

A Commodity-Indexed Pricing Framework for Autonomous AI Agents

Abha Dalmia

`abha.dalmia1@gmail.com`

Cloudgin.ai

Research Article

Keywords: AI agent pricing, SaaS monetization, usage-based pricing, outcome-based pricing, inference economics, platform economics

Posted Date: June 25th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9968130/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

A Commodity-Indexed Pricing Framework for Autonomous AI Agents

Lessons from Long-Term Energy Contracts for Agentic SaaS

[Author names and affiliations omitted for anonymous peer review]

Abstract

AI agents are breaking the SaaS business model. When a single agent replaces many users, per-seat licensing revenue falls while customer value rises. Providers need a new pricing approach. Cost-plus pricing causes customers to limit usage. Pure outcome pricing transfers all inference cost risk to the provider. We propose a three-term framework adapted from long-term Liquefied Natural Gas contracts: $P = \sum(\alpha_i \cdot C_i) + \beta \cdot V + \gamma$. This structure separates infrastructure cost pass-through, value capture, and a platform floor. We introduce the Inference Capture Ratio (ICR) as a health metric, and use Salesforce, Microsoft, Intercom, and Zendesk pricing evolutions to show how the market is converging on this structure.

Keywords: AI agent pricing, SaaS monetization, usage-based pricing, outcome-based pricing, inference economics, platform economics

Introduction

A SaaS company sells software to an enterprise customer for \$150 per seat per month across 200 agents. Annual revenue from that account is \$360,000. The company then releases an AI agent that resolves 80% of tickets autonomously. The customer reduces headcount by 160 and cancels

those licenses. Revenue from the same account drops to \$72,000. The company built the product that destroyed its own revenue.

This is not a hypothetical. It describes what is happening across SaaS categories as AI agents become operationally mature. Legal tech companies are automating document review that paralegals used to perform. Financial services platforms are automating reconciliation work that analysts used to own. Customer experience tools are resolving tickets that human agents used to handle. Healthcare administration platforms are processing prior authorizations that coordinators used to process manually. In every case, the seat count falls faster than pricing models can adapt.

The irony is sharp. SaaS companies have spent a decade building AI capabilities that, once deployed, will systematically eliminate the users those capabilities were priced around. Most SaaS boards have not yet modeled what their ARR looks like if per-seat revenue declines 30 to 50% over three years. That analysis produces uncomfortable numbers. It also produces urgency around pricing strategy that was not present two years ago.

The standard advice is to move from cost-based to value-based pricing^[1]. We think this advice is directionally right but incomplete. Pure value pricing solves the headcount problem but creates a new one: it shifts all inference cost risk to the provider. If an AI agent needs 125 attempts to produce 100 successful outcomes, the provider absorbs the cost of 25 failures. If success rates drop from 80% to 70% during a model transition, provider cost goes up 14% while revenue stays flat. At scale this is an insolvency risk, not a pricing inefficiency.

We propose a solution borrowed from an unexpected source: the long-term contracts that govern Liquefied Natural Gas (LNG) trade. The LNG industry has been solving a pricing problem with the same structure as AI inference for four decades. Producers need protection from commodity price volatility. Buyers need predictable costs. Both need contract stability over long horizons.

The industry solved this with linear formulas indexed to oil prices plus a take-or-pay floor. We adapt that structure to AI pricing.

Our framework has three terms: $P = \sum_i(\alpha_i \cdot C_i) + \beta \cdot V + \gamma$. The first term covers infrastructure cost across multiple inference providers. The second captures a share of outcome value. The third is a platform floor. The multi-provider decomposition in the first term is important and new. Real agentic systems draw inference from multiple providers simultaneously. A single blended coefficient hides cost structure and produces pricing errors when any one provider changes rates. We also introduce the *Inference Capture Ratio* (ICR), defined as $\beta \cdot V / (\sum_i(\alpha_i \cdot C_i) + \gamma)$, as a single forward-looking health metric with a target threshold of $ICR \geq 2.5x$.

