MCP SERVER | NO CODE | CLOUD HOSTED

Optum Claims & Billing MCP

Automate claim submission, tracking, and payment reconciliation.

Optum Claims & Billing MCP connects your AI agent directly to Optum's financial APIs. Submit professional and institutional medical bills, track their status immediately, download detailed payment remittances, and automatically resubmit corrections for denied claims. This tool handles the entire revenue cycle back-office process.

A+ Quality Score 100/100

medical-billing | edi | claims-processing | remittance | healthcare-compliance

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/optum-claims-billing

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.

Optum Claims & Billing MCP

5 tools available
Cloud-hosted on Vinkius

Processing medical payments shouldn't involve mountains of paper or endless manual reconciliation. This MCP gives your AI client direct access to Optum's core APIs, letting you take control of your billing lifecycle. You can construct and send complex claims—whether it's a doctor's office visit (professional) or a major hospital stay (institutional). Need to know if that claim went through? Check the status instantly to see if it's approved, pending review, or rejected. When payments land in the bank, you don't have to manually download statements; this MCP pulls down structured Electronic Remittance Advice records for easy reconciliation. If a payer denies a bill due to coding errors, you can even use the agent to fix and resubmit the corrected claim automatically. Vinkius hosts this critical connector so your AI client can manage all these complex financial actions without needing specialized software or manual intervention.

Core Capabilities

01 — Submit Medical Bills

Send professional (outpatient) and institutional (hospital stay) medical claims directly to the payer for payment processing.

02 — Track Claim Status

Check the real-time status of any submitted claim, knowing immediately if it's approved, denied, or still under review.

03 — Download Payment Records

Retrieve detailed Electronic Remittance Advice (ERA 835) files that break down exactly how much money was paid and why.

04 — Correct and Resubmit Claims

Automatically fix coding or administrative errors on a rejected claim and resend the corrected version to overturn the denial.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/optum-claims-billing** — connect your AI agent in three steps.

01 First, you insert your financial client ID into the runtime module. This securely authorizes your agent's ability to send or receive sensitive billing data.

02 Next, you tell your AI client what needs doing—for example, submitting an institutional claim or fetching recent remittance advice.

03 The MCP executes the API call, returns a structured result (like the final status code or a payment payload), and feeds it back to your agent for action.

The bottom line is that you get automated access to Optum's entire claims processing workflow without ever touching a manual form or waiting on batch file uploads.

Built For

Billing Managers, Revenue Cycle Analysts, and FinTech Operations teams need this. If your job involves tracking payments from insurance payers, managing claim denials, or reconciling complex financial data streams, you'll use this. It takes the guesswork out of multi-stage payment cycles.

Billing Manager

Uses this to monitor a backlog of claims by using opt_check_claim_status and coordinating resubmissions via opt_handle_denial_revision.

Revenue Cycle Analyst

Relies on the MCP to pull bulk payment data using opt_get_remittance_advice, ensuring all incoming funds are correctly posted against outstanding bills.

Healthcare IT Specialist

Integrates claim submission processes by running scripts that call both opt_submit_professional_claim and opt_submit_institutional_claim for mass processing.

What Changes When You Connect

01 Stop manually chasing payments. You can use opt_get_remittance_advice to pull detailed ERA 835 records instantly, giving you a complete financial picture of every incoming check.

02 Speed up the entire billing cycle. Instead of waiting for human review, your agent submits claims using opt_submit_professional_claim or opt_submit_institutional_claim and tracks them with opt_check_claim_status.

03 Eliminate denial bottlenecks. When a claim gets rejected due to coding errors, you don't have to start over. You just use opt_handle_denial_revision to fix the issue and resubmit it right away.

04 Gain comprehensive oversight. Your AI client can manage both the initial submission of bills and the final reconciliation of funds in one workflow, linking payments directly back to submitted claims.

05 Maintain compliance and security. This MCP is built specifically for hospital back-office finance, isolating billing data from general patient records.

Real-World Applications

Reconciling a large payment batch

A Billing Analyst needs to know what payments arrived last week. They ask their agent to use opt_get_remittance_advice, and the MCP pulls three detailed ERA 835 files, showing every dollar amount and which specific services they cover.

Following up on a denied claim

A Billing Manager finds out that a hospital stay was billed incorrectly. They instruct their agent to use opt_handle_denial_revision, fixing the coding errors in minutes and resubmitting the corrected 837I payload.

Submitting routine doctor visits

An office administrator needs to process fifty daily appointments. They ask their agent to use opt_submit_professional_claim, batch-submitting all the necessary professional bills (837P) in one secure operation.

Verifying pending claim status

A user needs to know if a major surgical bill is approved. They query their agent using opt_check_claim_status and get an immediate response: 'Approved, check date May 15th.'

Patterns to Avoid

Searching for payment records in PDF statements

✗ AVOID

Spending hours opening hundreds of PDFs from the bank or payer portal just to manually copy dollar amounts and service codes into a spreadsheet.

