

Insights

# **Built on Borrowed Intelligence:** Accounting & Reporting Challenges for the AI Application Layer



# Executive Summary

A new category of software company has emerged—one that builds its products not from raw code alone, but on top of foundation models created by others. These AI application layer companies span search and knowledge engines, enterprise AI middleware platforms, coding assistants, design tools, and autonomous agent frameworks. They represent some of the fastest-growing software businesses in history: multiple companies in this category have reached \$100 million in annual recurring revenue within 12–18 months of launch, growth trajectories that took traditional SaaS companies five to seven years to achieve.

Yet beneath the remarkable growth lies a set of accounting and reporting challenges that are genuinely novel. The economics of these businesses look fundamentally different from traditional SaaS: gross margins of 25–60% versus the 80–90% that software investors have come to expect, variable cost structures where every customer interaction triggers a real, measurable cost, and revenue models that blend subscriptions with consumption in ways that stress-test ASC 606's principal-agent framework.

The central question—whether these companies are principals selling an integrated solution or agents facilitating access to someone else's intelligence—has cascading implications across the P&L, from revenue presentation to cost of revenue classification to R&D capitalization. And the answer is rarely binary.

This whitepaper provides a practitioner's guide to the accounting and reporting challenges unique to the AI application layer. It is anchored in US GAAP with comparative notes on IFRS and Ind AS. Where relevant, we highlight SEC staff comment letter themes and disclosure expectations, recognizing that many companies in this category are approaching public markets. Since 2021, the SEC has issued several AI-related disclosure comments to different companies across industries, underscoring the regulatory spotlight on this space. The whitepaper is designed for CFOs, controllers, and audit committees at AI-native companies, as well as the investors, board members, and auditors evaluating them.

## Unique Perspective

The AI application layer has created a new software archetype whose accounting looks more like a marketplace or distribution business than traditional SaaS. The existing standards framework—ASC 606 (Revenue from Contracts with Customers), ASC 350-40 (Internal-Use Software)—provide tools to address these questions, but the judgments required are at the frontier of practice. Most companies in this category are making these calls with limited technical accounting resources, creating material misstatement risk that compounds as they approach public markets.

## ACKNOWLEDGEMENTS

We sincerely thank Shibu Ninan for his valuable inputs and inside-out perspective on the AI application layer, that strengthened several key areas of this publication.



**Sandip Khetan**

*Co-Founder, Global Head of Accounting & Reporting Consulting*



**Avinash Ramkumar**

*Partner, Accounting & Reporting Consulting*



**Shival Sehgal**

*Managing Director, Accounting & Reporting Consulting*



**Shibu Ninan**

*Senior Vice President, Chief Accounting Officer, at Okta, Inc.*

## Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

# 1 The AI Application Layer: A New Software Taxonomy

To analyze the accounting challenges systematically, it is helpful to define three distinct archetypes within the AI application layer. Each carries a different accounting profile, even though they share the common characteristic of building on third-party foundation models.

Executive Summary

**The AI Application Layer: A New Software Taxonomy**

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

## 1.1

### Archetype A: AI-Augmented Search and Knowledge Platforms

*Examples: Perplexity (AI-native search) and similar AI-powered answer engines*

These companies provide proprietary search, summarization, or question-answering engines powered by one or more foundation models. The user submits a query; the platform retrieves relevant information from the web or proprietary sources, invokes one or more foundation models for synthesis, and delivers a curated answer—often with citations, follow-up suggestions, and a conversational interface.

Revenue is typically subscription-based (consumer or enterprise tiers), sometimes with usage caps or overage charges. The platform's value proposition is the quality of the retrieval, synthesis, and presentation layer—not the raw model capability. The foundation model is an ingredient, not the product.

## 1.2

### Archetype B: Enterprise AI Middleware and Knowledge Orchestration

*Examples: Glean (enterprise AI search and knowledge), Moveworks (enterprise AI platform)*

These companies sit between an enterprise's proprietary data ecosystem and one or more foundation models, providing connectors, retrieval-augmented generation (RAG) pipelines, permissions and access control layers, semantic search, and conversational interfaces over internal knowledge. They are the “glue” that makes foundation models useful within the enterprise.

An emerging pattern is “BYOD-AI” (Bring Your Own Data — AI), where the customer provides proprietary data and the vendor provides the orchestration, model access, and interface layers. The customer's data is the primary input; the vendor's platform transforms it into AI-powered outputs. Under ASC 606-10-32-24, the



entity assesses whether it obtains control of the customer’s contributed data — in most BYOD-AI arrangements it does not, as the data remains the customer’s asset subject to contractual return or deletion obligations. Whether the promise is to process the customer’s data or to provide continuous platform access affects recognition timing **(see Section 5.1).**

Revenue is enterprise subscription, often tiered by number of seats, data volume, or number of connected data sources. Contracts are typically annual or multi-year. The foundation model is embedded in the workflow but the customer is paying for the integration, security, and orchestration layer.

1.3

Archetype C: AI-Native Development and Productivity Tools

Examples: Cursor (AI Coding assistant), Jasper (AI content platform)

These companies build coding assistants, design automation tools, writing aids, or workflow automation platforms on top of foundation models. The user interacts with the application, which makes API (software interface) calls to one or more models behind the scenes.

A rapidly emerging sub-category is agentic AI platforms — systems that can autonomously plan multi-step workflows, invoke external tools, and execute actions to achieve a defined goal without continuous human direction. These platforms introduce additional accounting complexity because the cost per customer outcome is unpredictable — each autonomous task may require a variable number of foundation model calls, tool invocations, and reasoning steps, creating cost variability that traditional per-token pricing models do not capture.

Revenue models vary widely—from pure subscription to consumption-based (priced per token, generation, or action) to hybrid subscription-plus-usage. Annual Recurring Revenue (ARR) is the standard growth metric in this category. This archetype has seen the most explosive growth, with several coding and design tools reaching \$100–200 million in ARR within 12–18 months. It also presents the most complex accounting questions, particularly around gross-vs-net determination and the classification of foundation model API costs.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

## Executive Summary

### The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

| Dimension                        | Archetype A: AI Search                                                                                                  | Archetype B: Enterprise AI                                                                                                           | Archetype C: Dev/ Productivity                                                                                                                                                                                     |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Revenue Model                    | Subscription ± usage caps                                                                                               | Enterprise subscription (seats/data)                                                                                                 | Hybrid: subscription + token consumption                                                                                                                                                                           |
| Gross Margin                     | 40–60%                                                                                                                  | 60–75%                                                                                                                               | 25–65%                                                                                                                                                                                                             |
| API Cost as % Revenue            | 30–50%                                                                                                                  | 15–30%                                                                                                                               | 35–70%                                                                                                                                                                                                             |
| Performance Obligation Structure | Single performance obligations — transformative relationship between proprietary retrieval/ ranking and model inference | Single performance obligations components highly interdependent (connectors, RAG, permissions, AI layer inseparable)                 | Most fact-dependent — <b>transformative</b> (single performance obligations) when deeply integrated; <b>additive</b> (multiple performance obligations) when on-device and cloud components function independently |
| Primary ASC 606 Question         | Principal vs. agent — supportable but warrants documentation given high API cost ratio                                  | Variable consideration — enterprise contracts with consumption-based components, tiered pricing, and seat true-up                    | Distinctiveness (transformative vs. additive) + Principal vs. agent + variable consideration                                                                                                                       |
| Gross vs. Net Risk               | Medium - strong proprietary transformation but significant API cost exposure                                            | Lower - deep proprietary orchestration layer typically supports principal                                                            | High - wide variation in degree of proprietary transformation across products                                                                                                                                      |
| Nature of Promise                | Stand ready (continuous access to AI search) ; may shift toward specified-quantity where usage caps limit access        | Stand ready (continuous access to enterprise platform) ; seat or data volume limits may introduce specified-quantity characteristics | Typically stand-ready for subscription component; may be specified-quantity for pure consumption products                                                                                                          |



Why are these the primary ASC 606 questions by Archetype:

### Archetype A:

The company delivers a fully integrated output — a synthesized search result, summary, or answer — built from foundation model inference combined with proprietary retrieval, ranking, and citation logic. The transformative relationship between the proprietary layer and the foundation model typically results in a single performance obligation (see [Section 1.4](#)). The primary judgment is whether the company controls the integrated output before transfer — a principal-vs-agent determination. While the principal conclusion is generally supportable given the depth of proprietary transformation, the relatively high API cost as a percentage of revenue (30–50%) means the economics warrant careful analysis and documentation.

### Archetype B:

The platform integrates connectors, RAG pipelines, model access, permissions, and security layers into a unified enterprise service. These components are highly interdependent — the connectors have no value without the AI layer, and the AI layer has no data to operate on without the connectors — resulting in a transformative relationship and typically a single performance obligation (see [Section 1.4](#)). The principal-vs-agent determination applies but generally resolves more clearly in favor of principal treatment given the significant proprietary orchestration layer. The more practice-relevant ASC 606 judgment for this archetype is variable consideration — enterprise contracts frequently include consumption-based components, tiered pricing, and seat-based true-ups alongside the base subscription, requiring estimation and constraint analysis.

BYOD-AI contracts may also include data residency, encryption, and deletion commitments, which are generally administrative obligations that do not rise to separate performance obligations absent significant incremental service delivery.

### Archetype C:

The hybrid subscription + consumption pricing model creates layered challenges: (1) the distinctiveness question — whether the proprietary transformation layer and foundation model access form a single or multiple performance obligations (see [Section 1.4](#)), which varies more widely across products in this category than in A or B — a deeply integrated AI coding assistant may reflect a transformative relationship (single obligation), while a product bundling an on-device component with cloud model access may reflect an additive relationship with distinct obligations carrying different recognition patterns; (2) gross-vs-net presentation, because the degree of proprietary transformation over raw model output varies by product and may yield different conclusions for different obligations; and (3) variable consideration estimation, because the consumption-based component introduces pricing variability that must be estimated and constrained.

Across all three archetypes, the nature of the promise is typically a stand-ready obligation — the AI platform is continuously available to process customer requests on demand rather than delivering a specified quantity of outputs. However, usage caps (Archetype A), seat or data volume limits (Archetype B), and per-token or per-action pricing (Archetype C) may introduce specified-quantity characteristics that affect the measure of progress and the applicability of series guidance. This determination is discussed in [Section 5.1](#).

Executive Summary

#### The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

## Performance Obligations and Distinctiveness: The Prerequisite Question

Under ASC 606-10-25-19 through 25-22, a promised good or service is distinct if both conditions are met: (a) the customer can benefit from the good or service either on its own or together with other readily available resources (capable of being distinct), and (b) the entity's promise to transfer the good or service is separately identifiable from other promises in the contract (distinct within the context of the contract).

For most AI application arrangements, condition (a) is met for the model access component — foundation models are commercially available through direct API access, so the customer could theoretically benefit from the model independently. The determinative question is therefore condition (b): whether the model access is separately identifiable from the application company's proprietary layer within the context of the contract. ASC 606-10-25-21 provides three factors for evaluating whether promises are separately identifiable or should be combined into a single performance obligation:

### Significant integration service

**Does the entity provide a significant service of integrating the goods or services with other goods or services promised in the contract into a bundle of goods or services that represent the combined output the customer has contracted to receive?**

For AI application companies, the proprietary orchestration layer — retrieval pipelines, prompt engineering, quality filters, model routing, and domain logic — often transforms the foundation model's raw inference into a fundamentally different output. The combined output (e.g., a curated search result, a permission-aware enterprise answer, or a validated code suggestion) is qualitatively different from what either the model or the orchestration layer delivers independently. Where the application company's layer is primarily a user interface or workflow convenience over the model's unmodified output — without meaningful retrieval, transformation, or quality filtering — the integration service argument weakens.

### Modification or customization

**Does one or more of the goods or services significantly modify or customize another good or service promised in the contract?**

Fine-tuning, prompt engineering, and domain-specific training data reshape the foundation model's output in ways that are specific to the application company's product. The model's generic capability is modified to serve the application's particular use case. Where the application company passes through the model's output with only cosmetic formatting or minor prompt wrapping, modification is limited and this factor is less supportive of combination.

### Highly interdependent or interrelated:

**Are the goods or services highly interdependent or highly interrelated — that is, each is significantly affected by one or more of the other goods or services in the contract?**

In AI application companies, model upgrades by the foundation model provider require prompt recalibration and filter updates by the application company; changes to the orchestration logic alter which models are selected and how outputs are processed. Neither component delivers its intended value without the other. Where the model and the application layer operate independently — each unaffected by changes to the other — this factor supports separate performance obligations.

Executive Summary

**The AI Application Layer: A New Software Taxonomy**

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

## The Transformative vs. Additive Framework

While the FASB has not issued AI-specific guidance on this question, the framework developed for hybrid cloud and software arrangements provides the most relevant analytical lens. ASU 2016-10 (Basis for Conclusions paragraphs BC29, BC32, and BC33(b)) distinguish between two types of relationships between components in a bundled arrangement:

### Transformative relationship

The application company's proprietary layer and the foundation model are so deeply intertwined that the combined output represents something fundamentally different from either component standing alone. The customer cannot meaningfully separate the value of the model inference from the value of the orchestration, retrieval, and transformation layers. This supports a **single performance obligation** conclusion.

### Additive relationship

The application layer and the foundation model each deliver value that is not significantly affected by the other. The application adds incremental features (UI, convenience, workflow integration) but does not fundamentally transform the model's core output, and removing the application layer would leave the model's output substantially intact. This may support **multiple performance obligations** — with potentially different gross-vs-net conclusions and recognition patterns for each.

## Applying Across Archetypes



### Archetype A (AI Search)

Typically transformative. The search platform's retrieval, synthesis, ranking, and citation layers combine with the foundation model to produce an output (a curated, sourced answer) that neither component delivers independently. This generally supports a single performance obligation conclusion.



### Archetype B (Enterprise Middleware)

Typically transformative. The platform's connectors, RAG pipelines, permission controls, and enterprise graph are deeply integrated with the model's inference — the customer is purchasing the integrated enterprise knowledge experience, not model access plus bolt-on features. Neither the connectors nor the AI layer delivers its intended value without the other. This generally supports a single performance obligation conclusion, though arrangements where model access is separately contracted or priced may warrant closer evaluation.



### Archetype C (Development Tools)

Most fact-dependent. A coding assistant that deeply integrates codebase context analysis, multi-model routing, and proprietary quality filters with the model's inference (e.g., Cursor) typically delivers a single transformative output. However, arrangements where an AI component is installed on a customer's device and connects to a cloud-hosted LLM as a clearly separable add-on may exhibit an additive relationship — particularly if the on-device component functions independently and the customer independently selected and contracted for the LLM. In such cases, the on-device component may constitute a distinct license of functional IP, while the cloud-based model access is a separate service obligation — stand-ready or specified-quantity depending on whether usage caps or consumption limits constrain access (see Section 5.1).

