MCP SERVER | NO CODE | CLOUD HOSTED

Revenue Recognition MCP

Automate GAAP/IFRS 15 Accounting Calculations

US GAAP & IFRS 15 Revenue Recognition automatically implements the complex five-step accounting model for finance professionals. It helps your agent analyze contracts, figure out what revenue streams are owed, distribute total contract prices across obligations, and calculate exactly how much earned versus deferred revenue you should report.

A+ Quality Score 100/100

us-gaap | ifrs-15 | revenue-recognition | asc-606 | accounting-automation

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/us-gaap-ifrs-15-revenue-recognition

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.

US GAAP & IFRS 15 Revenue Recognition MCP

3 tools available
Cloud-hosted on Vinkius

Calculating revenue under ASC 606 or IFRS 15 isn't simple arithmetic; it involves deeply analyzing the underlying promises made in a client contract. This MCP takes that complexity away from your team. You can use the `identify_performance_obligations` tool to analyze raw contract items and determine how many separate services or goods are actually being sold. Next, you run the `allocate_transaction_price` function to distribute the total deal value across those obligations based on market rates. Finally, by using `calculate_revenue_recognition`, your agent determines what percentage of revenue is earned out over time versus what needs to be deferred until later. When you connect this specialized engine via Vinkius, it means your AI client has access to a best-in-class accounting tool that handles the hardest parts of modern financial reporting.

Core Capabilities

01 — Determine Contract Obligations

The MCP analyzes contract details to identify every distinct service or good promised, establishing clear performance obligations.

02 — Distribute Total Deal Value

It calculates how a single, large payment amount should be proportionally spread across multiple separate revenue streams.

03 — Calculate Earned Revenue vs. Deferred Funds

The system determines the exact dollar amounts of revenue that are earned based on progress and those that must be held back until performance is complete.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/us-gaap-ifrs-15-revenue-recognition** — connect your AI agent in three steps.

01 Feed the MCP a client contract or list of items needing accounting treatment.

02 The system first identifies all distinct, separate obligations within the contract using one tool. Then, it allocates the total price across these obligations and finally computes the recognized revenue based on completion progress.

The bottom line is you get an automated, auditable calculation that follows international accounting standards for revenue reporting.

Built For

Anyone dealing with complex contracts—from corporate finance staff to specialized public accountants. If your job involves translating legal service agreements into GAAP-compliant financial entries, this MCP saves hours of manual calculation and auditing risk.

Financial Analyst

They use this MCP to model various contract scenarios, simulating how different performance obligations affect reported revenue before the quarter closes.

Controller / Accounting Manager

They rely on it to automate routine monthly revenue journal entries, ensuring every transaction adheres strictly to IFRS 15 rules without manual cross-checking.

Auditor

They use the MCP's detailed calculations to quickly validate a company's reported deferred and recognized revenue balances against source contracts.

What Changes When You Connect

01 Instead of manually reviewing contracts to separate services, use `identify_performance_obligations` to automatically build a definitive list of every promise made under the contract.

02 You don't have to guess how to split a payment. The `allocate_transaction_price` tool ensures the total revenue amount is correctly distributed across all identified obligations based on relative market value.

03 Stop worrying about timing differences. Running `calculate_revenue_recognition` tells you exactly what percentage of revenue belongs in this month versus next quarter's books, saving crucial compliance time.

04 The MCP enforces complex accounting standards (ASC 606 and IFRS 15) that are notoriously hard to interpret consistently across departments or business units.

05 It provides a single point of truth for revenue reporting. Your AI client performs the entire multi-step calculation chain, from obligation identification right through to final journal entry amounts.

Real-World Applications

Structuring a Multi-Part Software Deal

A SaaS company signs a contract covering software access, implementation support, and future maintenance. The agent first uses `identify_performance_obligations` to separate these three items. Then, it calls `allocate_transaction_price` to determine the fair value split for each service before finally running `calculate_revenue_recognition` to ensure the right revenue is booked monthly.

Analyzing a Consulting Agreement

The firm has a large retainer fee but performance is milestone-based. The agent uses `identify_performance_obligations` to confirm the primary service deliverable, then uses `calculate_revenue_recognition` to ensure payment only hits revenue when specific milestones are verifiable.

Handling Bundled Services

A client buys a bundle of product licenses and premium support. The analyst feeds the contract data into the MCP, which first uses `identify_performance_obligations` to separate the license from the support. It then calls `allocate_transaction_price` to split the cost fairly before calculating revenue.

Year-End Revenue Review

The controller needs to verify all remaining deferred revenue accounts. They use the MCP's tools in sequence, starting with `identify_performance_obligations` on historical contracts, and ending with `calculate_revenue_recognition` to confirm balances for the audit.

Patterns to Avoid

Treating all revenue as earned immediately

✗ AVOID

Manually booking 100% of a contract's value upon signing because it seems like 'money in the door.' This violates GAAP/IFRS requirements for deferred income.

