MCP SERVER | NO CODE | CLOUD HOSTED

Orb MCP

Automate billing and usage tracking instantly.

Orb connects usage-based billing logic to your AI agent. It lets you automate revenue operations tasks like ingesting usage events, managing customer accounts, and predicting invoices directly through natural conversation.

A+ Quality Score 100/100

usage-based-billing | revenue-operations | subscription-management | metering | invoicing | fintech

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

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.

Orb MCP

10 tools available
Cloud-hosted on Vinkius

Need to track complex usage metrics or manage subscriptions without opening a dozen dashboards? This MCP handles the heavy lifting of usage-based billing cycles. Your agent can read live data streams, check historical financial records, and update billing information on demand. You ask for it—say you need to know what next month's invoice will look like, or perhaps you want to process a batch of usage events from last night. The MCP handles the API calls, ensuring your revenue operations run smoothly.

Connecting this through Vinkius means you get access to all these financial tools in one place, no matter if you use Claude, Cursor, or another compatible agent. It’s about getting accurate billing data instantly, letting you focus on product strategy instead of spreadsheet reconciliation.

Core Capabilities

01 — Ingesting Usage Events

Send large batches of usage events to track consumption in real-time for billing purposes.

02 — Managing Customer Accounts

Create new customer profiles and retrieve detailed information or a list of all existing users, including external IDs.

03 — Handling Subscriptions

Set up, modify, cancel, or update an existing subscription tied to a specific billing plan.

04 — Predicting and Listing Invoices

Fetch draft invoices for upcoming billing cycles or retrieve a full list of past invoices.

05 — Adjusting Credits

Manage prepaid credits by creating detailed ledger entries, whether they're increases, decreases, or expirations.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Orb API Key.

02 Connect the credentials to your preferred AI client (like Cursor or Claude).

03 Instruct your agent using natural language; it will then execute billing actions against the Orb platform.

The bottom line is you talk to your agent, and the MCP runs the necessary billing commands in the background.

Built For

This MCP is for RevOps teams and Finance professionals who are tired of manually exporting usage data or cross-referencing dashboards just to check a customer's billing status. It's built for anyone whose job involves accurate, real-time financial reporting on usage.

Revenue Operations Manager

Manages the entire lifecycle of subscriptions, ensuring that every change—from a new customer signup to an account upgrade—is accurately recorded and billed.

Financial Analyst

Requires instant access to historical billing reports or needs to check upcoming draft invoices before month-end close.

Engineering Lead

Needs a quick way to test usage event ingestion or verify customer metadata without jumping into the dedicated dashboard tool.

What Changes When You Connect

01 Never manually export customer lists again. Use list_customers to get all user metadata, letting your agent find a specific ID or status for you.

02 Predict revenue accurately by calling get_upcoming_invoice. You'll know the estimated bill amount—including metered usage—before month-end close.

03 Process huge volumes of data efficiently. The ingest_events tool handles batch processing, ensuring every unit of consumption is tracked with unique idempotency keys.

04 Maintain accurate financial records by using create_ledger_entry to manage prepaid credits and adjustments without touching a spreadsheet.

05 Handle customer changes instantly. You can use create_subscription or update_subscription right from your chat interface when an account needs adjusting.

Real-World Applications

Checking Customer Credit Status

A financial analyst needs to know if a client has enough credits before enabling premium features. Instead of navigating through three tabs, they ask their agent to fetch_ledger for the account, getting an immediate yes or no.

Onboarding a New Client

The RevOps manager signs up a new user and needs them billed immediately. They prompt their agent with create_customer and then follow up to create_subscription in two steps, all without leaving the chat window.

Debugging Usage Failures

An engineering team member notices usage spikes but isn't sure if events were recorded. They use ingest_events with a specific customer ID and batch keys to verify delivery status immediately.

End-of-Cycle Billing Review

The finance department needs a summary of all invoices for the last quarter. The agent runs list_invoices, providing an immediate overview that saves hours of database querying.

Patterns to Avoid

Trying to manage billing via generic data tools

✗ AVOID

A user might try to use a general database tool to 'list all users' and manually figure out who is subscribed or what their current credit balance is.

✓ INSTEAD

Use list_customers for basic profiles, then follow up with fetch_ledger or get_upcoming_invoice to pull specific financial status data.

Manually tracking usage in spreadsheets

✗ AVOID

A team copies and pastes thousands of usage entries into Excel every week, leading to human error and massive cleanup time.