Executive Summary

#### The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

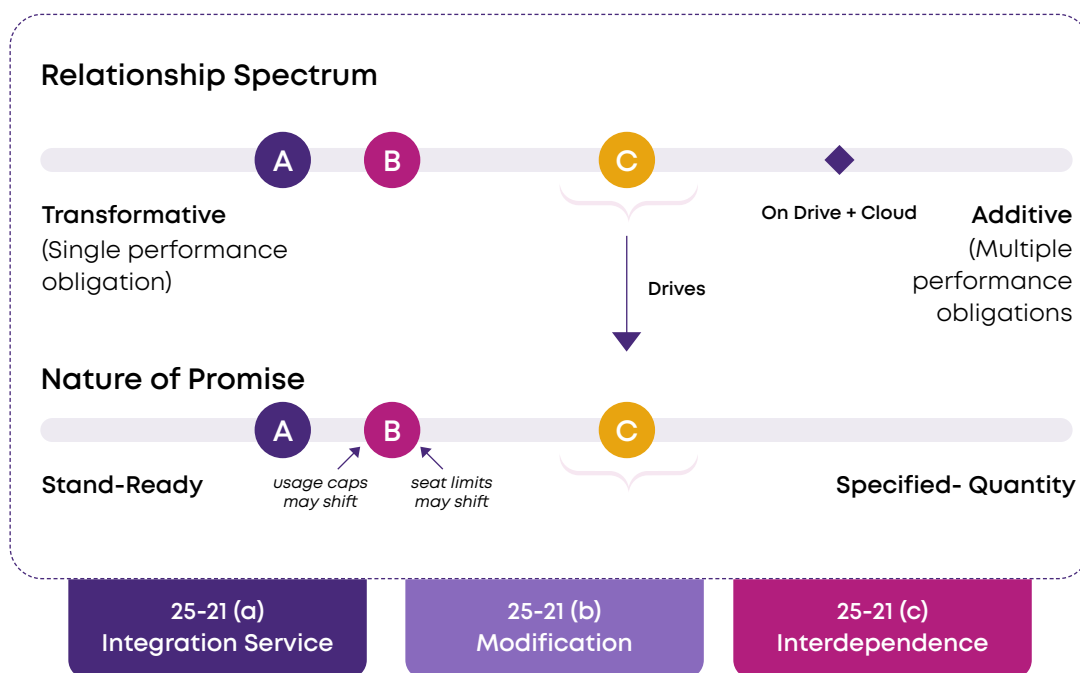
Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

# Performance Obligations and Decision Framework

A = AI Search      B = Enterprise Middleware      C = Dev and Productivity Tools



Note: This visual is for illustrative purposes only. Actual conclusions depend on the specific facts and circumstances of each arrangement.

Figure 1: Performance Obligation Decision Framework — Transformative vs. Additive Relationships

## The Cascade Effect

The distinctiveness conclusion cascades through the entire revenue recognition analysis. If the arrangement contains a single performance obligation, the gross-vs-net determination applies to the combined output — and for most AI application companies, the principal conclusion is well-supported (see Section 2.1). If multiple performance obligations exist, each must be separately evaluated for gross-vs-net, the transaction price must be allocated across performance obligations based on standalone selling prices, and recognition timing may differ — the platform obligation may be recognized ratably over the service period while the model access component follows a different pattern. The following section discusses these implications. Additionally, the nature of the promise for each identified performance obligation — stand-ready, specified-quantity, or a hybrid of both — must be determined, as this affects the measure of progress, applicability of series guidance, and the availability of practical expedients (see Section 5.1).

These judgments are inherently subjective today given the novelty of AI application business models and the absence of AI-specific FASB guidance. However, as the industry matures and pricing models, contractual structures, and delivery mechanisms become more standardized, the application of the existing distinctiveness framework is expected to become more objective and predictable — a natural evolution that has occurred with prior technology categories including cloud computing and SaaS.

Executive Summary

### The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

# 2 Revenue Recognition: Key ASC 606 Judgments

The most consequential accounting judgment for AI application layer companies is whether they are acting as a principal (reporting revenue gross) or an agent (reporting revenue net of foundation model costs) under ASC 606-10-55-36 through 55-40. The difference can be 40–60% of reported revenue—a variance that fundamentally alters growth metrics, valuation multiples, and investor perception.

Executive Summary

The AI Application Layer: A  
New Software Taxonomy

**Revenue Recognition: Key  
ASC 606 Judgments**

Cost of Revenue: The New  
Economics of COGS

R&D, Capitalization, and the  
Build vs. Buy Continuum

Subscription vs. Consumption:  
Revenue Model Complexity

Licensing, Intellectual Property,  
and Data Economics

Financial Statement  
Presentation and Disclosure

Internal Controls, SOX Readiness,  
and Audit Considerations

IFRS and Ind AS: Comparative  
Notes

How Uniquis Can Help

## 2.0

### Establishing the Framework

#### Prerequisite: Identifying the Specified Good or Service

Before applying the principal-agent indicators below, the entity must first identify the specified good or service — the promise being evaluated for gross-vs-net treatment. This requires completing the distinctiveness analysis discussed in [Section 1.4](#). The conclusions from that analysis determine the scope of the gross-vs-net evaluation:

- If the arrangement contains a **single performance obligation** (the typical conclusion for Archetypes A and B, where the relationship between the proprietary layer and the foundation model is transformative), the gross-vs-net determination applies to the combined integrated output. [Sections 2.1 and 2.2](#) address this scenario.
- If the arrangement contains **multiple performance obligations** (possible in Archetype C arrangements with additive relationships between components), each obligation must be evaluated separately — potentially reaching different gross-vs-net conclusions for different components. [Section 2.3 and 2.4](#) addresses these implications.

#### The Standard's Requirements

Under ASC 606-10-55-36, when another party is involved in providing goods or services to a customer, the entity must determine whether the nature of its promise is a performance obligation to provide the specified goods or services itself (i.e., the entity is a principal) or to arrange for those goods or services to be provided by the other party (i.e., the entity is an agent).

The core principle, per ASC 606-10-55-37A, is that an entity is a principal if it controls the specified good or service before that good or service is transferred to the customer. Control, consistent with ASC 606-10-25-25, refers to the ability to direct the use of, and obtain substantially all the remaining benefits from, the good or service—and the ability to prevent other entities from directing the use of, and obtaining the benefits from, the good or service.

Importantly, ASC 606-10-55-37A clarifies that an entity that is a principal obtains control of, among other things, a right to a service to be performed by the other party, which gives the entity the ability to direct that party to provide the service to the customer on the entity's behalf (ASC 606-10-55-37A(b)). For AI application



companies selling subscriptions that convey the right to receive AI-generated outputs on demand, the control analysis may focus on whether the company controls that right before it is transferred to the customer—which may be easier to establish than demonstrating control over each individual output itself. This framing is particularly relevant for stand-ready subscription arrangements where the company’s obligation is to provide continuous access to the AI service.

This is not merely a theoretical exercise—principal-vs-agent is consistently among the top five SEC comment letter topics under ASC 606 and has been the subject of OCA staff speeches at the AICPA Conference on Current SEC and PCAOB Developments every year since the standard’s adoption.

As **Barry Kanczucker**, then **Associate Chief Accountant** in the SEC’s Office of the Chief Accountant, observed at the 2017 AICPA Conference:

*Barry Kanczucker, OCA Associate Chief Accountant, Remarks before the 2017 AICPA Conference on Current SEC and PCAOB Developments (December 4, 2017)*

.....applying this guidance can be challenging in some fact patterns. I believe that some of the challenges are amplified in certain industries, such as the digital advertising industry or other industries in the technology space, where there are often multiple parties involved in providing the goods or services, and transactions often take place within the blink of an eye.....

*Sheri L. York, OCA Professional Accounting Fellow (December 10, 2018)*

.....significant judgment does not mean optionality. Registrants should apply a rigorous analysis of the facts in order to faithfully apply the principal versus agent model to their specific facts and circumstances.....

The SEC staff has routinely asked registrants to identify (a) who the **customer** is, (b) what the **specified good or service** is, and (c) how the entity determined it **controls** that good or service before transfer.

Identifying the Parties and the Specified Good or Service. In the AI application layer, **three** parties are typically involved:

| Party                                                       | Role                                                                                                                                                                        |
|-------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Foundation model provider (e.g., OpenAI, Anthropic, Google) | Supplies the underlying large language model via API or license. Acts as a vendor/supplier to the AI application company.                                                   |
| AI application company (the reporting entity)               | Builds a product or service layer on top of the foundation model—adding orchestration, retrieval, transformation, UI, and domain logic—and contracts with the end customer. |
| Customer (enterprise or consumer end user)                  | Purchases the AI application company’s product or service. The customer’s contract is with the AI application company, not with the foundation model provider.              |

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniqus Can Help

The specified good or service promised to the customer is the critical starting point. It is not raw foundation model inference—the customer is not purchasing API tokens from a model provider. Rather, the specified good or service is the integrated output delivered by the AI application company: the search result, the enterprise knowledge answer, the code suggestion, or the automated workflow. In most cases, the customer cannot replicate this output by calling the foundation model API directly, because the application company’s proprietary orchestration, data retrieval, context injection, and quality layers are integral to the deliverable.

This framing matters because the principal-agent analysis under ASC 606-10-55-37A asks whether the AI application company controls the specified good or service (the integrated output) before it is transferred to the customer—not whether the company controls the foundation model itself.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

**Illustrative Fact Pattern (referenced throughout this Section):**

AI AppCo is a coding productivity tool (Archetype C) that provides enterprise customers with an AI-powered code assistant. The customer subscribes at \$20/seat/month. When a developer writes code, the tool: (1) analyzes the codebase context, (2) selects the optimal foundation model from among multiple providers based on task complexity, (3) calls the selected model’s API, (4) applies proprietary quality filters and code validation, and (5) presents the refined suggestion. The customer’s contract is exclusively with AI AppCo; the customer has no direct relationship with the foundation model providers. AI AppCo sets its own pricing independent of the providers’ per-token rates. AI AppCo bears the risk that per-token costs may increase while its subscription pricing remains fixed.

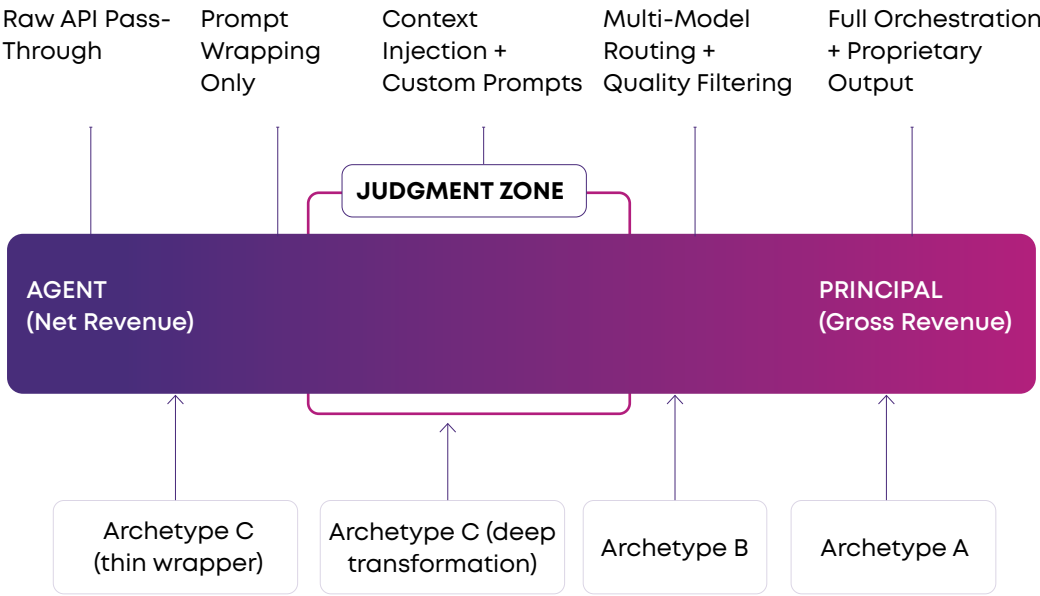


Figure 2: Principal-Agent Determination Spectrum. As the degree of proprietary transformation increases—from raw API pass-through (agent) to full orchestration with proprietary output (principal)—the gross-vs-net conclusion shifts. The “Judgment Zone” in the center reflects where the analysis is most fact-dependent under ASC 606-10-55-36 through 55-40.

## 2.1

# The Three-Indicator Framework Applied

ASC 606-10-55-39 provides three indicators (not requirements) to help determine whether an entity controls the specified good or service before transfer. Per ASC 606-10-55-39A, the relevance and weight of each indicator depends on the nature of the specified good or service and the terms of the contract. **No single indicator is determinative:**

### 1 Indicator Primary Responsibility for Fulfillment

Executive Summary

The AI Application Layer: A  
New Software Taxonomy

Revenue Recognition: Key  
ASC 606 Judgments

Cost of Revenue: The New  
Economics of COGS

R&D, Capitalization, and the  
Build vs. Buy Continuum

Subscription vs. Consumption:  
Revenue Model Complexity

Licensing, Intellectual Property,  
and Data Economics

Financial Statement  
Presentation and Disclosure

Internal Controls, SOX Readiness,  
and Audit Considerations

IFRS and Ind AS: Comparative  
Notes

How Uniquis Can Help

This indicator examines whether the AI application company is the party that the customer looks to for fulfillment of the promise. The assessment focuses on the customer's perspective: when the customer experiences a service failure or quality issue, who does the customer hold accountable?

Across all three archetypes, the AI application company is typically the party that contracts with, invoices, and provides customer support to the end customer. The foundation model provider generally has no direct relationship with the customer—in many cases, the customer may not even know which foundation model powers the service. This is typically a strong indicator of principal status. However, the strength of this indicator varies with the degree of proprietary transformation:

**For Archetype A** (AI search platforms), the company curates the query, selects which model or models to invoke, applies its own retrieval and ranking algorithms, synthesizes information from multiple sources, and delivers a proprietary output that the user could not have obtained by calling the foundation model API directly. The company is producing an integrated, differentiated output. This typically represents a strong case for **principal** treatment.

**For Archetype B** (enterprise AI middleware), the Company integrates foundation model capabilities with the customer's proprietary data ecosystem, providing connectors, RAG pipeline, security layers, and permissions management. The customer's contract is with the middleware provider, which bears responsibility for the integrated platform's performance. Primary responsibility for fulfillment typically lies with the application company.

