closing tasks out in minutes, not hours.

Core Capabilities

01 — Create and Modify Deliveries

You can generate new delivery tasks or update existing order details, such as customer notes.

02 — Control Fleet Status

Manually assign a specific driver to an order, bypassing default routing algorithms when needed.

03 — Manage Task Workflow States

Force tasks into START or COMPLETE status instantly, keeping your dispatch board accurate even if the physical workflow was interrupted.

04 — Track Live Orders and Customers

Pull real-time geolocated data and estimated arrival times for any active order, plus access historical customer records.

05 — Audit Fleet Resources

List all registered human drivers in your network to check their availability and current limits.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Bringg Access Token.

02 Connect the service token to your preferred AI client (Claude, Cursor, etc.).

03 Ask your agent directly for specific actions, like 'Check the timeline for order 3109' or 'Force task 9481 as complete'.

The bottom line is that your AI agent handles all the complex logistics routing and data calls using natural language.

Built For

This MCP is essential for dispatch managers, support reps, and operations teams who spend too much time jumping between multiple screens just to get an order status. If your job involves coordinating physical movement of goods, this connector saves you hours every week.

Dispatch Manager

You use this MCP to reroute drivers mid-shift and manually force task states when the system optimization fails or a driver gets stuck.

Support Representative

When a customer calls, you instantly retrieve their full delivery history and real-time status without leaving your support dashboard.

Operations Team Lead

You build complex end-of-day reports by pulling lists of active tasks and analyzing fleet efficiency across the entire network.

What Changes When You Connect

01 You can manually assign a driver to any task using `assign_driver_to_task`, overriding the system's default routing when you know better.

02 Need to close out an order quickly? Use `force_task_complete` to mark a delivery as finished, even if the driver forgot to update the status in the field.

03 Check real-time locations and ETAs using `get_task_timeline`, giving you immediate visibility into where every package is right now.

04 Support reps can answer customer questions instantly by calling `list_customer_crm` for a full history of past drop-offs.

05 Keep your dispatch board clean by generating new orders with `create_delivery_task` or clearing old ones using `cancel_task_dispatch`.

Real-World Applications

A driver gets stuck and needs a manual reroute

The dispatch manager notices John Doe is stalled on Route A. They ask their agent to use the `assign_driver_to_task` tool, manually directing him to take over two pending orders that were originally scheduled for Sarah Connor.

Handling a no-show delivery

The support rep learns a customer missed their appointment. They use the agent to `update_task_details` immediately, adding a note about rescheduling and calling `list_active_tasks` to confirm the order is still open.

Analyzing fleet capacity at month-end

The operations lead asks their agent to `list_fleet_drivers`, getting a clear roster of all available staff and their current active limits, making efficiency reports simple.

Patterns to Avoid

Trying to update status via the web GUI

✗ AVOID

A dispatcher has to click into Task 9481, find the status dropdown, and click 'Complete.' This takes three clicks every time.

✓ INSTEAD

Instead, tell your agent to run `force_task_complete` for Task 9481. It handles the entire workflow in a single command.

Manually listing all customer details

✗ AVOID

A support rep opens five different tabs just to find if Customer X received an order last week.

✓ INSTEAD

Ask your agent to `list_customer_crm`. It pulls the full historical record instantly, saving you minutes of tedious clicking.

Overriding assignment without context

✗ AVOID

A manager just assigns a driver to a task based on proximity without checking if that driver is actually available or clocked in.

✓ INSTEAD

First, use `list_fleet_drivers` to confirm the driver's status. Then, use `assign_driver_to_task` only after confirming availability.

The Right Fit

Use this Bringg MCP if your daily job involves coordinating physical delivery movements and you need an AI agent to act as a virtual dispatcher. You absolutely need it when you must override default logic, like manually assigning drivers or forcing task statuses; the system sometimes gets messy, and you need direct control. Don't use it, though, if you only need basic tracking information—for simple status checks, your existing dashboard is fine. But if you need to *change* the data (create tasks, update details, force completion), this MCP is non-negotiable.