The article proceeds as follows. Section 2 covers why existing pricing models fail. Section 3 develops the LNG analogy. Section 4 states the framework and ICR metric, with Figure 1 showing the three-component structure. Section 5 covers the infrastructure needed to make the framework operational. Section 6 presents four case studies showing market convergence. Section 7 addresses risk and implementation. Section 8 discusses limitations and boundary conditions. Section 9 discusses broader implications.

2. Why Existing AI Pricing Models Fail

The pricing literature tells us to price based on value delivered, not cost incurred^[2]. This principle is correct. The problem is that AI inference has a cost structure that is high, volatile, and correlated with the outcomes pricing is supposed to capture. Three failure modes result.

2.1 Per-Seat Pricing

Per-seat pricing worked because software value used to be proportional to human adoption. More seats meant more users, which meant more organizational processes automated, which meant

more value delivered. The model was self-regulating. Customers who extracted more value naturally had more users, and therefore paid more. Enterprise procurement teams could forecast software cost as a function of headcount, which was itself planned and budgeted. The whole financial architecture of SaaS rested on this proportionality.

Autonomous agents break this link. One well-configured agent can do the work of many human users. Seat count drops, value rises, revenue drops. There is no corrective mechanism. Providers who build better agents get punished by their own pricing model. This is a rare incentive structure in commercial software: excellence actively reduces revenue. The longer a provider delays releasing an effective agent, the longer seat revenue holds. The better the agent performs, the faster seat revenue collapses. No rational management team wants to be in this position, yet most SaaS companies building agentic features are in exactly this position today.

2.2 Cost-Plus Pricing

Cost-plus pricing computes token costs, infrastructure amortization, and API fees, then adds a margin. It solves provider solvency but creates what we call the *meter-watching problem*: customers who can see per-query costs in real time cut usage to avoid bill shock, regardless of the return on each query^[3]. This happens because finance teams receiving AI invoices usually lack the attribution infrastructure to tie spend to outcomes. When they cannot see the return, they treat all AI cost as uncontrolled overhead.

Cost-plus also misses willingness-to-pay entirely. A customer's marginal willingness to pay for a resolved ticket depends on the value of that outcome, not on the token cost of producing it.

Cost-plus captures cost but leaves value uncaptured. It is structurally incomplete.

2.3 Pure Outcome Pricing

Outcome pricing is often called the ideal model. Providers charge only for successful outcomes. Customers bear no cost risk. Adoption barriers fall. Most AI vendors cite this as their target state.

The problem is the risk structure. At an 80% success rate, producing 100 successful outcomes requires 125 attempts. The cost of 25 failures sits with the provider. If success rates drop from 80% to 70% during a model transition or distribution shift, provider costs rise 14% while revenue stays flat. This is not a small variance. It is enough to push a thin-margin AI business into losses within a single quarter.

The tail risk is worse than the average case. A *runaway agent* scenario, in which an AI system enters a recursive loop or a prompt injection triggers repeated expensive calls, turns an acceptable loss rate into a catastrophic one. Pure outcome pricing is unlimited provider liability for customer-side misconfigurations. No provider should accept this structure at scale.

No existing model satisfies all three requirements at once: protecting provider costs, capturing outcome value, and giving both sides a predictable revenue baseline. The LNG industry solved a structurally similar problem in the 1970s.

3. The LNG Pricing Analogy

3.1 The LNG Contract Problem

LNG trades under contracts that run 15 to 25 years^[4]. Contracts are this long because the infrastructure (liquefaction plants, cryogenic tankers, regasification terminals) costs billions per project. Neither the producer nor the buyer can recover this capital without a committed long-term relationship.

But long contracts create a new problem. The underlying commodity price is volatile. Natural gas prices can move 50% within a year. Oil prices, which are the traditional index, move

similarly. If a 20-year contract fixes price at spot, one side will be badly disadvantaged within a decade. The industry solved this with a linear formula indexed to Brent Crude oil, with three parts: a slope coefficient that controls how much oil price volatility passes through to gas price, a premium for delivery infrastructure, and a take-or-pay floor that guarantees minimum revenue regardless of actual offtake.

