MCP SERVER | NO CODE | CLOUD HOSTED

ShipEngine MCP

Compare rates, validate addresses, and print labels.

ShipEngine MCP gives your AI agent full control over complex multi-carrier shipping and logistics tasks. Instantly compare rates across UPS, FedEx, USPS, and dozens of global carriers. You can validate addresses, generate physical labels, programmatically create shipments, and pull real-time tracking status—all through natural conversation.

A+ Quality Score 100/100

shipping-api | label-generation | address-validation | rate-comparison | multi-carrier | logistics-automation

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

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.

ShipEngine MCP

12 tools available
Cloud-hosted on Vinkius

You're running an e-commerce operation, and shipping is the biggest headache. This MCP connects your AI agent directly to ShipEngine's platform, giving you full control over every step of the logistics process. Instead of opening carrier sites and copy-pasting data, you simply ask your agent what you need. It handles comparing rates across multiple carriers, validating if an address will actually work for delivery, and even pulling real-time tracking details from any major service. This makes your AI act like a dedicated shipping clerk. Through Vinkius, the entire catalog of tools is available in one place, so you don't have to switch between specialized services. You can manage labels, check account statuses, and orchestrate complex shipments without leaving your chat window.

Core Capabilities

01 — Compare shipping rates

Get instant cost estimates from multiple carriers for a single shipment based on weight and destination.

02 — Validate delivery addresses

Check if sender or recipient addresses are accurate and deliverable, preventing costly return shipments.

03 — Generate shipping labels

Create finalized, printable labels for a shipment record using the proper carrier format.

04 — Track package status

Retrieve current location and delivery updates for any given tracking ID through natural language queries.

05 — Create shipment records

Establish a formal shipment record that ties together the sender, receiver, and desired carrier details.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP on Vinkius and enter your ShipEngine API Key in the configuration settings.

02 Your AI client authenticates with the service, allowing it to access all carrier data and tools.

03 You issue a command through natural language—for example, 'What are the rates for two boxes from Austin to Miami?'—and the agent executes the necessary checks.

The bottom line is that your AI client handles all the API calls and complex logistics work behind the scenes, giving you only the final, actionable answer.

Built For

Anyone who ships physical goods regularly needs this. Think of the e-commerce owner staring at a stack of shipping labels, or the warehouse manager who just wants to know if an address is valid before printing anything.

Logistics Manager

Uses this MCP daily to compare rates across all major carriers and generate batches of shipping labels without touching a web dashboard.

E-commerce Operations Specialist

Relies on it to validate customer addresses during checkout flow, ensuring the package can actually be delivered before payment is processed.

Software Developer

Integrates multi-carrier rate comparison and label generation into custom back-end code or agent workflows.

What Changes When You Connect

01 Saves time on rate comparison. Instead of checking UPS, FedEx, and USPS websites individually, you ask your agent to use `get_shipping_rates`, which instantly compares all major carriers' costs in one go.

02 Guarantees deliverability upfront. Before you even promise shipping, run the customer's address through `validate_address` to catch typos or incomplete ZIP codes that will cause costly failures later.

03 Simplifies label printing. Generating a physical label is as simple as calling `create_shipping_label`. You get the full metadata and the printable file without manual API integration code.

04 Centralizes tracking data. You don't need to check three different carrier websites; just use `track_package` and your agent pulls the latest status from whichever service handles that ID.

05 Streamlines setup and monitoring. Use `get_connection_status` or `list_carriers` to quickly confirm which services are connected, keeping your logistics infrastructure clean and visible.

Real-World Applications

Need to calculate shipping costs for a new product line.

A founder needs to know the optimal cost per unit. They ask their agent: 'What are the rates for 5kg from Texas to New York?' The agent uses `get_shipping_rates` and presents a clean comparison chart, telling them exactly which carrier offers the best deal.

A customer provided a suspicious or incomplete address.

The support team member asks: 'Can you check this address?' The agent runs `validate_address` and reports back immediately that the street name is invalid, preventing an immediate failed shipment attempt.

Time to fulfill 100 orders for a holiday rush.

The ops manager tells their agent: 'Create labels for these 10 order IDs.' The agent uses `create_shipment_object` first, then calls `get_shipping_rates`, and finally executes `create_shipping_label` for all 10 in sequence.

Need a historical overview of past shipments.

A developer needs to audit recent activity. They ask the agent to 'List my last five shipments.' The agent uses `list_shipments` and provides enough data for them to investigate specific records or pull detailed information using `get_shipment_info`.

Patterns to Avoid

Manual rate checking

✗ AVOID

A user checks UPS's website for rates, then logs into FedEx's site, and repeats the process for USPS. This takes 15 minutes per job.

✓ INSTEAD

Use `get_shipping_rates` with your AI client. Give one prompt asking for a rate comparison across all necessary carriers; it handles the multi-step query instantly.