Bringg MCP for AI Agents: Solving Dispatch Management Headaches

Right now, managing a dispatch schedule means jumping between the main dashboard to see pending tasks, opening another window to check driver availability, and then clicking into the individual task to update its status. It's copy-paste hell that slows down every decision.

With this MCP, you simply tell your agent what needs to happen—for example, 'Mark Task 9481 as done and assign John Doe to the next one.' Your AI client executes all those background steps instantly, giving you immediate operational control.

Bringg MCP for AI Agents: Enhancing Real-Time Fleet Visibility

Before, checking a live timeline required finding the task ID and then opening the dedicated tracking page. If you needed to check multiple orders, it was a repetitive, slow process.

Now, your agent handles that complexity for you. Ask it for 'the timeline on all pending tasks.' It consolidates real-time geolocated data and ETAs into one coherent answer.

Bringg MCP: 10 Tools for Fleet Management & Dispatch

Use these specific tools in your AI agent to create orders, check timelines, force statuses, manage drivers, and handle all aspects of last-mile logistics.

#	TOOL	DESCRIPTION
01	assign_driver_to_task	Manually overrides optimization rules to assign a specific driver to an outstanding task.
02	force_task_complete	Marks any delivery task status as successfully delivered, closing the order record.
03	create_delivery_task	Adds a brand new delivery order into your main dispatch hub for processing.
04	cancel_task_dispatch	Removes an existing delivery task permanently from the active dispatch schedule.
05	get_task_timeline	Retrieves comprehensive, live tracking details and status estimates for a specific order ID.
06	list_customer_crm	Pulls historical data on all delivery recipients associated with your account.
07	list_fleet_drivers	Lists every human driver registered in the fleet, allowing you to check their current status and availability.
08	list_active_tasks	Retrieves a paginated list of all delivery tasks that are currently pending or active.
09	force_task_start	Marks a task as actively started, signaling the driver is en route to the location.
10	update_task_details	Modifies non-status details of an order, such as adding customer notes or changing dropoff instructions.

See It in Action

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

U What are the top 3 active deliveries right now, and who should I assign to the unassigned ones?

Active Deliveries Report

TASK ID	DESTINATION	STATUS
9481...	Main St.	AT PICKUP LOCATION (John Doe)
7421...	Grove St.	CREATED (UNASSIGNED)
3109...	Pine Ave.	ON THE WAY (Sarah Connor)

The second task needs a driver; I recommend assigning available driver ID: 556, who is nearby.

U Can you mark Task 7421 as complete and also update the notes for that customer?

Done. Status updated to **COMPLETE**.

The following details were added:

- **Customer Notes:** Signed receipt obtained from recipient.
- **Task ID:** 7421...
- **Completion Timestamp:** [Current Time]

The order is now closed out in the dispatch system.

U Show me all drivers who are available today and list their IDs.

Available Drivers Today

Here are the human resources currently logged into the Bringg fleet network:

- John Doe (ID: 102) - Available
- Jane Smith (ID: 155) - Limited Capacity
- Marcus Bell (ID: 301) - Available

You can now assign one of these drivers to a new task.

Frequently Asked Questions

01 **How does the Bringg MCP help me manage my daily delivery tasks?**

The Bringg MCP lets you control every step of your dispatch process using natural language. You can create new orders, change details, and update statuses without ever leaving your AI client.

02 **Can I use the Bringg MCP to force a task status if something goes wrong?**

Yes, you can manually override the system by forcing tasks into START or COMPLETE status. This is useful when the driver's app fails but you know the job was done.

03 **Is Bringg MCP good for finding out where a specific delivery order is?**

It provides real-time tracking details and estimated arrival times by pulling the live timeline for any given task. You get exact geolocated data right from your chat window.

04 **Does Bringg MCP help me with driver assignments?**

You can manually assign specific drivers to tasks, bypassing default algorithms if a certain person is required for a route or job. It keeps the assignment control with you.

05 **What kind of customer information can I get using Bringg MCP?**

The MCP connects to your CRM and lets you pull historical data on past delivery recipients, so support reps have instant context for every call or ticket.

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 `"bringg": { "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

Bringg 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 Bringg. 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	Bringg MCP
Server ID	`019d7563-c38d-7157-ba8e-ef37404fcbda`
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/bringg.