The slope coefficient is the most instructive part. If the slope is zero, the buyer pays a fixed price regardless of oil and bears no commodity risk; the producer absorbs all volatility. If the slope is one, the buyer pays proportional to oil and bears all commodity risk. In practice slopes run between 0.10 and 0.15, which means roughly 10 to 15% of oil price volatility passes through to the gas price. This calibration is not arbitrary. It reflects decades of bilateral negotiation over how much commodity risk each side is willing to bear, and it is revisited at price review points every three to five years.

The formula is: $P_{\text{LNG}} = S \cdot (\alpha \cdot \text{Brent}) + \beta_{\text{delivery}} + \gamma_{\text{ToP}}$. It has governed trillion-dollar bilateral trades for four decades without systematic insolvency on either side.

3.2 Why This Maps to AI Inference

The structure is the same. AI providers face volatile commodity costs (LLM API prices change on monthly timescales), high fixed capital (GPU clusters, orchestration, reliability engineering), and outcome-correlated cost exposure.

LLM token prices are the Brent Crude of AI inference. They are the upstream commodity whose cost the provider cannot control and should not fully absorb. The α coefficient plays the role of the LNG slope: it sets how much upstream volatility passes to the customer and how much the provider absorbs. The γ constant is the take-or-pay commitment: the minimum monthly revenue

that justifies the provider's fixed infrastructure. The $\beta \cdot V$ term is new, not present in LNG contracts, and captures the outcome value that makes AI worth paying for above raw cost.

LNG Contract Term	Energy Market Meaning	AI Inference Analog
Brent Crude index	Volatile upstream commodity price	LLM token price (Anthropic, OpenAI, Google)
Slope coefficient S	Share of commodity risk passed to buyer	α_i , the per-provider infrastructure coefficient
Delivery premium	Liquefaction, shipping, regasification capital	Orchestration, latency SLA, redundancy
Take-or-pay floor	Minimum revenue regardless of offtake	Platform subscription, the committed ARR baseline
S-curve kink	Volume discount above threshold	Circuit breaker ceiling plus volume tier discount

Table 1. Structural mapping between LNG contract terms and the proposed AI pricing framework.

3.3 The Multi-Provider Extension

The LNG analogy on its own would suggest a single α coefficient. That works for LNG because each cargo has one origin. But a real agentic AI system uses multiple inference providers at once, each with different cost structures and pricing trajectories.

A production customer service agent might use one model for long-context reasoning, another for multimodal inputs when a customer sends a screenshot, a voice API for phone channels, and an embedding model for retrieval. Each has its own unit cost and its own price volatility. Using a single blended α hides this structure. When one provider cuts prices (which happens regularly in the LLM market), a blended coefficient does not reflect the actual cost reduction, and the provider either over-prices or under-prices relative to the intended margin.

The cleaner approach is to decompose α by provider, maintain separate pass-through coefficients, and aggregate at the bill level. AWS, Google Cloud, Azure, and Twilio all publish

per-service unit costs this way. Customers compose these into higher-level workflow costs. The pattern works because costs are transparent at the component level and outcomes are measured at the composed level.

4. The Framework and the Inference Capture Ratio

4.1 The Framework

For an AI-augmented SaaS product using autonomous agents across $n \geq 1$ inference providers in a billing cycle T , we propose:

$$P = \sum_{i=1}^n (\alpha_i \cdot C_i) + \beta \cdot V + \gamma$$

Where $\alpha_i \in (0,1)$ is the infrastructure coefficient for provider i , set to that provider's current API cost; C_i is the total compute consumption from provider i in cycle T (tokens, API calls, or minutes); $\beta \in (0,1)$ is the value multiplier, set per vertical and outcome type; V is the quantified outcome value generated in cycle T in the customer's currency; and $\gamma > 0$ is the platform constant, set to cover fixed costs.