✓ INSTEAD

Don't assume. Run `identify_performance_obligations` first to break down the promises, and then use `calculate_revenue_recognition` to ensure only earned amounts are booked immediately.

Ignoring relative standalone selling prices

✗ AVOID

Simply splitting a contract price 50/50 between two services because they were listed together. This misrepresents the actual economic value of each service.

✓ INSTEAD

Use `allocate_transaction_price`. This tool correctly assigns values based on what those services would sell for independently, ensuring accuracy.

Overlooking distinct promises

✗ AVOID

Lumping a one-time setup fee and ongoing support into a single service line in the accounting ledger.

✓ INSTEAD

Always start by running `identify_performance_obligations`. This forces you to see if the contract contains multiple, measurable promises that need separate tracking.

The Right Fit

Use this MCP if your core problem is translating complex, multi-item client contracts into compliant financial journal entries under ASC 606 or IFRS 15. You need to know *how* the total price must be split and *when* each portion of revenue can legally be recognized.

Don't use this if you only need basic arithmetic (e.g., calculating simple interest) or if all your contracts are single-item, fully realized services. If your process is already standardized and non-negotiable by GAAP/IFRS standards, a simpler calculation tool might suffice. But if the contract structure itself is the source of the difficulty—the mix of licenses, support, and professional fees—this MCP is necessary.

The headache of revenue accounting starts with manual review.

Today, finance teams spend days piecing together contracts. They have to read dozens of pages, cross-reference service lists, and manually ask: 'Is this one distinct promise or part of a bigger package?' Then they attempt to split the total price using internal spreadsheets, hoping their manual allocation matches what GAAP requires.

With this MCP, your agent handles that initial deep dive. You give it the contract, and it uses `identify_performance_obligations` to break down the legal language into clear, actionable components. The result is a clean list of promises ready for precise financial modeling.

Using allocate_transaction_price gives you guaranteed accuracy.

The biggest manual risk is pricing. Spreadsheets rarely account for the relative standalone selling price (SSP) of services versus physical goods, leading to misstated revenue allocations that auditors notice immediately. You're forced into guesswork when splitting the total contract value.

By using `allocate_transaction_price`, you eliminate the guess work entirely. The MCP calculates the correct proportional split based on market data, delivering an allocation that is both accurate and defensible.

US GAAP & IFRS 15 Revenue Recognition MCP with 3 Tools

These tools allow your agent to manage the full lifecycle of compliant revenue accounting, from initial contract analysis to final journal entry calculations.

#	TOOL	DESCRIPTION
01	`allocate_transaction_price`	Distributes the total amount paid in a contract across all the separate goods or services provided.
02	`calculate_revenue_recognition`	Determines which portion of revenue has been earned and which portion must be kept as deferred until future milestones are hit.
03	`identify_performance_obligations`	Analyzes contract wording to pinpoint every distinct, measurable promise the company makes to the client.

See It in Action

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

U I have a contract with two items: 'Software License' and 'Support Services'. How many performance obligations are there?

The `identify_performance_obligations` tool identifies 2 distinct performance obligations: Software License and Support Services.

U Allocate a $10,000 total price between 'License' ($8,000 SSP) and 'Support' ($2,000 SSP).

The allocated amounts are: License: $8,000 and Support: $2,000.

U If I have an allocated amount of $5,000 and I am 40% complete, how much revenue is recognized?

$2,000 has been recognized as revenue, and $3,000 remains as deferred revenue.

Frequently Asked Questions

01 **How does US GAAP & IFRS 15 Revenue Recognition help with complex contracts?**

It automates the five-step model required by accounting standards. It guides your agent through identifying obligations, allocating prices, and calculating the earned revenue amount for any contract structure.

02 **Do I need to use all three tools in order with US GAAP & IFRS 15 Revenue Recognition?**

Yes, they work as a chain. You should run `identify_performance_obligations` first; that output feeds into `allocate_transaction_price`, which then provides the necessary input for `calculate_revenue_recognition`.

03 What happens if my contract has variable consideration?

The MCP can process these scenarios. While complex, you use the tools to model how the total transaction price changes based on performance triggers or bonuses defined in the agreement.

04 Can I use US GAAP & IFRS 15 Revenue Recognition for non-accounting tasks?

No. This MCP is highly specialized for finance and accounting rules (GAAP/IFRS). It cannot process general business logic or operational data outside of contract financial terms.

05 What does the `calculate_revenue_recognition` tool need to run?

It needs two main inputs: a total allocated amount and a measure of completion progress (e.g., 40% complete). It then outputs the precise recognized and deferred revenue amounts.

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 `"us-gaap-ifrs-15-revenue-recognition": { "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

US GAAP & IFRS 15 Revenue Recognition 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 US GAAP & IFRS 15 Revenue Recognition. 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	US GAAP & IFRS 15 Revenue Recognition MCP
Server ID	`019ee68f-26a3-7388-acd1-d053239d7767`
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/us-gaap-ifrs-15-revenue-recognition.