**For Archetype C** (coding and productivity tools), the analysis is more nuanced and fact-dependent. If the tool primarily passes the user's prompt to a foundation model API and returns the raw output with minimal transformation, the company may not be primary responsible for the substance of the output, and the agent argument strengthens. However, most commercially successful tools in this category add substantial proprietary value: context injection from the customer's codebase or project, multi-model routing based on task complexity, code validation and testing, quality filtering, and iterative refinement. The greater the degree of transformation between the raw model output and the delivered result, the stronger the argument that the company is primarily responsible for fulfillment.

#### Applying to the illustrative fact pattern

AI AppCo selects the foundation model, injects codebase context, applies quality filters, and presents the refined suggestion. The customer interacts exclusively with AI AppCo's interface and contacts AI AppCo for support. The foundation model providers are invisible to the customer. AI AppCo is primarily responsible for fulfilling the promise to provide the code assistance service. This is consistent with **principal** treatment under this indicator.

## 2<sup>Indicator</sup> Inventory Risk

This indicator examines whether the AI application company obtains, or commits to obtain, the specified good or service before it is transferred to the customer—bearing the risk that it may be unable to recover the cost of what it has acquired.

For AI application companies, inventory risk takes less conventional but equally substantive forms. Many companies negotiate minimum annual API commitments with foundation model providers in exchange for volume-discounted per-token rates (**see Section 3.1**), obligating the company to pay regardless of whether customers consume the underlying inference capacity. Others pre-purchase token blocks at volume discounts or license foundation model weights for self-hosting (**see Section 6.1**), investing in productive capacity before any customer transaction. In each case, the company has committed to obtain the service before transfer—economically equivalent to a retailer purchasing goods for resale.

This indicator is most relevant for companies with subscription-based pricing models (common in Archetypes A and B, and the fixed-fee component of Archetype C). The company incurs real, variable API costs with every customer interaction while collecting fixed monthly revenue. Per-token costs for leading models have declined by over 80% in some cases within 18 months—a favorable tailwind—but the risk runs in both directions. The company absorbs these fluctuations, profiting or suffering from the spread between its pricing and its input costs.

Conversely, if the company passes through API costs with a transparent markup or charges customers based on actual tokens consumed at a near-zero margin on the model cost component, inventory risk is minimal—the company has not committed to obtain the service in advance of the customer’s consumption. This pass-through model is more common in pure consumption-based Archetype C arrangements where pricing is directly tied to token usage, and it strengthens the **agent** argument.

### Applying to the illustrative fact pattern

AI AppCo charges \$20/seat/month regardless of how many tokens are consumed on behalf of the developer. AI AppCo has negotiated minimum annual API commitments with its foundation model providers, obligating it to pay for inference capacity whether or not its customer base fully utilizes it. AI AppCo bears the full risk that per-token costs may increase. This is consistent with principal-like inventory risk.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

### 3<sup>Indicator</sup> Pricing Discretion

Most AI application companies set their own pricing independent of the foundation model provider's API pricing. A coding assistant might charge \$20/month while its underlying model costs fluctuate between \$0.50 and \$5.00 per user per month depending on usage intensity. That level of pricing discretion is a strong **principal** indicator.

This indicator may carry less weight in consumption-based arrangements where the application company's pricing is transparently linked to per-token costs with a fixed markup percentage. In such case, the company's price discretion is limited to the markup, which may not be sufficient on its own to support principal treatment. Conversely, for subscription priced model where the customer has no visibility into or connection between the price paid and the underlying model costs, pricing discretion is typically strong across all archetypes.

#### Applying to the illustrative fact pattern

AI AppCo sets its own \$20/seat/month price, which bears no transparent relationship to per-token API costs. The customer does not know or care about the underlying model pricing. AI AppCo has full discretion in establishing the price. This level of pricing discretion is consistent with **principal** treatment.

## 2.2 The Multi-Model Complication

Many application layer companies route queries across multiple foundation models—selecting the optimal model based on task complexity, cost, latency, or specialized capability. When the company makes the model selection decision on behalf of the customer, this strengthens the principal argument. The customer is purchasing the output and the intelligence of the routing layer, not access to any specific model. The foundation model providers are suppliers to the application company, much like component manufacturers are suppliers to an electronics OEM.

## 2.3 Gross-vs-Net Implications When Multiple Performance Obligations Exist

When the distinctiveness analysis in [Section 1.4](#) results in multiple performance obligations — for example, an on-device AI component identified as a distinct license and a separate cloud-based model access service — the gross-vs-net determination and revenue recognition pattern must be evaluated independently for each obligation. Several consequences follow:

Executive Summary

The AI Application Layer: A  
New Software Taxonomy

Revenue Recognition: Key  
ASC 606 Judgments

Cost of Revenue: The New  
Economics of COGS

R&D, Capitalization, and the  
Build vs. Buy Continuum

Subscription vs. Consumption:  
Revenue Model Complexity

Licensing, Intellectual Property,  
and Data Economics

Financial Statement  
Presentation and Disclosure

Internal Controls, SOX Readiness,  
and Audit Considerations

IFRS and Ind AS: Comparative  
Notes

How Uniqus Can Help



## Split gross vs. net conclusions

The entity may reach different principal-agent conclusions for different obligations within the same contract. For example, a company may be **principal** for its proprietary platform component (reporting that revenue **gross**) while acting as **agent** for a separable model access pass-through (reporting **net**) — or principal for both, depending on the level of integration, control, and risk-bearing analyzed under the indicators in **Section 2.1**.



## Key Judgment

The **gross-vs-net** determination is not a one-time assessment. As AI application companies evolve their products—adding more proprietary transformation layers, switching between models, or adjusting pricing structures—the analysis must be revisited. A company that was appropriately reporting **net** in its early stage (when it was primarily passing through model outputs) may need to transition to **gross** reporting as its proprietary value-add deepens. This transition requires careful documentation, consistent methodology, and proactive auditor engagement.



## Different recognition patterns

Even where the entity is **principal** for all obligations, recognition timing may differ. The platform service obligation is typically recognized ratably over the service period, while a model access component may follow a different pattern depending on whether it constitutes a right-to-use license (functional IP — recognized at a point in time) or a right-to-access license (symbolic IP — recognized over time) under ASC 606-10-55-54 through 55-60.

Executive Summary

The AI Application Layer: A  
New Software Taxonomy

**Revenue Recognition: Key  
ASC 606 Judgments**

Cost of Revenue: The New  
Economics of COGS

R&D, Capitalization, and the  
Build vs. Buy Continuum

Subscription vs. Consumption:  
Revenue Model Complexity

Licensing, Intellectual Property,  
and Data Economics

Financial Statement  
Presentation and Disclosure

Internal Controls, SOX Readiness,  
and Audit Considerations

IFRS and Ind AS: Comparative  
Notes

How Uniquis Can Help

## Transaction Price Allocation and SSP for AI Products

When the analysis in [Sections 1.4](#) identifies multiple performance obligations with potentially different gross-vs-net conclusions and recognition patterns, the transaction price must be allocated to each obligation — a process that depends on establishing reliable standalone selling prices.

### Transaction price allocation:

When multiple performance obligations exist, the transaction price must be allocated across obligations based on relative standalone selling prices under ASC 606-10-32-28 through 32-35. For AI application companies, establishing standalone selling prices for components that are rarely sold separately (e.g., model access without the proprietary platform) requires significant judgment — often relying on [adjusted market assessment](#) or [expected cost plus margin](#) approaches.

### SSP Considerations for Novel AI Products and Highly Variable Pricing:

Establishing standalone selling prices for AI products presents challenges that traditional software companies rarely face. The SSP hierarchy under ASC 606-10-32-33 through 32-35 applies, but its practical application is strained by the AI application layer's characteristics:

- **Limited transaction history:** Many AI features are new to market. A product launched two months ago may have only 10–15 enterprise transactions—insufficient for a statistically meaningful observable price dataset.
- **High pricing variability:** When management is actively experimenting with pricing (varying by customer segment, geography, or competitive situation), the observed prices may reflect management discretion rather than a stable standalone selling price. ASC 606-10-32-34(c) provides that a standalone selling price is “highly variable” when the entity sells the same good or service to different customers for a broad range of amounts such that a representative standalone selling price is not discernible from past transactions.
- **Bundling as default:** Most AI products are sold as part of a platform bundle rather than standalone. Truly standalone transactions may not exist for newly released features.

### Estimation Approaches

**Adjusted market assessment:** Where competitor pricing for comparable AI products is available, the entity may reference market data—adjusting for differences in functionality, model quality, and customer segment. Given the rapid evolution of AI pricing benchmarks, this data has a short shelf life and should be refreshed frequently.

**Expected cost-plus-margin:** For AI products where the dominant cost driver is foundation model inference, the entity may estimate SSP based on the expected API cost per user plus a target margin. This approach has the advantage of being grounded in observable, measurable inputs (per-token costs, expected utilization) but requires careful documentation of the target margin assumption.

**Residual approach:** Under ASC 606-10-32-34(c), the residual approach is permitted only when the selling price is [highly variable](#) or [uncertain](#). For newly launched AI modules with limited history, this may be appropriate—but companies should document why neither the adjusted market assessment nor the expected cost-plus-margin approach can be applied.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&amp;D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help



### Practical Tip

Establish a “SSP Committee” or formalized review process that reassesses standalone selling prices for AI products at least **quarterly**. Document the estimation methodology, inputs, and rationale at each reassessment date. As transaction history accumulates, transition from estimation-based approaches to observable prices. The SEC staff has consistently emphasized the need for **contemporaneous documentation of SSP determinations**.

Executive Summary

The AI Application Layer: A  
New Software Taxonomy

**Revenue Recognition: Key  
ASC 606 Judgments**

Cost of Revenue: The New  
Economics of COGS

R&D, Capitalization, and the  
Build vs. Buy Continuum

Subscription vs. Consumption:  
Revenue Model Complexity

Licensing, Intellectual Property,  
and Data Economics

Financial Statement  
Presentation and Disclosure

Internal Controls, SOX Readiness,  
and Audit Considerations

IFRS and Ind AS: Comparative  
Notes

How Uniquis Can Help

## 2.5

# Marketplace Sales and Purchases

AI application companies increasingly distribute products through cloud marketplaces (AWS Marketplace, Azure Marketplace, Google Cloud Marketplace) and purchase foundation model access through the same channels. This creates layered principal-agent questions:

### Selling Through Marketplaces

- **Who is the principal?** When an AI application company sells its product through a cloud marketplace, three parties are involved: the **AI company** (vendor), the **marketplace operator** (e.g., AWS), and the **end customer**. The vendor must determine whether it or the marketplace operator is the principal. In most current marketplace arrangements, the AI company is the principal—the marketplace operator facilitates distribution and billing but does not control the specified good or service before transfer. The marketplace commission (typically 3–20%) is a selling expense, not a reduction of the transaction price — the marketplace operator is providing a distribution service, not purchasing the vendor’s product for resale. If the marketplace operator is also a customer of the AI company, the commission should be evaluated as potential consideration payable to a customer under ASC 606-10-32-25.
- **Revenue recognition timing:** Marketplace billing cycles may differ from the vendor’s direct billing cycles. The vendor must recognize revenue based on when the performance obligation is satisfied—not when the marketplace remits payment.
- **Consumption data reconciliation:** For consumption-based products sold through marketplaces, the vendor must reconcile its internal usage metering with the marketplace’s reported consumption. Discrepancies create accrual estimation challenges.

### Purchasing Through Marketplaces

When the AI application company purchases foundation model API access through a cloud marketplace (e.g., using committed cloud spend credits to pay for model inference), the accounting treatment depends on whether the marketplace purchase is economically equivalent to a direct API contract or involves different terms, pricing, or risk allocation. The principal-agent indicators from **Section 2.1** should be applied to the vendor’s own arrangement with the marketplace—specifically, whether the marketplace is acting as **agent** (pass-through) or **principal** (reseller) for the model access.

## 2.6 Mid-Contract Modifications

AI application layer pricing models are evolving so rapidly that contract modifications—adding new AI modules, changing pricing tiers, switching foundation model providers, or adjusting usage caps—are occurring mid-contract with increasing frequency. Under ASC 606-10-25-10 through 25-13, a contract modification is accounted for as either:

- **A separate contract** (ASC 606-10-25-12)—when the modification both adds distinct goods or services and prices them at their standalone selling prices;
- **A termination of the existing contract and creation of a new contract** (ASC 606-10-25-13(a))—when the remaining goods or services are distinct from those transferred before the modification; or
- **A cumulative catch-up adjustment** (ASC 606-10-25-13(b))—when the remaining goods or services are not distinct and form part of a single performance obligation that is partially satisfied.

### Contract Modification Decision Framework - ASC 606-10-25-12 through 25-13

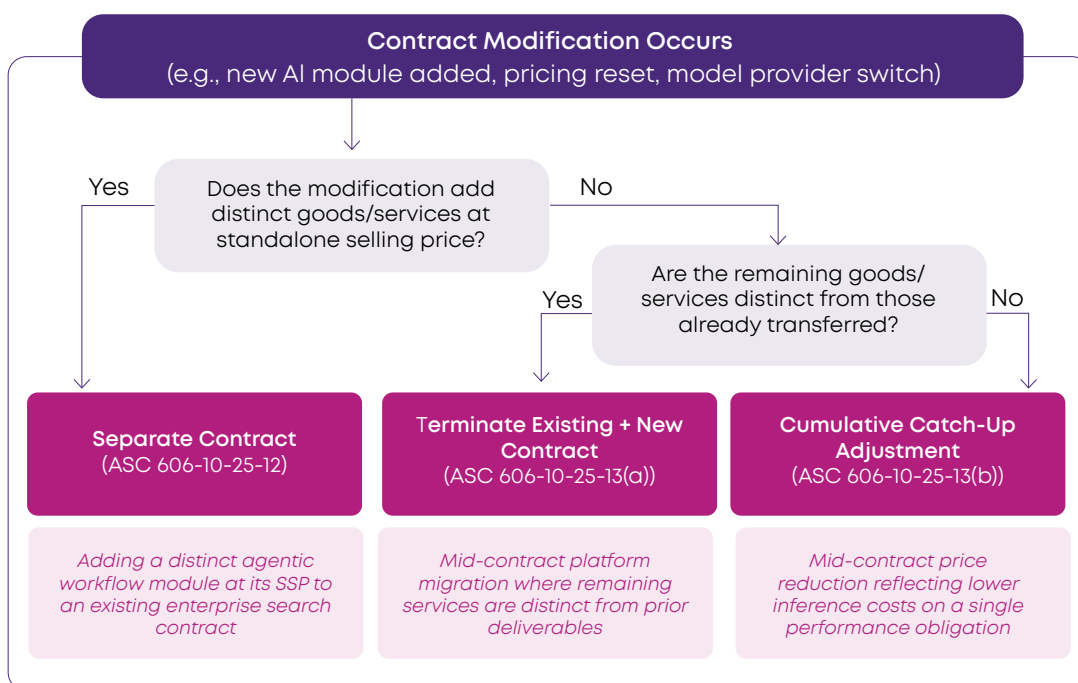


Figure 3: Contract Modification Decision Framework — ASC 606-10-25-12 through 25-13. The three modification pathways applied to AI-specific scenarios.