Under these definitions, $P \geq \sum_i (\alpha_i \cdot C_i)$ always holds. Provider solvency on the infrastructure layer is guaranteed by construction, not by outcome performance. This is the same property that makes LNG contracts robust. The producer's recovery of commodity and capital cost does not depend on how much gas the buyer lifts.

Figure 1 shows how the three components combine. Each term serves a distinct economic function, and each has a direct LNG analog except the value-capture term $\beta \cdot V$, which is specific to AI pricing and has no equivalent in commodity contracts.

INFRASTRUCTURE $\Sigma_i (\alpha_i \cdot C_i)$	VALUE CAPTURE $\beta \cdot V$	PLATFORM FLOOR γ
Cost pass-through for each inference provider <i>Anthropic, OpenAI, Google, Twilio</i>	Share of outcome value delivered to customer <i>Resolved tickets, leads, claims</i>	Committed minimum revenue (take-or-pay) <i>Covers fixed R&D, SLA, support</i>
Customer bears commodity risk	Shared upside	Customer bears capacity risk
<i>LNG analog:</i> S · Brent slope	<i>LNG analog:</i> (none, AI-specific)	<i>LNG analog:</i> Take-or-pay floor

Figure 1. The three components of the proposed pricing framework. The infrastructure and platform-floor terms map directly to LNG contract mechanics. The value-capture term is specific to AI and has no LNG analog.

4.2 The Inference Capture Ratio

The framework suggests a natural health metric for AI pricing. We call it the Inference Capture Ratio (ICR):

$$\text{ICR} = \beta \cdot V / (\Sigma_i (\alpha_i \cdot C_i) + \gamma)$$

ICR measures the ratio of outcome-linked revenue to combined infrastructure and platform cost. An ICR of 1.0 is break-even on AI delivery. An ICR of 2.5x means the provider captures \$2.50 in revenue for every \$1.00 of combined cost. We suggest 2.5x as the minimum threshold for healthy AI monetization, based on observed SaaS gross margin structures where sustainable businesses need roughly 60 to 70% gross margin.

A worked example helps. Suppose a provider runs a customer service agent for an enterprise customer. Monthly inference cost across providers is \$8,000 (the $\alpha \cdot C$ term). Platform cost allocation is \$2,000 (the γ term). The agent resolves 5,000 tickets at an estimated outcome value of \$25 per ticket, or \$125,000 in monthly value created. If the contract sets β at 0.20, outcome

revenue is \$25,000. Total revenue is \$27,000 ($\beta \cdot V$ plus γ once). ICR is \$25,000 divided by \$10,000, or 2.5x. This is at the target threshold. If the customer's business scales and ticket volume doubles, $\alpha \cdot C$ grows linearly but outcome revenue grows linearly too, so ICR stays stable. If the model becomes more efficient and inference cost drops 20%, ICR rises to 2.78x, and the provider can share some of that efficiency with the customer through an S-curve discount while preserving margin.

ICR has one property that Net Revenue Retention lacks and that makes it more useful as a board-level metric. ICR is predictable from inputs. A provider that knows its α_i values, its γ , and its β can project ICR for any expected V and C combination. This makes it a planning tool, not just a reporting metric.

4.3 Calibrating β by Vertical

The value multiplier β is the hardest coefficient to set correctly, and it is also where most of the negotiation effort should concentrate. Different verticals have different relationships between AI-delivered work and business value, and a single β does not work across the portfolio.

Customer service resolution is the easiest case. A resolved ticket has a quantifiable value (avoided human handling cost, customer retention impact, reduced escalation risk). If that value is \$25 per ticket and the provider wants to capture 20% of it, β is 0.20. Sales lead qualification is harder. The value of a qualified lead depends on downstream conversion rates, which vary by product, by sales team, and by season. A lead-qualification agent might justify β of 0.10 to 0.30 depending on these factors. Contract review is harder still. The value of an AI-reviewed contract is a function of risk avoidance, which is statistical and hard to quantify per contract.