✓ INSTEAD

Use opt_get_remittance_advice. This tool downloads structured ERA 835 data, giving you clean, machine-readable payment breakouts immediately.

Starting the billing process from scratch

✗ AVOID

A claim was rejected last month because of a minor coding mistake, and now the user has to rebuild the entire submission payload manually.

✓ INSTEAD

Use opt_handle_denial_revision. This tool takes the original claim details, fixes the errors, and resubmits it for you, saving hours of reconstruction.

Guessing if a claim was processed

✗ AVOID

A manager wonders if the payer even saw that bill yet, forcing them to call the payer and wait on hold.

✓ INSTEAD

Use opt_check_claim_status. This gives you instant confirmation—approved, denied, or pending—so you know exactly where your time needs to go.

The Right Fit

You need this MCP if your core job revolves around the full financial lifecycle of medical claims: submission, tracking, payment, and dispute resolution. Use it when you need to move data across systems—for example, taking a claim status check (opt_check_claim_status) and then using that outcome to trigger a corrected resubmission (opt_handle_denial_revision). Don't use this if you just need to read a single PDF payment statement. For simple document retrieval or data visualization, a basic file-read tool is

enough. If your goal is only to *write* a claim payload without confirming its status or receiving payment advice, the MCP still manages that workflow but might be overkill.

The mess of manual medical billing reconciliation

Today, processing payments from insurance payers is slow and painful. You're stuck opening payer portals, downloading PDF remittance statements, logging into accounting software to cross-reference amounts, and then manually copy-pasting service codes and dollar totals into your general ledger. It’s a cycle of clicks, downloads, and inevitable data entry errors.

With this MCP, that whole process collapses. Your agent pulls structured payment records using opt_get_remittance_advice. You get clean, machine-readable payment breakouts instantly, letting you spend time analyzing cash flow instead of fighting with spreadsheets.

Getting final status and correcting denials

The biggest time sink is the denial process. When a claim doesn't pay, manually figuring out if it was a coding error or a policy issue, fixing the payload, and resubmitting it takes days of coordination across different departments.

Now, you use opt_handle_denial_revision. You fix the mistake within the agent environment, and the MCP handles the resubmission to Optum’s system automatically. The claim moves from 'Denied' directly toward 'Approved'.

Optum Claims & Billing: 5 Tools

These tools allow you to manage the entire financial lifecycle of medical billing—from initial submission through final payment reconciliation.

#	TOOL	DESCRIPTION
01	`opt_check_claim_status`	Checks the current status of a medical bill, telling you if it's pending, denied, or paid.
02	`opt_get_remittance_advice`	Downloads payment records (ERA 835) that detail exactly how much money was received and for which services.
03	`opt_handle_denial_revision`	Fixes coding errors on a rejected claim and resubmits the corrected version to overturn the denial.
04	`opt_submit_institutional_claim`	Submits large medical bills (837I) associated with hospital stays or major facility services.
05	`opt_submit_professional_claim`	Sends professional medical claims (837P) for outpatient visits and services provided by doctors.

See It in Action

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

U Check the status of the institutional claim ID 88829281X.

Query completed. The medical claim 88829281X is currently Approved and expects a check date on May 15.

U List the latest ERA remittances available for clearing.

Pulled 3 recent ERA 835s from UnitedHealth Group clearinghouse totaling $12,450.00 ready for auto-posting.

U Draft an 837 payload for an office visit code 99213.

The ANSI 837 Professional Claim string has been successfully prepared for code 99213 passing all pre-adjudication compliance checks.

Frequently Asked Questions

01 **Does Optum Claims & Billing help with both hospital and office visits?**

Yes, it handles both major facility payments (institutional) using opt_submit_institutional_claim and smaller doctor/outpatient services (professional) via opt_submit_professional_claim.

02 **How can I check the status of a bill submitted through Optum Claims & Billing?**

You use the opt_check_claim_status tool. This gives you an immediate, definitive answer on whether your medical claim is approved, denied, or still pending review.

03 **What kind of data does opt_get_remittance_advice provide?**

This tool downloads the ERA 835. It provides a detailed breakdown of payments, showing exactly which payer paid how much and for what specific services rendered.

04 **Is this MCP secure enough for handling financial data?**

Yes. The architecture is engineered specifically to handle back-office finances, ensuring high security and compliance by keeping billing functions separate from general patient records.

05 **Can I use Optum Claims & Billing to fix a rejected claim?**

Absolutely. You use opt_handle_denial_revision. This process allows you to correct coding errors and resubmit the necessary documentation to overturn a payer denial.

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 `"optum-claims-billing": { "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

Optum Claims & Billing 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 Optum Claims & Billing. 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	Optum Claims & Billing MCP
Server ID	`019d75eb-4c85-7048-8ac4-fc00e3bfb3d2`
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/optum-claims-billing.