When the AI application company switches its underlying foundation model (e.g., migrating from one provider to another mid-contract), the customer's experience may or may not change materially. If the switch is transparent to the customer and does not alter the nature of the promise, it may not constitute a contract modification. If the switch changes the scope or price of the arrangement, **modification accounting** applies.

When the vendor adds a new AI capability (e.g., agentic workflow automation) to an existing enterprise contract, the entity must assess whether the new feature is distinct from the existing services and whether the pricing reflects its standalone selling price. Given the novelty of many AI features, SSP evidence may be limited (see the SSP discussions in [Section 2.4](#)).

In the rapidly declining AI cost environment, customers may negotiate mid-contract price reductions reflecting lower model inference costs. A pricing-only change that does not add or remove goods or services is a modification that typically results in a **prospective adjustment** to the remaining transaction price—recognized over the remaining term.

## Variable Consideration and the Constraint

For companies with consumption-based or hybrid revenue models (common in Archetype C), several specific contract terms create variable consideration under ASC 606-10-32-5 through 32-8:

- **Volume tiers:** The per-unit price decreases as cumulative usage crosses defined thresholds (e.g., the first 10,000 tokens at \$0.01, the next 50,000 at \$0.008).
- **Usage credits with overage pricing:** The customer pre-purchases a block of credits at a discounted rate, then pays overage rates for usage exceeding the block.
- **Minimum commitments with variable upside:** The customer guarantees a minimum annual spend but actual consumption may far exceed it.
- **Outcome-based pricing components:** Some contracts tie a portion of fees to measurable results (documents processed, tickets resolved), creating variability dependent on the customer's internal adoption.

These terms are not unique to Archetype C, Enterprise middleware (Archetype B) contracts increasingly include consumption-based components, and even subscription-based search platforms (Archetype A) may include usage-cap overages.

Under ASC 606-10-32-8, the entity must estimate variable consideration using either the **expected value method** (probability-weighted) or the **most likely amount method**, whichever better predicts the amount of consideration to which entity is entitled.

**The entity then applies the constraint under ASC 606-10-32-11:** variable consideration is included in the transaction price only to the extent that it is probable that a significant reversal in the amount of cumulative revenue recognized will not occur when the uncertainty associated with the variable consideration is subsequently resolved. In assessing the constraint, the entity considers factors in ASC 606-10-32-12, including whether:

- The amount of consideration is highly susceptible to factors outside the entity's influence;
- The uncertainty is not expected to be resolved for a long period of time;
- The entity's experience with similar contracts is limited;
- The entity has a practice of offering a broad range of price concessions or changing the payment terms and conditions of similar contracts in similar circumstances;
- The contract has a broad range of possible consideration amounts.

The constraint is particularly relevant for contracts with enterprise customers that include large pre-purchased token or credit packages. Usage patterns may be unpredictable—a customer might consume 20% of their allocated tokens in one month and 200% the next, depending on project cycles and AI adoption maturity within the organization. These customer-driven usage fluctuations represent factors outside the entity's influence (ASC 606-10-32-12(a)). Additionally, companies with limited operating history selling AI-native products, the limited experience factor under ASC 606-10-32-12(c) may weigh toward constraining a larger portion of the variable consideration, at least in early contract periods. As the company accumulates usage data across customer cohorts, the constraint may be relaxed.

Pricing for new AI modules and features is uniquely volatile. A company may launch a new agentic workflow product at one price point, adjust it 60 days later based on market feedback, and restructure the pricing model entirely within a quarter. This creates challenges across multiple ASC 606 dimensions:

Executive Summary

The AI Application Layer: A  
New Software Taxonomy

**Revenue Recognition: Key  
ASC 606 Judgments**

Cost of Revenue: The New  
Economics of COGS

R&D, Capitalization, and the  
Build vs. Buy Continuum

Subscription vs. Consumption:  
Revenue Model Complexity

Licensing, Intellectual Property,  
and Data Economics

Financial Statement  
Presentation and Disclosure

Internal Controls, SOX Readiness,  
and Audit Considerations

IFRS and Ind AS: Comparative  
Notes

How Uniqus Can Help

- **Transaction price determination:** When prices are changing rapidly, the “price stated in the contract” may not reflect the economic substance of the arrangement if the vendor has an established pattern of providing concessions or pricing adjustments.
- **Standalone selling price estimation:** When a new AI feature has been sold at three different price points in its first 90 days, no single price may be “directly observable” under ASC 606-10-32-33. The entity must estimate SSP using adjusted market assessment or expected cost-plus-margin approaches (see the SSP discussion in [Section 2.4](#)).



#### Practical Tip

Companies should document their constraint analysis contemporaneously, including the estimation method selected, the historical data (or lack thereof) supporting the estimate, and the specific factors considered under ASC 606-10-32-12. This documentation will be a focal point for auditors and, for companies approaching public markets, SEC staff review.

Executive Summary

The AI Application Layer: A New Software Taxonomy

#### Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniqus Can Help

## 2.8

### Breakage: Unused Tokens and Credits

Pre-sold token packages, credit bundles, and pre-paid usage allowances create **contract liabilities** on the balance sheet.

Breakage – the portion of prepaid rights that customers are not expected to exercise—is addressed in ASC 606-10-55-46 through 55-49. The guidance provides two recognition methods depending on whether the entity expects to be entitled to a breakage amount:

- If the entity expects to be entitled to breakage (based on the constraining guidance in ASC 606-10-32-11 through 32-13): The entity recognizes the expected breakage amount as revenue in proportion to the pattern of rights exercised by the customer (ASC 606-10-55-48). This means breakage revenue is recognized ratably as the customer consumes credits—not upfront at contract inception.
- If the entity does not expect to be entitled to breakage: The entity recognizes breakage revenue only when the likelihood of the customer exercising its remaining rights becomes remote (ASC 606-10-55-48).

In a prepaid token package, the transaction price is fixed at the amount paid by the customer (e.g., \$50,000 for 1 million tokens) regardless of the expected breakage amount. Breakage affects only the timing of revenue recognition—how the fixed transaction price is recognized over the contract period—not the total amount of revenue to be recognized. Accordingly, the reassessment requirement under ASC 606-10-32-14 (which requires updating estimates of variable consideration at end of each reporting period) does not apply to changes in breakage estimates. A change in expected breakage may shift the recognition pattern of revenue among periods but does not change the total transaction price allocated to the performance obligation.

### Illustrative AI fact pattern

An enterprise customer purchases a 12-month token package of 1 million tokens for \$50,000. Based on historical cohort analysis, the AI application company estimates that customers in this tier typically use 80% of their tokens (800,000 tokens). The entity expects to be entitled to a breakage amount associated with the unused 20%. Under ASC 606-10-55-48, the \$50,000 total transaction price is recognized as revenue proportionally as the customer exercises its rights—with the breakage portion recognized in proportion to the pattern of actual token consumption, not upfront, and not deferred until expiration.

### SEC and regulatory lens

Breakage methodologies are a recurring area of SEC staff inquiry, particularly for technology companies with prepaid credits and virtual currencies. For AI application companies with limited operating history, the SEC may expect conservative breakage estimates supported by robust cohort data and frequent re-estimation as usage patterns mature. Companies approaching IPO should expect breakage policy to be an **area of focus in S-1 review**.

Estimating breakage requires historical usage data, customer cohort analysis, and an understanding of how AI usage patterns evolve within organizations. For a rapidly growing company with limited operating history, the breakage estimate is inherently judgmental and may require frequent revision.

# 3 Cost of Revenue: The New Economics of COGS

The AI application layer has reintroduced meaningful variable costs into the software business model. Traditional SaaS companies enjoyed near-zero marginal cost per additional user; AI-native companies face real, measurable inference costs for every customer interaction. Industry data paints a stark picture: traditional SaaS gross margins of 78–90% contrast sharply with AI-first companies averaging 25–65% depending on maturity and infrastructure strategy.

This Section addresses the accounting classifications and presentation of these costs – particularly foundation model API costs– and the implications for financial statement presentation and investor communication. The discussion in this Section assumed the AI application company has concluded it is a **principal** under the analysis in [Section 2](#) (i.e., revenue is reported on a **gross** basis). If the entity is **agent**, API costs are **netted** against revenue and do not appear as a separate cost of revenue line items.

## 3.1 Foundation Model API Costs: Classification and Presentation

Foundation model API costs (the per-token charges) paid to model providers—are classified within cost of revenue. For many AI application companies, this single line item represents 30–60% of revenue, creating a gross margin profile that investors and analysts conditioned on SaaS economics find unfamiliar.

The classification of API costs within cost of revenue is straightforward under US GAAP when the entity is a principal, these are direct costs of providing the services to customers.

The more interesting questions are:



### Granularity of cost tracking

Companies using multiple models at different price points may need cost allocation systems that associate specific API costs with specific revenue streams, customer segments, or product tiers. This is relevant for segment reporting under ASC 280 to the extent that API cost data at the segment level is regularly provided to the CODM and included in the reported measure of segment profit or loss. Under ASU 2023-07 (amendments to ASC 280), significant segment expenses that are regularly provided to the CODM or that are easily computable from information regularly provided, must be disclosed.

*Example: A company uses Model A (lower cost, adequate for routine queries) and Model B (higher cost, used for complex reasoning tasks) across two product lines—a consumer search product and an enterprise analytics platform. If the CODM reviews profitability by product line, the company should allocate Model A and Model B API costs to each product line for segment reporting purposes.*

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help



### Timing of cost recognition

API costs are typically billed monthly based on consumption. For subscription customers, the company may incur API costs in a pattern that differs from revenue recognition. Under general US GAAP accrual principles, costs are recognized when incurred, irrespective of when the related revenue is recognized. However, proper accrual of estimated API costs at each reporting date is necessary when billing occurs in arrears and actual usage data is not yet available.



### Committed spend arrangements

Some companies negotiate minimum annual commitments with foundation model providers in exchange for volume discounts. The committed amount is typically recognized as a prepaid asset and expensed as API costs are incurred. If actual usage falls below the commitment, the entity should evaluate whether a loss provision is appropriate ([see Section 5.3](#) for the minimum commitment accounting framework, including the ASC 450-20 and ASC 842 analysis)

For **IFRS reporters**, the entity would apply IAS 37 (Provisions, Contingent Liabilities) to evaluate whether the contract is onerous.



### Agentic workflow cost attribution

For AI application companies deploying agentic AI platforms ([see Section 1.3](#)), the cost per customer outcome is inherently variable — a single autonomous task may trigger anywhere from 3 to 50+ foundation model calls depending on complexity, branching logic, and the number of tool invocations required. Unlike traditional per-query models where cost-per-interaction is relatively predictable, agentic workflows create a distribution of costs per outcome that must be estimated for accrual purposes. Companies should track cost-per-task distributions by task type and customer segment to support both COGS accruals and the gross margin narrative discussed in [Section 3.2](#).

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

**Cost of Revenue: The New Economics of COGS**

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniqus Can Help

## The Gross Margin Narrative Problem

AI application companies face a **disclosure** and **investor communication** challenge. A 40% gross margin may be entirely healthy for a company whose value proposition is the intelligence of its orchestration and transformation layer—but it triggers alarm bells for investors benchmarking against SaaS norms.

Companies should proactively address this **perception gap** in their investor communications and, for public companies, in MD&A. Several dynamics support the case that AI application layer gross margins are structurally different from — but not inferior to — traditional SaaS:

### Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

### Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

### Cost deflation tailwind

Per-token costs have declined 80–90%+ for leading models since 2023. A company operating at 40% gross margin today could see margins expand toward 60–70% as cost curves continue to decline—if it maintains pricing power. Disclosing the trajectory of per-unit inference costs (without revealing competitively sensitive absolute numbers) helps investors model margin expansion.

### Model routing optimization

Companies that build intelligent routing layers (sending simple queries to lower-cost models and complex queries to higher-cost ones) may significantly reduce blended API costs over time. The resulting margin improvement is a tangible, engineered outcome, not a one-time benefit. (For the accounting treatment of routing layer development costs – **capitalization** under ASC 350-40 vs. **expense** under ASC 730 – **see Section 4.1**)

### Self-hosted model migration

At sufficient scale, some companies bring model inference in-house, reducing per-query costs by 60–80% but requiring \$2–5 million+ in upfront infrastructure investment. The accounting treatment of this transition depends on several factors:

- **GPU and server hardware** purchases shift spend from operating expenses (API costs in COGS) to capital expenditures on the balance sheet, with depreciation flowing through COGS over the asset's useful life (commonly 3–5 years). Gross margins improve on a reported basis, but free cash flow may decline during the investment period.
- **Software development** costs incurred to build the self-hosting platform (model serving infrastructure, optimization layers) may be capitalizable under ASC 350-40 if the criteria are met (**see Section 4.1** below), or under ASC 985-20 if the software is to be sold, leased, or marketed externally, further shifting spend from income statement to the balance sheet.
- The decision to capitalize is not automatic—it depends on the project's scope and the applicable criteria under ASC 350-40 (as amended by ASU 2025-06, when adopted).

This transition can create the appearance of improving economics on the income statement while simultaneously consuming more cash. Transparent disclosure of the transition, its financial impact, and the expected timeline for ROI is essential for investor confidence. (For further discussions on scoping under ASC 360 and ASC 350-40 – **see Section 4.1 and 4.2**)



**Practical Tip**

Consider disclosing (a) the trend in **blended inference cost** per unit of output (without absolute dollar amounts if competitively sensitive), (b) the **percentage of queries** routed to lower-cost models versus premium models, and (c) the **expected gross margin** trajectory as cost deflation and routing optimization take effect. This gives investors a framework to evaluate margin expansion potential without requiring them to benchmark against SaaS comparables.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

**Cost of Revenue: The New Economics of COGS**

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind.AS: Comparative Notes

How Unique Can Help

### 3.3

## DISE Implications for AI Application (Public) Companies

Under ASU 2024-03 (DISE) (effective for reporting periods after December 15, 2026 with early adoption permitted), AI application companies will need to disaggregate relevant expense (including cost of revenue) into prescribed **natural expense** categories. The foundation model API cost likely sits in the “other” residual category—it is neither employee compensation, nor depreciation, nor purchased inventory in the traditional sense. It is a **purchased service**.

For companies where API costs are the dominant COGS component, this creates a presentation challenge: the “other” category may dwarf the named categories, potentially requiring **supplemental disclosure** to explain what “other” comprises.