The practical approach is to set β with a prior band and refine it with observed data. Start at a conservative β that is clearly below the customer's willingness-to-pay ceiling. Measure outcomes for two billing cycles. Adjust β upward if ICR remains above 3.0x and customer satisfaction is stable, or adjust downward if renewal risk emerges. The structure should feel familiar to anyone who has negotiated commodity slope coefficients. It is the same iterative calibration, applied to a different variable.

4.4 Comparison to Existing Approaches

Table 3 compares the four main pricing models on provider risk, customer experience, margin stability, and achievable ICR.

Model	Provider Risk	Customer UX	Margin Stability	ICR
Per-seat	Low	Predictable	Eroding as AI scales	< 1.0x
Cost-plus	Medium	Meter-watching	Moderate	1.0 to 1.8x
Pure outcome	Critical	Excellent	Volatile / loss risk	Unbounded
Proposed framework	Shared (floor)	Excellent	Protected	$\geq 2.5x$

Table 2. Comparison of AI pricing models on four criteria.

ICR Range	Diagnosis	Recommended Response
-----------	-----------	----------------------

< 1.0x	Insolvent on AI delivery	Restructure α floor; audit attribution
1.0 to 1.8x	Marginal, meter-watching zone	Improve outcome attribution; renegotiate β
1.8 to 2.5x	Developing, viable but fragile	Decompose multi-provider α ; enforce γ floor
$\geq 2.5x$	Healthy, scalable monetization	Expand scope; deploy S-curve volume tiers
$\geq 4.0x$	Excellent, market-leading capture	Share upside with customers; lock in multi-year deals

Table 3. ICR thresholds for AI-augmented SaaS providers.

5. Operationalizing the Framework

A pricing framework that cannot be metered cannot be billed. This section covers the infrastructure needed to make the framework work in production.

5.1 Multi-Provider Metering

The hardest technical requirement is computing $\sum_i(\alpha_i \cdot C_i)$ across providers with sub-cent precision in real time. Traditional SaaS billing runs as a batch process at month-end. That works when unit costs are stable (fixed per seat) and usage is sparse. It does not work for AI inference, where the event stream runs at millions of events per hour, unit costs vary by provider and model version, and pricing errors compound quickly.

The architecture needs four properties. First, asynchronous event capture, so billing logging adds no more than a few milliseconds to agent response latency. Second, provider-level attribution on every event: provider ID, model version, input and output token counts, and timestamp, so $\alpha_i \cdot C_i$ can be computed later without reprocessing. Third, an immutable ledger of attributed cost events, since β -weighted success fees require audit trails. Fourth, real-time aggregation so running totals of $\sum_i(\alpha_i \cdot C_i)$ and $\beta \cdot V$ are available with sub-minute latency, both for the circuit breaker (Section 7) and for financial dashboards.

5.2 The Financial Pane of Glass

Most AI deployments do not fail for technical reasons. They fail because CFOs see a variable cost line they cannot predict, cannot attribute, and cannot explain, and they kill the deployment. This is the right decision given the information they have. The solution is not to simplify the pricing model until it looks like a seat license. That would defeat the purpose. The solution is to build the financial visibility infrastructure that makes the new model legible.

We call this layer the *financial pane of glass*. It is a dashboard that presents the three framework components in financial management terms, not engineering terms. Engineers see tokens, API calls, and latency. CFOs need OPEX, margin, and ROI. The gap between these views is not just naming. It is a real translation problem that requires purpose-built mapping. In practice that means converting $\alpha_i \cdot C_i$ events into cost center allocations, $\beta \cdot V$ outcomes into revenue recognition line items, and γ commitments into fixed OPEX forecasts. The result is a P&L where the AI product appears as a normal business unit with identifiable revenue, cost, and margin.