Assuming address validity

✗ AVOID

The system allows a user to proceed to checkout even if the ZIP code is wrong or the street name doesn't exist.

✓ INSTEAD

Always run `validate_address` first. This tool confirms that the specific combination of street, city, and zip actually works for shipping.

Getting stuck on label formats

✗ AVOID

A developer spends hours trying to format a label file in XML or PDF to meet carrier specifications.

✓ INSTEAD

Let the agent handle it. Use `create_shipping_label` and just provide the necessary shipment data; the MCP handles all the required formatting for that specific carrier.

The Right Fit

Use this MCP if your primary bottleneck is comparing rates, validating addresses, or generating labels across multiple carriers in one workflow. If you need to track a single package and only care about its current location, `track_package` might be enough, but this MCP gives you the full context of creating that tracking ID first.

Don't use this if your problem is purely internal data management (e.g., organizing inventory counts). You don't need to list carriers using `list_carriers` unless you are building a setup guide or auditing your account; for daily operations, focus on the rate and label tools. If all you want is simple reporting on past shipments without generating labels, `list_shipments` works, but this MCP offers the full lifecycle control.

The constant struggle with shipping rates and addresses.

Right now, figuring out how much it costs to ship something is a nightmare of tabs. You have to log into UPS, get their rate, then switch over to FedEx, copy the details, and repeat for USPS—all before you even start building your product page. It's manual, slow, and prone to human error every single time.

With this MCP connected through Vinkius, that entire process disappears. You just ask your agent for a rate comparison; it runs `get_shipping_rates`, pulls data from all the carriers at once, and presents you with optimized options immediately. Your AI acts like an expert logistics analyst.

Generate Shipping Labels Instantly

Previously, generating a label meant jumping into a specific carrier portal, manually entering all the shipment details, and clicking through several forms until you got the PDF. This is tedious and breaks your flow.

Now, you just tell your agent to create the label using `create_shipping_label`. The MCP handles validating the address first, calculating the rates, and then generating the perfectly formatted label file—all in a single conversation.

ShipEngine with 12 Tools

These tools let you manage the entire shipping process: from comparing costs to printing final tracking labels.

#	TOOL	DESCRIPTION
01	`create_shipping_label`	Generates a complete, ready-to-print shipping label for the shipment.
02	`create_shipment_object`	Adds and saves a formal record of an intended shipment to the system.
03	`get_connection_status`	Checks if your API connection key is active and properly set up for use.
04	`get_carrier_details`	Retrieves specific information or metadata about a connected shipping carrier.
05	`get_label_info`	Fetches detailed data and metadata associated with an existing label.
06	`get_shipping_rates`	Compares and returns the current shipping costs from multiple carriers for a given package size and location.
07	`get_shipment_info`	Retrieves comprehensive details about a previously created shipment record.
08	`list_carriers`	Shows a list of all the shipping carriers you have connected to this account.
09	`list_created_labels`	Lists records of labels that have been successfully purchased and generated.
10	`list_shipments`	Provides a list of your most recent shipment activities and details.
11	`track_package`	Checks the current status and location of a package using its tracking number.
12	`validate_address`	Verifies if a shipping or receiving address is correctly formatted and usable by carriers.

See It in Action

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

U Validate this address: 123 Main St, Austin, TX 78701.

I've validated the address. It's a valid residential address in Austin, Texas. Would you like to check shipping rates for a package to this location?

U List all active carrier accounts in my ShipEngine.

Retrieving carrier accounts... You have 3 active accounts: FedEx, UPS, and USPS. Would you like to see the profile metadata for any of them?

U Track the shipment with ID sh_123456789.

Checking tracking status... Shipment sh_123456789 is currently 'Out for Delivery' via FedEx. It was last scanned at the local distribution center. Should I notify you when it's delivered?

Frequently Asked Questions

01 **How does ShipEngine MCP compare shipping rates?**

It uses the `get_shipping_rates` tool to query multiple carriers simultaneously. You provide the dimensions and destination, and it returns a side-by-side comparison of costs.

02 **Can I use ShipEngine MCP to validate addresses?**

Yes, you can run `validate_address` to check if any sender or recipient address is accurate before you commit to generating a label. This prevents failed shipments.

03 **Does the ShipEngine MCP only work for US carriers?**

No, it supports dozens of global carriers through its API connections. You can compare rates and create labels across multiple countries.

04 **What is the difference between `create_shipment_object` and `get_shipping_rates`?**

`get_shipping_rates` only gives you a cost estimate using sample data. You must first use `create_shipment_object` to save a formal record of the shipment before finalizing rates or generating labels.

05 **How often should I run get_connection_status?**

You should check this whenever you suspect your API keys might have expired or if you're having trouble running any shipping tool. It confirms the health of the connection.

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 `"shipengine": { "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

ShipEngine 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 ShipEngine. 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	ShipEngine MCP
Server ID	`019dd15a-413a-7160-acb8-6e8f3fcdafad`
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/shipengine.