Companies should consider whether **voluntary** additional disaggregation (breaking out “foundation model inference costs” as a named line) would better serve users of the financial statements. Before considering voluntary additional disaggregation, companies should evaluate:

#### Investor expectations

Sell-side analysts and institutional investors increasingly expect visibility into AI inference cost trends. Earnings calls for publicly traded AI companies have featured questions about **API cost trajectory** and **gross margin sustainability**. Voluntary disaggregation may preempt these inquiries and demonstrate transparency.

#### Competitive sensitivity

Disclosing the precise magnitude of foundation model costs relative to revenue reveals cost structure details that competitors could exploit. Companies should weigh transparency against competitive harm.

#### MD&A implications

Under Regulation S-K, Item 303 (MD&A), registrants must discuss known trends, events, or uncertainties that are reasonably likely to have a material effect on results of operations or financial condition. If API cost deflation (or inflation) is a material driver of gross margin trends, MD&A discussion is likely required regardless of whether the company provides voluntary DISE disaggregation. Companies should ensure **consistency between their DISE footnote and MD&A narrative**.

#### Materiality

If API costs are immaterial relative to total cost of revenue, additional disaggregation may be unnecessary. However, for most AI application layer companies, these costs are likely to be material.

For a comprehensive analysis of ASU 2024-03’s requirements — including the prescribed expense categories, tabular footnote format, and implementation considerations — see Uniqus Consultech’s ASC Insights publication: **FASB’s Accounting Standard Update (ASU 2024-03) — Disaggregation of Income Statement Expenses (Subtopic 220-40)** (March 2026), available at [uniquis.com/insights](https://uniquis.com/insights)

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

**Cost of Revenue: The New Economics of COGS**

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniqus Can Help

# 4 R&D, Capitalization, and the Build vs. Buy Continuum

## 4.1

### What may be Capitalizable Under ASC 350-40?

AI application companies develop a range of proprietary assets that may qualify for capitalization under **ASC 350-40** (Internal-Use Software). Importantly, the scoping determination is a critical threshold question: if the company derives significant revenue from licensing its AI platform to third parties (as distinct from hosting it as a service), **ASC 985-20** (Costs of Software to be Sold, Leased, or Marketed) may apply instead of – or in addition to – ASC 350-40. ASC 985-20 imposes a higher capitalization threshold (technological feasibility), which may result in more costs being expensed.

**ASU 2025-06 (Issued September 2025)** The FASB's recent Targeted Improvements to the Accounting for Internal-Use Software is directly relevant to AI application companies. The ASU eliminates the legacy stage-based capitalization model (preliminary project stage, application development stage, post-implementation stage) and replaces it with a two-criteria threshold that must be met simultaneously before capitalization begins. The ASU is effective for annual periods beginning after December 15, 2027, with early adoption permitted. Notably, the ASU does not create separate guidance for AI software – the same framework applies to all internal-use software. However, the criteria will require particularly careful judgment for AI projects, where iterative development and rapid model evolution can blur the line between experimentation and committed development.

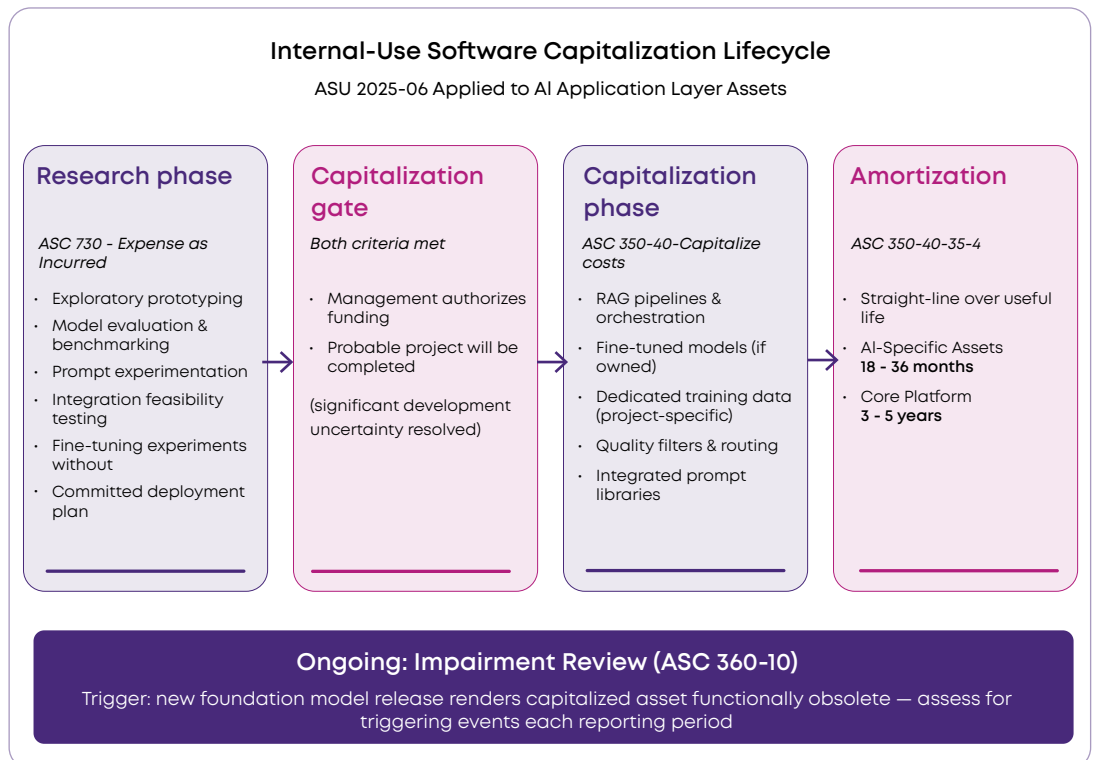


Figure 4: Internal-Use Software Capitalization Lifecycle Under ASU 2025-06 — Applied to AI Application Layer Assets. Research-phase costs (ASC 730) are expensed as incurred. Capitalization under ASC 350-40 begins when two criteria are met simultaneously. AI-specific useful lives typically range from 18–36 months for rapidly evolving assets to 3–5 years for core platform components.

Under [ASU 2025-06](#), capitalization begins when both of the following criteria are met simultaneously:

- **Management has authorized and committed to funding the project.** This requires more than informal approval—it requires a documented commitment of resources (personnel, infrastructure, budget) sufficient to complete the project. For AI application companies, this criterion may be met when the project transitions from an exploratory prototype to a resourced development effort with assigned engineering teams and committed compute budgets.
- **It is probable the project will be completed.** Central to this assessment is whether significant development uncertainty has been resolved — meaning the software’s novel or unproven functions have been validated through coding and testing, and performance requirements are substantially defined. For AI projects, completion probability should consider the track record of similar projects at the entity, the availability of the required foundation model capabilities, and the stability of the underlying technology stack. Projects that depend on foundation model capabilities that do not yet exist may not meet this criterion.

For AI application layer assets, significant development uncertainty may manifest as:

- Uncertainty about whether the foundation model can deliver the required output quality for the intended use case (e.g., whether a model can consistently generate accurate legal contract analysis at an acceptable error rate).
- Uncertainty about whether the proprietary orchestration layer (RAG pipeline, model routing, quality filters) can integrate with the foundation model to produce a viable product at acceptable latency and cost.
- Uncertainty about whether the combination of multiple AI components (prompt libraries, fine-tuned models, retrieval systems) will function together as intended.

The resolution of significant development uncertainty requires demonstrated — not merely designed — functionality. [For a coding assistant](#), this might mean the model routing and quality filtering layers have been tested against benchmark code completion tasks and meet predefined acceptance criteria. [For an enterprise search platform](#), this might mean the RAG pipeline has been validated against a representative dataset with measurable retrieval accuracy thresholds.

Executive Summary

The AI Application Layer: A  
New Software Taxonomy

Revenue Recognition: Key  
ASC 606 Judgments

Cost of Revenue: The New  
Economics of COGS

**R&D, Capitalization, and the  
Build vs. Buy Continuum**

Subscription vs. Consumption:  
Revenue Model Complexity

Licensing, Intellectual Property,  
and Data Economics

Financial Statement  
Presentation and Disclosure

Internal Controls, SOX Readiness,  
and Audit Considerations

IFRS and Ind AS: Comparative  
Notes

How Uniquis Can Help

Applying these principles to AI application layer assets:

### Proprietary orchestration and retrieval layers

RAG pipelines, vector databases, query routing algorithms, ranking and synthesis engines, and permissions frameworks are generally capitalizable once the probable-to-complete threshold is met—provided these assets are used internally (including to provide SaaS service) rather than licensed to customers. These are the company's core software assets.

### Fine-tuning of foundation models

This is the judgment-intensive area. Fine-tuning a foundation model involves modifying a third-party model's weights to improve performance for specific use cases. Whether the resulting fine-tuned model is capitalizable depends on several factors:

#### Ownership

Does the company own the fine-tuned weights, or do license terms restrict modification, deployment, or redistribution? If the foundation model provider retains ownership of the fine-tuned weights — with the entity accessing the customized model only via API — the base model is likely not the entity's internal-use software asset, and fine-tuning costs may not be capitalized as upgrades or enhancements under ASC 350-40-25-9.

#### Persistence

Is the fine-tuned model deployed as a persistent, long-lived asset used across multiple product versions, or is it ephemeral—frequently retrained, superseded, or treated as an optimization experiment? Under ASC 350-40-35-4, capitalized software is amortized over its useful life. If the useful life is negligible (e.g., the model is retrained monthly as new foundation model versions are released), capitalization may technically apply but the resulting asset would be amortized almost immediately — making expense-as-incurred the more practical treatment.

#### Probable completion

Has the fine-tuning effort moved beyond experimentation (significant development uncertainty resolved under ASU 2025-06) to a state where the model is expected to be completed and used for its intended function?

Fine-tuning runs that are exploratory — testing whether a model can be adapted for a use case without a clear path to completion and deployment — remain research activity expensed under ASC 730.

Companies should document specific facts supporting capitalization.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

**R&D, Capitalization, and the Build vs. Buy Continuum**

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

### Prompt engineering libraries

Curated, tested, and maintained libraries of prompts that constitute the company's domain expertise (e.g., a legal AI's library of contract analysis prompts) are not independently capitalizable as a separate asset class under ASC 350-40 or any other US GAAP standard. However, when prompt development is an integral part of a broader internal-use software project — for example, prompts that are embedded in the application's codebase and essential to its functionality — the associated development costs may be capitalizable as part of that project's overall cost basis under ASC 350-40. Prompt development that is standalone — curated independently, used across multiple applications, or maintained as a separate library — more closely resembles internally generated intangible asset development, which US GAAP does not generally permit capitalizing. Such costs should be expensed as incurred under ASC 730.

### Training data curation

Costs to acquire, clean, structure, and label proprietary training datasets require careful evaluation. Licensed data dedicated exclusively to a specific internal-use AI software project — with no other use to the entity — may be capitalizable as a direct cost of that project under ASC 350-40, provided the costs are incurred after the applicable capitalization criteria are met. Licensed data that has other uses to the entity (e.g., used across multiple AI applications or to retrain existing models) may be capitalized as a separate intangible asset under ASC 350-30; however, amortization of that data asset is not capitalizable into the AI software project under ASC 350-40, as it represents an indirect cost. General-purpose data acquisition and curation that is not tied to a specific software development project or is incurred in the preliminary project stage is more appropriately characterized as a research activity expensed under ASC 730.

### LLM access fees used during development

Companies that use LLM tools (e.g., AI coding assistants such as GitHub Copilot) during the development of their own internal-use software should evaluate whether the associated fees are direct, capitalizable costs or indirect, noncapitalizable costs. **Usage-based LLM fees** incurred specifically for and dedicated to an identified software project may qualify as capitalizable external direct costs. **Subscription-based LLM fees** used across multiple projects or for general productivity purposes are generally indirect costs that should be expensed as incurred. The key distinction is whether the LLM access can be directly attributed to a particular software project and whether it represents an incremental cost that would not have been incurred absent that project.

Executive Summary

The AI Application Layer: A  
New Software Taxonomy

Revenue Recognition: Key  
ASC 606 Judgments

Cost of Revenue: The New  
Economics of COGS

**R&D, Capitalization, and the  
Build vs. Buy Continuum**

Subscription vs. Consumption:  
Revenue Model Complexity

Licensing, Intellectual Property,  
and Data Economics

Financial Statement  
Presentation and Disclosure

Internal Controls, SOX Readiness,  
and Audit Considerations

IFRS and Ind AS: Comparative  
Notes

How Uniquis Can Help

## 4.2

### Costs Eligible for Capitalization

Once the required criteria are met, the following costs are capitalizable under ASC 350-40:

- **Compensation and benefits** for employees directly involved in the development effort.
- **External direct costs** of materials and services consumed in the development (e.g., contractor fees, cloud compute costs dedicated to the project, licensed data acquired specifically for the project).
- **Interest costs** incurred during the development period, if the project is a qualifying asset under ASC 835-20.

Costs that are not capitalizable include general and administrative costs, overhead allocations, training costs, and costs incurred during the research phase before the required criteria are met.

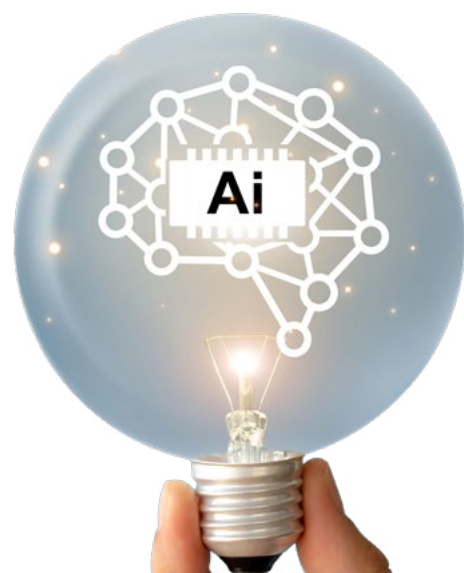
## 4.3

### The Capitalization Threshold Question

The applicable capitalization framework depends on how the AI application company delivers its product. Companies that deliver their product as a hosted service (SaaS) apply ASC 350-40 (Internal-Use Software) — this covers the vast majority of AI application layer companies. Companies that derive significant revenue from on-premise software licensing or distributing model weights must apply ASC 985-20 (Costs of Software to Be Sold, Leased, or Marketed), which imposes a higher capitalization threshold. Companies should carefully evaluate the nature of their customer arrangements to determine the applicable literature, and document the basis for their scoping conclusion.