✓ INSTEAD

Send the raw batch data directly using ingest_events. The MCP handles the keying and recording process accurately for you.

Assuming subscription status is visible everywhere

✗ AVOID

A product manager checks a user's profile but doesn't see if their billing cycle has been canceled or updated.

✓ INSTEAD

Always use cancel_subscription, update_subscription, and list_customers to confirm the current state of an account.

The Right Fit

Use this MCP if your job involves tracking revenue, usage metrics, or subscription status in a way that requires interacting with structured financial data. It excels when you need to execute multi-step processes—like creating a customer, then subscribing them, and finally ingesting their first batch of usage events.

Don't use this MCP if you only need simple read access to unstructured data (e.g., retrieving a document by name). For those cases, a general knowledge retrieval tool is better. Also, don't try to perform complex accounting logic; the tools handle the *recording* of financial changes, but they aren't an accounting ledger themselves.

If your need is merely checking simple contact info, list_customers works. If you are dealing with billing cycles and money, this MCP is what you need.

Billing data used to be a mess of dashboards and exports.

Today, tracking revenue means logging into the billing dashboard, running a report on customer usage, exporting that CSV, opening it in Excel, and then manually cross-referencing those numbers against your subscription sheet. It takes hours just to get a clean picture of who owes what.

With this MCP, you ask your agent directly: 'What's the billing status for Acme Corp?' The system runs the necessary checks—from listing customer metadata to checking their usage and predicting invoices—and gives you one clear answer right where you are working. It just works.

Manage subscriptions with Orb MCP.

Previously, changing a subscription required navigating through an account portal, finding the customer record, and hitting 'update plan.' If that process failed halfway, you were stuck until someone manually fixed it.

Now, your agent can handle complex actions. You ask to update or cancel a subscription; the MCP executes those specific commands directly, confirming success every time.

The Orb MCP: 10 Tools for Billing Management

These tools allow your AI agent to handle every aspect of your billing lifecycle, from creating new users to processing massive batches of usage data.

#	TOOL	DESCRIPTION
01	`cancel_subscription`	Stops an existing subscription for a given customer.
02	`create_customer`	Adds a brand new user record to the billing system using their name and email.
03	`create_ledger_entry`	Records a financial adjustment, either adding or subtracting prepaid credits.
04	`create_subscription`	Initializes a new subscription plan for a customer account.
05	`fetch_ledger`	Retrieves the complete history of credit adjustments for a specific customer.
06	`get_upcoming_invoice`	Looks up the predicted amount and details for an invoice before it's finalized.
07	`ingest_events`	Processes batches of usage data, requiring customer ID, event name, timestamp, and a unique idempotency key.
08	`list_customers`	Pulls a complete list of all customers currently in the system.
09	`list_invoices`	Retrieves historical records of billing invoices.
10	`update_subscription`	Modifies an existing subscription's details or plan type.

See It in Action

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

U List the last 5 customers created in Orb.

I've retrieved the latest customers. The most recent entries include 'Acme Corp' (ID: cust_821), 'Global Tech' (ID: cust_932), and 'Starlight Inc' (ID: cust_104). Would you like to see the subscription details for any of them?

U Ingest a usage event for customer cust_123: 50 units of 'api_requests'.

Usage event successfully ingested for 'cust_123'. I've used a unique idempotency key for this batch to ensure accurate tracking. The 50 units of 'api_requests' are now reflected in their usage metrics.

U What is the upcoming invoice amount for subscription sub_987?

The upcoming draft invoice for subscription `sub_987` is currently at $450.00. This includes the base plan fee plus $120.00 in metered usage accrued during the current period.

Frequently Asked Questions

01 **How do I ingest usage events using Orb MCP?**

You use the ingest_events tool by providing the required customer ID, event name, timestamp, and a unique idempotency key. The agent ensures the data is recorded accurately in bulk.

02 **Can Orb MCP create new customers?**

Yes. Use the create_customer tool to add a brand new user record, specifying both their name and email address for billing purposes.

03 **How do I check if an invoice is coming up with Orb MCP?**

Call get_upcoming_invoice. This retrieves the draft total, letting you see exactly how much will be billed next without needing to wait for the official date.

04 **Does Orb MCP handle credit adjustments?**

Yes, it uses create_ledger_entry to manage prepaid credits. You can record entries for increments, decrements, or expirations easily.

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 `"orb": { "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

Orb 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 Orb. 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	Orb MCP
Server ID	`019e38d0-5939-7046-a597-dbf567206e47`
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/orb.