Under ASC 985-20, costs are capitalized only after technological feasibility is established — a concept that requires either a detailed program design or a working model. Under current ASC 350-40, capitalization begins when the project enters the application development stage. Under ASC 350-40 as amended by ASU 2025-06, if adopted, the threshold shifts to a probable-to-complete assessment with resolution of significant development uncertainty.

All three thresholds present challenges for AI product development. When is an AI product “feasible” — or past preliminary stage, or probable to complete — given that its performance improves continuously through fine-tuning, data additions, and model upgrades? These milestones map imperfectly onto AI development, which is inherently iterative and often lacks a bright-line “done” state.



Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

**R&D, Capitalization, and the Build vs. Buy Continuum**

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

## 4.4

### Useful Life and Amortization

Capitalized internal-use software under ASC 350-40 is amortized over its estimated useful life on a straight-line basis (unless another systematic method better reflects the pattern of consumption). For AI application layer assets, useful life estimation is particularly challenging due to the pace of foundation model evolution. Companies should consider establishing shorter useful lives (18–36 months) for assets that are tightly coupled to specific foundation model versions, with longer lives (3–5 years) for proprietary platform infrastructure that is model-agnostic. Useful life estimates should be reviewed at each reporting period and adjusted if circumstances change.

## 4.5

### Impairment Risk for Capitalized AI Assets

Foundation model capabilities advance so rapidly that proprietary fine-tuned models, prompt libraries, and even custom RAG architectures can become obsolete when a new base model natively handles what the customization achieved. A fine-tuned model that took six months and \$2 million to develop may be rendered redundant by a new foundation model release that incorporates the same capability out of the box.

This creates **recurring impairment triggers** under ASC 360-10-35 (for long-lived assets, including capitalized internal-use software) that traditional software companies rarely face. The useful life of these assets—often estimated at 2–3 years—may prove optimistic given the pace of model evolution. A long-lived asset is tested for recoverability when events or changes in circumstances indicate that its carrying amount may not be recoverable. The asset fails the recoverability test when its carrying amount exceeds the sum of the undiscounted future cash flows expected from the asset's use and eventual disposition. If the asset is not recoverable, an impairment loss is measured as the amount by which the carrying value exceeds fair value.

Companies should establish technology review processes that assess, at least quarterly, whether capitalized AI assets retain their expected utility.



#### Practical Tip

Establish a clear, documented capitalization policy that specifically addresses AI-unique asset categories: fine-tuned models, prompt libraries, training datasets, and evaluation benchmarks. Define the criteria for each—what constitutes the probable-to-complete threshold, how useful life is estimated, and what triggers an impairment review. Auditors will expect this specificity, and companies approaching IPO will need it for SEC staff questions.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

**R&D, Capitalization, and the Build vs. Buy Continuum**

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniqus Can Help

## 4.6

# R&D Cost Presentation and the Investor Narrative

A significant portion of engineering spend at AI application companies goes to activities that do not produce capitalizable assets: prompt engineering experimentation, model evaluation and benchmarking, exploratory integration testing across model versions, and research into new AI capabilities. Under ASC 730-10-25-1, research and development costs are generally expensed as incurred. The resulting R&D-to-revenue ratio can reach 40–60%+ for early-stage companies.

This ratio is not inherently concerning—many successful software companies invested heavily in R&D during their growth phase. But for AI application companies, the composition of R&D is different: a substantial portion is consumed by external model evaluation costs and infrastructure experimentation, not traditional engineering salaries. **Clear disclosure of R&D composition** helps investors distinguish between productive investment and unsustainable cost structures.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

**R&D, Capitalization, and the Build vs. Buy Continuum**

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help



# 5 Subscription vs. Consumption: Revenue Model Complexity

## 5.1

### The Hybrid Model Challenge

The dominant pricing architecture in the AI application layer is hybrid: a base subscription fee plus consumption-based overage charges.

Under ASC 606, the fixed subscription and variable usage may represent – and the answer depends in part on the distinctiveness analysis in [Section 1.4](#):

- **A single performance obligation with variable consideration** – this may be the case if the platform delivers a series of distinct services (e.g., daily or monthly periods of AI-powered access) that are substantially the same and have the same pattern of transfer. Under ASC 606-10-25-14(b), such a series is accounted for as a single performance obligation, even though each period of service is individually distinct. The variable usage fees would then be allocated to distinct periods within the series under ASC 606-10-32-39 through 32-41.
- **Two or more distinct performance obligations requiring transaction price allocation** – if the subscription and consumption components are distinct under the framework in [Section 1.4](#), the transaction price is allocated between them based on relative standalone selling prices (ASC 606-10-32-28 through 32-35). The subscription component may be recognized ratably while the consumption component may qualify for the as-invoiced expedient if it independently meets the criteria.

#### The as-invoiced practical expedient

For AI application companies with pure consumption-based pricing—where the right to consideration corresponds directly with the value delivered to the customer (e.g., \$X per token consumed, \$Y per API call)—the practical expedient under ASC 606-10-55-18 may significantly simplify the accounting. Under this expedient, if an entity has a right to consideration from a customer in an amount that corresponds directly with the value to the customer of the entity's performance completed to date, the entity may recognize revenue in the amount to which it has a right to invoice. When applicable, this expedient eliminates the need to estimate variable consideration for the entire contract term and avoids the complexity of the constraint analysis. However, the entity should evaluate the conditions carefully—the expedient is not available when the invoiced amounts do not faithfully depict the value transferred (for example, when upfront fees are disproportionate to value delivered in early periods).

#### Stand-Ready vs. Specified-Quantity Obligations

An additional judgment that affects recognition timing is whether the AI application company's promise is a stand-ready obligation (providing continuous access to AI capabilities over a period) or a specified-quantity obligation (delivering a defined number of outputs — tokens, queries, or actions). The following indicators help distinguish between the two:

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

**Subscription vs. Consumption: Revenue Model Complexity**

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

### Indicators of a stand-ready obligation

- The customer benefits from the entity's continuous availability to process requests on demand
- The entity cannot predict or control when or how frequently the customer will use the service
- The contract does not specify a fixed number of outputs or transactions the entity must deliver
- Pricing is primarily time-based (monthly or annual subscription) rather than tied to a specified volume of deliverables
- Usage is driven primarily by the customer's end users or downstream customers rather than deliberate procurement decisions — the entity cannot predict or control the volume or timing of interactions
- The contract includes a usage cap, but the cap is set as a protective measure substantially higher than expected usage — effectively providing unlimited access

### Indicators of a specified-quantity obligation

- The contract specifies a defined number of tokens, credits, API calls, or transactions
- The customer's right to service terminates or changes when the specified quantity is consumed
- Pricing is directly tied to a volume of deliverables rather than a period of access
- The entity can measure progress toward satisfaction by counting units delivered against the contractual quantity
- The customer has the right to roll over unused volume (tokens, credits, or outcomes) into a future period, and must make a separate purchasing decision to obtain additional units beyond the specified quantity

**For subscription-based arrangements (common in Archetypes A and B), the promise is typically stand-ready:** the customer has continuous access to the platform's AI capabilities throughout the subscription period, regardless of how much they actually use it. Revenue is recognized ratably over the access period.

**For consumption-based arrangements (common in Archetype C), the characterization depends on the contract structure.** If the contract specifies a defined quantity of tokens or outputs (e.g., '1 million tokens'), the promise is a specified-quantity obligation and revenue is recognized as units are delivered. If the contract provides unlimited access at a per-unit price with no guaranteed minimum, the arrangement may be a stand-ready obligation with variable consideration.

The distinction matters because it affects the measure of progress, the applicability of series guidance under ASC 606-10-25-14(b), and the available practical expedients. Hybrid arrangements — where a base subscription provides stand-ready access and usage above a cap is priced per unit — may contain elements of both, requiring careful disaggregation.

Executive Summary

The AI Application Layer: A  
New Software Taxonomy

Revenue Recognition: Key  
ASC 606 Judgments

Cost of Revenue: The New  
Economics of COGS

R&D, Capitalization, and the  
Build vs. Buy Continuum

**Subscription vs. Consumption:  
Revenue Model Complexity**

Licensing, Intellectual Property,  
and Data Economics

Financial Statement  
Presentation and Disclosure

Internal Controls, SOX Readiness,  
and Audit Considerations

IFRS and Ind AS: Comparative  
Notes

How Uniqus Can Help

## Stand-Ready vs. Specified-Quantity Obligations

### Revenue Recognition Patterns for AI Application Companies

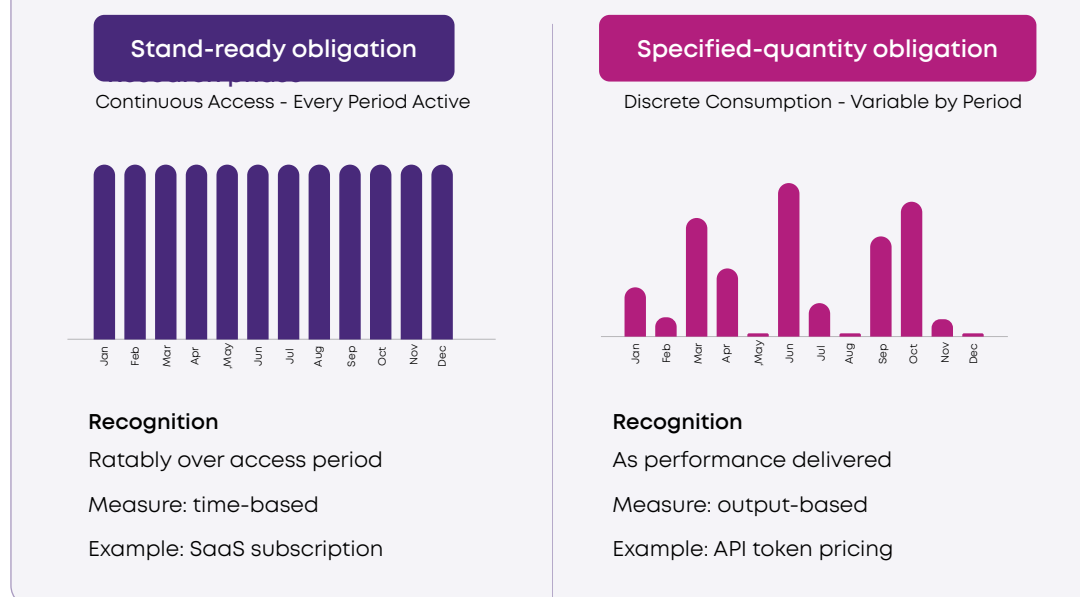


Figure 5: Stand-Ready vs. Specified-Quantity Obligations — Revenue Recognition Patterns for AI Application Companies

**ELA Flex (Product-Mix-Swap) Models:** Enterprise License Agreement (ELA) Flex arrangements allow customers to reallocate committed spend across the vendor's product portfolio during the contract term. In the AI application layer, this structure is gaining traction because the pace of model innovation and product releases makes it impractical for customers to commit to specific product allocations for multi-year terms.

- Performance obligation identification :** The threshold question is whether the ELA Flex conveys a **single stand-ready obligation** (continuous access to the vendor's platform portfolio) or **multiple distinct obligations** (access to each product). If the customer can benefit from each product independently and the vendor's promise to provide each product is separately identifiable (ASC 606-10-25-19 through 25-22), multiple performance obligations may exist—even though the customer's payment is pooled. However, where the products share a common infrastructure layer, common data model, and unified interface—and the reallocation right means the customer is purchasing access to the portfolio rather than specific products—a single stand-ready obligation may be supportable.
- Transaction price allocation :** The reallocation right creates uncertainty in transaction price allocation — while the total committed spend is fixed, the allocation across performance obligations (if multiple exist) depends on which products the customer ultimately consumes. The vendor must estimate this allocation using the relative standalone selling price framework under ASC 606-10-32-28 through 32-35.
- Practical approach — SaaS stand-ready ratable:** Where the ELA Flex is structured as a stand-ready obligation to provide continuous access to the AI platform portfolio, the entire committed amount is recognized ratably over the access period. This is the simpler, and often more supportable, conclusion where the products are deeply integrated. Where consumption-based overages exist above the committed amount, the overage component follows the variable consideration and constraint analysis in [Section 2.7](#).

### Key Judgment

The distinction between structuring an ELA Flex as a SaaS-stand-ready-ratable arrangement versus a consumption model has material revenue recognition implications. The stand-ready structure provides predictable **ratable revenue**; the consumption structure may result in **lumpy recognition** tied to actual product-mix utilization. Companies should involve technical accounting advisors at the contract design stage—not after execution.

## 5.2

### Freemium and Free-Tier Accounting

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

**Subscription vs. Consumption: Revenue Model Complexity**

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniqus Can Help

Most AI application companies offer **free tiers** to drive adoption and build network effects. The costs of serving free users—foundation model API costs, hosting, customer support—are period expenses recognized as incurred. A free-tier arrangement typically **does not meet** the contract criteria under ASC 606-10-25-1 — specifically, the requirement for identifiable payment terms (ASC 606-10-25-1(c)) and commercial substance (ASC 606-10-25-1(d)) — and therefore falls outside the scope of ASC 606.

The more relevant revenue recognition question arises when a free-tier user converts to a paid contract. At that point, the entity should evaluate whether the paid contract contains a **material right** under ASC 606-10-55-41 through 55-45 — for example, if the paid customer receives a discounted conversion price that they would not have received without the free-tier relationship, or if accumulated free-tier credits carry over into the paid tier at a discount to standalone selling price. If such a material right exists, it constitutes a **separate performance obligation** within the paid contract, requiring allocation of a portion of the transaction price. In practice, this analysis is most relevant for companies with structured upgrade pathways.

*Example “your first 1,000 tokens are free; paid tier starts at token 1,001 at a discounted rate for free-tier converts”).*

**Regarding capitalization of free-tier costs:** Some management teams have argued that free-tier costs have a future benefit (through feedback loops, training data generation, or usage pattern analysis). Under ASC 350-40, capitalization requires costs to relate to a specific internal use software project meeting defined criteria. Under ASC 730, general R&D costs are expensed as incurred.

## 5.3

### Enterprise Contracts with Minimum Commitments

Enterprise AI platform contracts often include minimum annual commitments—the customer agrees to consume (and pay for) at least a specified dollar amount of compute or tokens per year, typically in exchange for volume-discounted pricing.

**From the seller’s perspective**, the minimum commitment creates a fixed component of the transaction price (an unconditional right to consideration), which may be recognized over the service period relative to a pure usage-based model. The variable consideration above the minimum remains subject to the constraint under ASC 606-10-32-11 (*see Section 2.7* for the constraint analysis framework).

**From the buyer’s perspective**, the minimum commitment should be evaluated as an executory contract. If actual usage is expected to fall materially below the commitment, the entity should consider whether a loss provision is appropriate under ASC 450-20 (Loss Contingencies).

**Do minimum compute commitments create a lease?** A natural question when an AI company commits to a minimum annual spend with a foundation model provider is

whether the arrangement contains an **embedded lease** under ASC 842. The answer turns on whether the arrangement conveys the right to control the use of an identified asset — a specific, physically distinct piece of infrastructure that the customer directs and benefits from. In most AI API arrangements, it does not: the customer accesses shared, multi-tenant cloud infrastructure, does not control specific servers or GPUs, and the model provider retains substantive substitution rights (it can change which hardware serves the customer's requests at any time). Accordingly, most token- or API-based minimum commitments do not convey an identified asset and would not be in scope of ASC 842. An exception may arise in dedicated infrastructure arrangements where the customer is allocated specific, identified GPU capacity that the provider cannot substitute — such arrangements should be evaluated under ASC 842's identified asset and substitution rights criteria (ASC 842-10-15-9 through 15-16).

## 5.4 Non-Co-Terminus Contract

Traditional enterprise software contracts are co-terminus—all product subscriptions share the same start and end dates, simplifying revenue recognition and renewal. In the AI application layer, **co-terminus structures are breaking down** for several reasons:

- **Rapid product launches:** New AI modules are released and added to existing customer relationships on their own timelines, creating staggered subscription start dates.
- **Pilot-to-production transitions:** Enterprise customers often begin with a short-term pilot (3–6 months) for one AI product and expand to full deployment on a different timeline.
- **Multi-vendor orchestration:** Customers may layer AI products from different vendors with different contract cycles.

When contracts are non-co-terminus, the entity must determine whether overlapping contracts should be combined under ASC 606-10-25-9. Contracts are combined when they are entered into at or near the same time with the same customer and meet any of the criteria in ASC 606-10-25-9(a) through (c)—including when the contracts are negotiated as a package with a single commercial objective, when consideration in one contract depends on the other, or when the goods or services are a single performance obligation.

For AI application companies with staggered product add-ons, the combination assessment should be performed each time a new product is added. If the new product is priced below its standalone selling price because the customer has an existing

relationship, this may indicate the contracts were negotiated as a package with a single commercial objective (ASC 606-10-25-9(a)) — potentially requiring combination and reallocation of the transaction price across all performance obligations



Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

**Subscription vs. Consumption: Revenue Model Complexity**

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

## Outcome-Based and Success-Fee Pricing

An emerging trend is outcome-based pricing, where the AI application company charges based on measurable results (documents processed, tickets resolved, code deployments completed) rather than raw token consumption. Under ASC 606, outcome-based fees represent variable consideration subject to the estimation and constraint framework discussed in [Section 2.7](#).

It is now rapidly becoming a primary pricing model for certain AI applications, particularly in enterprise middleware (Archetype B) and agentic workflow platforms. The shift from seat-based to outcome-based pricing has several additional implications:

### Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

### Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniqus Can Help

### Seat-based vs. outcome-based—not just pricing, but performance obligation

A seat-based arrangement conveys the right to access; the outcome-based arrangement conveys the right to results. This distinction may affect whether the promise is a stand-ready obligation (access to the platform) or a specified-quantity obligation (delivery of a defined number of outcomes). [See Section 5.1](#) for the stand-ready vs. specified-quantity framework.

### Constraint implications

When pricing is tied to measurable outcomes (tickets resolved, documents processed, code deployments completed), the entity's ability to estimate variable consideration depends on the customer's internal adoption and workflow integration—factors outside the entity's influence. The constraint under ASC 606-10-32-12(a) may require excluding a significant portion of the estimated outcome-based revenue until uncertainty resolves.

### Cost recognition

The entity incurs API costs per action but recognizes revenue per outcome (which may require multiple API calls). As discussed in [Section 3.1](#), US GAAP does not require matching of cost and revenue timing — API costs are recognized when incurred regardless of when the related outcome-based revenue is recognized.

### Minimum Guarantee Fee

When outcome-based contracts include minimum guaranteed fees, the fixed minimum component is recognized over the service period while the outcome-based upside remains subject to the constraint — see [Section 5.3](#) for the minimum commitment framework.

**The estimation challenge is acute.** the company must predict the volume of outcomes the customer will achieve, which depends on the customer's internal adoption and usage patterns — factors largely outside the company's control, compounded by the likelihood that the company has limited operating history with outcome-based contracts.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

Illustrative Fact Pattern — Outcome-Based Pricing

AI LegalCo provides an enterprise customer with an AI-powered contract review platform that autonomously analyzes and redlines vendor agreements. The contract provides:

- Annual platform access fee: \$400,000 (fixed, regardless of usage)
- Per-contract fee: \$25 per agreement autonomously reviewed and redlined without attorney escalation

AI LegalCo determines the arrangement contains a **single stand-ready performance obligation** — the promise is to provide continuous access to the AI contract review platform, with each day of access being a distinct service within the series under ASC 606-10-25-14(b).

**Revenue recognition:** The \$400,000 fixed fee is recognized ratably over the contract term. The \$25 per-contract variable fee is allocated to each distinct day within the series under ASC 606-10-32-40, because the variable consideration relates specifically to AI LegalCo’s performance on that day and allocating it to each day is consistent with the allocation objective. AI LegalCo recognizes the variable fee as each contract review is completed.

**Cost recognition:** Each autonomous contract review triggers an average of 12 foundation model API calls (ranging from 5 to 40+ depending on agreement complexity, clause count, and the number of redline iterations). AI LegalCo recognizes API costs when incurred — not matched to the timing of per-contract revenue recognition.



# Licensing, Intellectual Property, and Data Economics

## 6.1

### Foundation Model Licensing Arrangements

The contractual relationship between AI application companies and foundation model providers is a critical accounting input. From the application company's perspective as the licensee/customer, these arrangements typically take one of several forms, each with different accounting consequences:

#### API-as-a-service (pay-per-token)

The application company calls the model provider's hosted API on a pay-per-token basis. This is a service contract—no IP license is conveyed, and costs are operating expenses recognized as incurred. This arrangement was discussed in [Section 3.1](#) in the context of COGS classification (when the application company is a principal and reports revenue gross)

#### Model weights licensing (self-hosted)

The application company licenses the right to host and run model weights on its own infrastructure (or cloud instances). The license cost is capitalizable under ASC 350-40 as software "obtained for internal use" (ASC 350-40-15-2), provided the arrangement meets the internal-use software scoping criteria. If the company uses the licensed model to deliver hosted services (SaaS), the license is scoped as internal-use software, with amortization over the license term or useful life, whichever is shorter ([see Section 4.1](#) for the licensing analysis).

#### Hybrid API + fine-tuning access

Arrangements that provide both API access and the ability to fine-tune models combine elements of forms 1 and 2. From the application company's perspective, the key question is whether the fine-tuned model represents a [distinct asset](#) owned by the application company ([see Section 4.1](#) on capitalization). From the model provider's perspective, the arrangement may contain [multiple performance obligations](#) requiring separate allocation.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

**Licensing, Intellectual Property, and Data Economics**

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

## 6.2

# Data Licensing and Content Obligations

AI application companies—particularly those in the search and knowledge archetype—may face **contingent liabilities** from data licensing disputes, copyright claims, or regulatory actions related to the training data used by the underlying models or by the company's own retrieval systems.

Under ASC 450-20 (Loss Contingencies), a **loss contingency** is accrued when it is probable that a loss has been incurred, and the amount can be reasonably estimated. If a loss is reasonably possible but not probable, disclosure is required. While these cases target model providers rather than application layer companies directly, AI application companies face related exposure: potential downstream liability if the foundation models they rely on are found to infringe, indemnification obligations under API terms of service, and direct risk from their own retrieval systems surfacing copyrighted content in AI-generated outputs.

The legal landscape around AI-generated content and training data rights is evolving rapidly. Companies should establish a robust framework for monitoring and assessing these contingencies, with input from both legal counsel and technical accounting advisors.

## 6.3

# Revenue Share Arrangements

Some AI application companies negotiate revenue-sharing terms with foundation model providers or data partners as part of their commercial relationship — for example, a model provider offers favorable API pricing in exchange for a percentage of the application company's revenue. Two accounting questions arise:

- **Is the model provider also a customer?** Under ASC 606-10-32-25 through 32-27, if the application company separately provides goods or services to the model provider (e.g., usage data, co-marketing, or platform integration services), the revenue share may constitute consideration payable to a customer — reducing the transaction price of the arrangement rather than being classified as a cost of revenue. If the model provider is purely a vendor with no reciprocal purchasing relationship, the revenue share is a cost of revenue.
- **Does the revenue share affect the principal-agent analysis?** The existence of a revenue-sharing arrangement does not automatically change the principal-agent conclusion from **Section 2**, but it is a relevant factor. If the revenue share effectively means the application company retains only a thin margin on the model cost component, this may support the agent argument for that component.

Some licensing arrangements include **royalty-based pricing** — for example, the model provider charges X% of the application company's revenue from products using the licensed model. The royalty payments are a cost of the license, recognized as incurred — typically classified within cost of revenue when the application company is a principal. For the less common scenario in which the application company is itself the licensor of proprietary models, the sales- and usage-based royalty exception under ASC 606-10-55-65 through 55-65B applies.



Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

# 7 Financial Statement Presentation and Disclosure

## 7.1 Income Statement Architecture

The AI application layer's Statement of profit and loss looks fundamentally different from traditional SaaS. The following illustrative income statement assumes the AI application company is a principal (gross revenue presentation) with foundation model API costs classified within cost of revenue:

| Line Item                                           | AI App Co. (Illustrative) | Traditional SaaS |
|-----------------------------------------------------|---------------------------|------------------|
| <b>Revenue</b>                                      | \$100M                    | \$100M           |
| <b>Less:</b> Foundation model API / inference costs | (\$35M)                   | —                |
| <b>Less:</b> Hosting & infrastructure               | (\$10M)                   | (\$12M)          |
| <b>Less:</b> Customer support & operations          | (\$5M)                    | (\$5M)           |
| <b>Gross Profit</b>                                 | \$50M (50%)               | \$83M (83%)      |
| R&D expense                                         | (\$40M)                   | (\$25M)          |
| Sales & marketing                                   | (\$25M)                   | (\$30M)          |
| General & administrative                            | (\$10M)                   | (\$10M)          |
| <b>Operating Income (Loss)</b>                      | (\$25M)                   | \$18M            |

### Note

Both columns assume gross revenue presentation. The difference in gross profit is driven by the AI application company's foundation model inference costs in cost of revenue—a cost category that traditional SaaS companies generally do not incur. Traditional SaaS companies may have comparable hosting and infrastructure costs but typically do not face per-interaction variable inference costs of this magnitude. The illustrative comparison is intended to highlight the structural difference in cost profiles; specific fact patterns will vary by company.

The disclosure challenge is helping investors understand that the 50% gross margin is the structural norm for this business model—not a sign of competitive weakness—and that margin expansion will come from cost deflation, routing optimization, and scale.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

**Financial Statement Presentation and Disclosure**

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

## Balance Sheet Considerations

Key balance sheet areas requiring disclosure and judgment include:



### Contract liabilities

Pre-paid token packages, annual subscription prepayments, and enterprise credits create material contract liabilities. Disclosure should include when the entity typically satisfies its performance obligations, the methodology for estimating breakage (*see Section 2.8*), and the balance roll-forward required under ASC 606-10-50-8.



### Contract assets

For consumption-based revenue models, the entity may recognize revenue (upon release of the variable consideration constraint) before invoicing the customer, creating a contract asset (unbilled receivable). Disclosure is required under ASC 606-10-50-4, including the balance roll-forward and an explanation of the significant changes in contract asset balances during the period.



### Deferred contract costs (sales commissions)

Many AI application companies use product-led growth strategies where individual users adopt the product organically (often starting on a free tier), and sales representatives are later assigned to convert the account to an enterprise contract. Commission structures for these conversions may include:

- **Expansion commissions:** Paid when a free-tier or self-serve customer converts to an enterprise contract.
- **Usage-based commissions:** Paid as a percentage of customer consumption above a threshold. These are period costs—recognized as the usage occurs—because they are not incremental costs of obtaining the contract but rather costs of fulfilling it.
- **Renewal commissions at reduced rates:** When renewal commissions are not commensurate with initial commissions, ASC 340-40-35-1 requires that the amortization period for the initial commission include expected renewals — not just the initial contract term — because the renewal commission does not separately compensate for the renewal period.

AI application companies increasingly sell through **cloud marketplace partners, system integrators, and technology alliance partners**. Commission and referral fee structures in these channels differ from direct sales:

- **Marketplace referral fees:** Cloud marketplace operators (AWS, Azure, GCP) charge referral fees (typically 3–20% of the transaction price). These are selling expenses, not commissions subject to ASC 340-40 capitalization—they are costs of the distribution channel, not incremental costs of obtaining the specific customer contract.
- **Co-sell commissions:** When a system integrator or partner receives a commission for introducing the customer, the commission is typically an incremental cost of obtaining the contract.
- **Influence fees:** Fees paid to partners for “influencing” a deal (without the partner being the contracting party) require judgment—are they incremental costs of obtaining the contract, or marketing expenses? The answer depends on whether the fee is contingent on the specific contract being obtained.

AI companies with **enterprise sales teams** typically capitalize incremental costs of obtaining contracts under ASC 340-40-25-1 through 25-2, amortized over the expected period of benefit. The estimated benefit period for capitalized commissions is particularly

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

#### Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

difficult for AI application companies because:

- Contract **renewal patterns** are not yet established (many companies have fewer than two years of enterprise contract history).
- **Customer churn** in AI products may differ significantly from traditional SaaS due to rapid competitive displacement and technology shifts.
- The expected **period of benefit** may not extend to renewals if the renewal commission is commensurate with the initial commission — under ASC 340-40-35-1, the amortization period extends only to the period over which the entity expects to benefit from the cost asset, and a commensurate renewal commission separately compensates for the renewal period.

Companies should document their benefit period estimate with reference to available data (even if limited), comparable industry benchmarks, and the specific characteristics of their customer base. As renewal data accumulates, the estimate should be refined. Controls over this estimate should include quarterly or semi-annual reassessment of the benefit period assumption and comparison of estimated to actual customer lifecycles.

Under DISE (ASU 2024-03), the amortization of deferred commissions flowing through sales and marketing expense will need to be disaggregated within the tabular footnote — likely as employee compensation, depending on the entity's classification policy (*see Section 3.3*).

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

**Financial Statement Presentation and Disclosure**

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help



### Capitalized software costs

Companies capitalizing internal-use software under ASC 350-40 should disclose the nature of capitalized AI assets, useful lives, and amortization methods, and — under ASU 2025-06 (effective for periods beginning after December 15, 2027) — the disclosures required under ASC 360-10 (Property, Plant, and Equipment), including roll-forwards, significant judgments, and impairment information. Under DISE (ASU 2024-03), amortization of capitalized software flowing through cost of revenue or R&D expense will need to be disaggregated as “intangible asset amortization” in the tabular footnote (*see Section 3.3*).



### Contingent liabilities

Data licensing disputes, copyright claims, and evolving AI regulatory obligations may require disclosure or accrual under ASC 450. Disclosure should include the nature of the contingency and an estimate of the possible loss or range of loss — or, if an estimate cannot be made, a statement to that effect (ASC 450-20-50-3 through 50-4). *See Section 6.2* for further discussion.



### Minimum purchase commitments

Committed API spend with foundation model providers should be disclosed as purchase obligations under ASC 440 (Commitments).

Under ASC 440-10-50-2, disclosure of unconditional purchase obligations is required when the obligation meets all of the following criteria: (a) it is noncancelable or contains a penalty for cancellation; (b) it was negotiated as part of arranging financing with the lender or investor for the facilities that will provide the contracted goods or services, or for costs related to those goods or services; and (c) it has a remaining term in excess of one year.

In practice, most AI application companies' API commitments with foundation model providers do not meet criterion (b)—these commitments are negotiated as part of commercial procurement arrangements, not as part of arranging financing for facilities. As a result, the ASC 440-10-50-2 disclosure requirements technically may not apply.

However, for SEC registrants, the disclosure obligation extends beyond ASC 440. Under Regulation S-X, Rule 5-02(24), registrants must disclose information about commitments that are material, regardless of whether they meet the specific criteria of ASC 440-10-50-2. Additionally:

- **Regulation S-K, Item 303 (MD&A):** Known contractual obligations—including material API purchase commitments—should be discussed in the context of the company's liquidity and capital resources analysis. The SEC staff has emphasized that MD&A should address known trends, demands, commitments, events, or uncertainties that are reasonably likely to affect liquidity or results of operations.
- **Regulation S-K, Item 105 (Risk Factors):** Material concentration risk in foundation model provider relationships—including committed spend obligations—should be addressed in risk factor disclosures.
- **Contractual Obligations Table:** Although no longer required by regulation, many companies voluntarily provide a tabular summary of contractual obligations in MD&A. Material API commitments should be included in this table if the company chooses to present one.

Executive Summary

The AI Application Layer: A  
New Software Taxonomy

Revenue Recognition: Key  
ASC 606 Judgments

Cost of Revenue: The New  
Economics of COGS

R&D, Capitalization, and the  
Build vs. Buy Continuum

Subscription vs. Consumption:  
Revenue Model Complexity

Licensing, Intellectual Property,  
and Data Economics

**Financial Statement  
Presentation and Disclosure**

Internal Controls, SOX Readiness,  
and Audit Considerations

IFRS and Ind AS: Comparative  
Notes

How Uniquis Can Help



#### Practical Tip

Even when ASC 440-10-50-2 disclosure requirements are not triggered (because criterion (b) is not met), companies should evaluate whether the commitment is material and requires disclosure under Regulation S-X and S-K. For AI application companies where foundation model API commitments represent a significant portion of future cash outflows, disclosure is generally expected regardless of the ASC 440 scoping outcome. The SEC staff's focus on vendor concentration and supply chain dependencies in the AI space reinforces this expectation.

## 7.3

### IFRS 18 Impact on AI Application Company P&Ls

For IFRS reporters, IFRS 18 (Presentation and Disclosure in Financial Statements)—effective for annual reporting periods beginning **on or after January 1, 2027**—replaces IAS 1 and introduces a restructured income statement with five categories (operating, investing, financing, income taxes, and discontinued operations) and two new mandatory subtotals: operating profit and profit before financing and income taxes. Unlike DISE (which adds a tabular footnote disaggregation without changing the face of the income statement), IFRS 18 restructures the income statement itself – requiring all items to be classified into defined categories. Like DISE, it aims to improve comparability and transparency, but the mechanism is fundamentally different. All foundation model API costs, R&D expenses, and selling costs will reside within the operating category.

The management-defined performance measures (MPM) requirement under IFRS 18 is particularly relevant: AI application companies that communicate non-GAAP measures externally – such as “adjusted EBITDA” or “adjusted gross profit” will need to disclose these as MPMs in the financial statements, provide reconciliations to the most directly comparable IFRS defined subtotal and disclose the tax and non-controlling interest effects of each adjustment. MPM definitions must be applied consistently across reporting periods, with any changes explained.



# 8 Internal Controls, SOX Readiness, and Audit Considerations

AI application companies face a distinctive set of internal control requirements driven by the novelty of their business model and the judgmental nature of their key accounting policies. The controls below are not unique in concept – but their application in the AI context introduces challenges that traditional software companies do not face.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

**Internal Controls, SOX Readiness, and Audit Considerations**

IFRS and Ind AS: Comparative Notes

How Uniquis Can Help

## 8.1

### Controls Over Gross vs. Net Determination

The principal-agent conclusion is a critical accounting judgment that must be supported by a documented analysis, reviewed by qualified personnel, and updated when facts and circumstances change (e.g., when new products are launched, pricing models change, or the degree of proprietary transformation evolves). A management review control over this determination should be a keystone of the company's ICFR framework.

#### What makes this uniquely challenging for AI companies

The degree of proprietary transformation may shift incrementally as the product evolves – adding a new model routing layer, switching foundation model may change the analysis without an obvious triggering event. Controls should include periodic reassessment triggers when contracts change.



## 8.2

### Controls Over Usage Metering and Revenue

For consumption-based revenue models, the accuracy of usage metering (tokens consumed, queries processed, actions completed) directly drives revenue recognition. This creates a control imperative analogous to utility companies' metering controls. Companies must demonstrate that their metering systems are complete, accurate, and reconciled to both the billing system and the foundation model provider's usage reports.

#### What makes this uniquely challenging for AI companies

Metering must capture not just volume (how many tokens) but also attribution (which customer, which product tier, which model) – because pricing, revenue recognition, and COGS allocation may all vary by these dimensions.



## 8.3

### Controls Over API Cost Accruals

Foundation model API costs are billed in arrears, creating a need for month-end accrual estimates. The accrual process must incorporate actual usage data from the billing period, estimated costs for unbilled usage, and reconciliation to provider invoices when received.

#### What makes this uniquely challenging for AI companies

Per-token pricing can change mid-period (model providers have adjusted rates with little or no advance notice), and multi-model routing means the cost per query is not fixed — it depends on which model was selected. Accrual models must account for both pricing variability and routing mix.



## 8.4

### Controls Over Capitalization Decisions

The decision to capitalize or expense AI development costs requires ongoing judgment as the technology evolves. A control framework should include formal project accounting that tracks each development initiative through its lifecycle, documented criteria for the probable-to-complete threshold (under ASU 2025-06), approval workflows for capitalization decisions, and periodic reviews of useful life estimates and impairment indicators.

#### What makes this uniquely challenging for AI companies

The rapid pace of foundation model releases can trigger impairment reviews for capitalized assets that were expected to have a 2-3 year useful life but become functionally obsolete within months.



## 8.5

### Controls Over Variable Consideration and Commission Estimates

For companies with consumption-based or hybrid pricing, the variable consideration constraint analysis ([Section 2.7](#)) and breakage estimation ([Section 2.8](#)) require documented methodologies, cohort data, and periodic reassessment. Similarly, companies capitalizing sales commissions under ASC 340-40 must establish controls over the estimated benefit period — a judgment that is particularly difficult for AI application companies with limited contract renewal history. Controls should include formal documentation of estimation methodologies, approval of key assumptions, and comparison of estimates to actual outcomes as data accumulates.



#### Practical Tip

For AI application companies preparing for an IPO, the four internal control areas that commonly require remediation are: (1) documentation of the gross-vs-net determination with sufficient specificity for audit, (2) reconciliation controls between internal usage metering and foundation model provider billing, and (3) formal capitalization policies that address AI-specific asset categories, and (4) documented methodologies for variable consideration constraint and breakage estimates. Start building these controls 18–24 months before your planned filing date.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

**Internal Controls, SOX Readiness, and Audit Considerations**

IFRS and Ind AS: Comparative Notes

How Uniqus Can Help

# 9 IFRS and Ind AS: Comparative Notes

This Section is retained as a concise comparative reference for companies with multi-jurisdictional reporting obligations. Many AI application companies have global operations requiring dual framework reporting.

## 9.1 Revenue Recognition (IFRS 15)

IFRS 15's principal-agent indicators are broadly aligned with ASC 606, but the IFRS framework places relatively more emphasis on the concept of "primary responsibility for fulfillment" as an indicator of principal status. For AI application companies that bear the customer-facing obligation to deliver the integrated output—regardless of which model generates the underlying inference—this may provide a marginally stronger basis for principal treatment under IFRS than under US GAAP.

## 9.2 Capitalization of Development Costs (IAS 38)

IAS 38 (Intangible Assets) requires capitalization of development costs when six criteria are met, including technical feasibility, intention to complete, and probable future economic benefits. The threshold is lower than US GAAP's ASC 730 research-expense-as-incurred model. This means that IFRS reporters may capitalize more AI development costs than their US GAAP counterparts—particularly fine-tuning costs and proprietary model development. Ind AS 38, which mirrors IAS 38 with limited carve-outs, follows the same approach.

However, the lower capitalization threshold also means greater impairment risk under IAS 36 when capitalized AI assets are rendered obsolete by foundation model advances. The net effect on the P&L may be similar over time, but the period-to-period volatility can differ materially between frameworks.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniqus Can Help

## 9.3

### Onerous Contract Provisions (IAS 37)

Under IFRS, IAS 37 requires recognition of a provision when a contract becomes onerous—i.e., the unavoidable costs of meeting the obligations exceed the economic benefits expected. For AI application companies with minimum API commitments or long-term enterprise contracts where the cost of fulfillment (inference costs) may exceed revenue, the onerous contract assessment is relevant. US GAAP has a narrower onerous contract framework (limited to specific areas like inventory and long-term construction), creating a **potential GAAP difference** for dual reporters (*see Section 3.1* for the US GAAP treatment of committed spend arrangements).

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

**IFRS and Ind AS: Comparative Notes**

How Uniquis Can Help

## 9.4

### Ind AS Considerations for Indian AI Companies

India's AI startup ecosystem is growing rapidly, with several companies building application-layer products for both domestic and global markets. Ind AS 115 (Revenue from Contracts with Customers) is converged with IFRS 15, so the principal-agent analysis follows the same framework. However, Indian companies must also consider transfer pricing implications of cross-border API costs (payments to US-based model providers), indirect tax treatment of AI services, and local securities regulator disclosure expectations for companies contemplating IPO.



# How Uniquus Can Help

Uniquus Consultech brings specialized, conflict-free advisory expertise to the unique accounting challenges of the AI application layer:

## Revenue Recognition Advisory

Gross-vs-net determination, performance obligation analysis, variable consideration estimation, breakage policies, and ASC 606 memo documentation for AI-native business models.

## IPO & Capital Markets Readiness

S-1 accounting policy design, SEC comment letter preparation, historical financial restatement, EGC election strategy, and SOX readiness programs for AI companies.

## DISE & IFRS 18 Implementation

Expense disaggregation mapping, system configuration, internal control design, and dual-framework planning for AI application companies preparing for new presentation standards.

## Multi-Jurisdictional Reporting

Tri-framework expertise (US GAAP, IFRS, Ind AS) for AI companies with global operations, cross-border API arrangements, and multi-market compliance requirements.

Executive Summary

The AI Application Layer: A New Software Taxonomy

Revenue Recognition: Key ASC 606 Judgments

Cost of Revenue: The New Economics of COGS

R&D, Capitalization, and the Build vs. Buy Continuum

Subscription vs. Consumption: Revenue Model Complexity

Licensing, Intellectual Property, and Data Economics

Financial Statement Presentation and Disclosure

Internal Controls, SOX Readiness, and Audit Considerations

IFRS and Ind AS: Comparative Notes

How Uniquus Can Help

## About Uniquis Consultech:

Uniquis Consultech is a global tech-enabled consulting company that specializes in Accounting & Reporting, Governance, Risk & Compliance, Sustainability & Climate, Tech Consulting, and Valuations. The Company is co-founded by consulting veterans Jamil Khatri and Sandip Khetan and backed by marquee investors such as Nexus Venture Partners, Sorin Investments, and UST.

Uniquis has a global team of 800+ professionals led by 100+ Partners & Directors across eleven offices in the USA, the Middle East, and India. The company serves more than 300 clients, including marquee names in each of the markets it operates in.

Uniquis is committed to leveraging technology and an integrated global delivery model to provide best-in-class consulting services to its clients.

## About Our Accounting & Reporting Consulting Services:

In today's dynamic business environment, organizations are continuously required to adapt to the dynamic shifts in accounting & reporting standards and regulations. Enhanced management reporting requirements and accelerated reporting timelines necessitate increased automation within the finance function. Special events, such as preparing for an IPO/capital market transaction or fulfilling the reporting requirements arising from an M&A transaction, introduce additional complexities and challenges for the finance function.

As a result, organizations need to transform their finance functions from time to time with a focus on people, processes, and technology. Our Accounting & Reporting Consulting (ARC) practice is designed to partner with the finance function through these challenges. Our teams function as an extension of your finance teams to execute your plans seamlessly.

We bring the necessary technical expertise, skills, technology, and bandwidth, enabling you to navigate your plans in real time without adding additional manpower costs within the organization.

## A Team That You Can Trust To Deliver



**Sandip Khetan**  
*Co-Founder, Global Head  
of Accounting & Reporting  
Consulting*



**Arda Kaya**  
*Partner, Accounting &  
Reporting Consulting*



**Avinash Ramkumar**  
*Partner, Accounting &  
Reporting Consulting*



**Sharad Chaudhry**  
*Partner, Accounting &  
Reporting Consulting*



**Puneet Chowdhary**  
*Associate Partner, Accounting &  
Reporting Consulting*



**Harsimran Singh Sangha**  
*Managing Director, Accounting  
& Reporting Consulting*



**Shoichi Ohno**  
*Managing Director, Accounting  
& Reporting Consulting*



**Shival Sehgal**  
*Managing Director, Accounting  
& Reporting Consulting*

Scan this code to find  
more content like this



### Disclaimer

This publication is provided for informational purposes only and does not constitute professional accounting, legal, or tax advice. The views and opinions expressed are those of the author and Uniquis Consultech and do not necessarily reflect the official positions of any regulatory body, standards-setter, or audit firm. Readers should consult with their own professional advisors before making decisions based on the information contained herein. All references to accounting standards, regulations, and market data are current as of the date of publication and are subject to change.

Visit us at [www.uniquis.com](http://www.uniquis.com)

or follow us on