6. How the Market is Converging

We now turn to empirical evidence. The primary case is Salesforce Agentforce's pricing evolution from 2024 to 2026. We then examine three other cases (Microsoft, Intercom, Zendesk) that are at different points on similar trajectories.

6.1 Salesforce Agentforce, 2024 to 2026

Salesforce's pricing trajectory is useful because it is exogenous to our framework. Salesforce made pricing decisions under market pressure, without prior theory, and without knowledge of the LNG analogy. The trajectory still traced the path our framework would predict, moving from failure mode to failure mode to the convergent structure^[5]. Figure 2 summarizes the trajectory.

2024 LAUNCH \$2 / conversation Pure β	2025 PIVOT \$0.10 / action Pure α	2026 CURRENT Flex Agreement $\alpha + \beta + \gamma$
Outcome only, no floor	Usage only, no outcome capture	All three components aligned
✗ CFO rejection <i>unpredictable invoices</i>	✗ Meter-watching <i>deployment self-limited</i>	✓ CFO-compatible <i>committed ARR + upside</i>

Figure 2. Salesforce Agentforce's three-phase pricing evolution. Each phase tested a single-component model (pure β , then pure α) before converging on the three-component structure in the 2026 Flex Agreement.

The 2024 launch at \$2 per conversation was pure β pricing. No floor, no infrastructure recovery. CFO rejection was fast and predictable. A per-conversation price produces monthly invoices that swing with volume, and enterprise finance teams cannot plan quarterly OPEX around that kind of variability. Salesforce's sales motion also struggled because account executives could not tell customers what the monthly cost would be. The answer depended on conversation volume, which customers could not forecast either. Deal cycles stretched. Pilots stayed small. Expansion was limited.

The 2025 Flex Credit pivot at \$0.10 per action was pure α pricing. It was usage-based, granular, with no outcome capture. It solved the variability problem by giving customers a meter they could read. It created the meter-watching problem the same way. Customers capped usage to stay within mental budgets, regardless of the return the capped actions would have generated. One Salesforce customer described the experience as watching a taxi meter run during every AI interaction, which is not the user experience that produces broad adoption.

The 2026 Flex Agreement lets enterprises convert unused human seat commitments into AI agent capacity credits, with a committed baseline (γ) and flexible consumption above it (α and β). All three components are present. The seat-to-agent conversion is structurally a take-or-pay

swap. Customers commit to a minimum baseline, previously counted in seats, now in agent capacity, and get flexibility to redirect that commitment to higher-value work. This structure removes the adoption friction of pure usage pricing while preserving the ARR predictability enterprise finance teams need. It took Salesforce 24 months to arrive at this structure through market pressure. Our argument is that providers starting from the framework can get there on a much shorter timeline.

6.2 Microsoft, Intercom, and Zendesk

Three other cases show partial convergence at different stages.

Microsoft Copilot. Microsoft launched 365 Copilot at \$30 per user per month in 2024. This is per-seat pricing applied to agentic functionality, which is the model our framework predicts will erode fastest as agent capability scales^[6]. By 2025 Microsoft had added Copilot Studio consumption-based billing alongside the per-seat license. This is an implicit admission that agent-driven work cannot be priced by headcount. The trajectory is mid-evolution, with a γ floor (the seat license) and an emerging α component (consumption billing), but no β yet.

Intercom Fin. Intercom launched Fin at \$0.99 per resolution. This is pure β pricing. Intercom reports strong adoption, but the structural vulnerability from Section 2.3 is still there. If inference cost per attempt rises while resolution rates fall, unit economics compress fast. Intercom has publicly discussed adding fair-use policies^[7], which is an implicit admission that pure β pricing needs a de facto α protection layer at scale.

Zendesk. Zendesk's 2024 Advanced AI add-on was a flat \$50 per agent per month. This is per-seat again. By late 2025 Zendesk had added resolution-based pricing for autonomous agents

alongside the per-seat components^[8], which puts them closer to the $\alpha + \beta + \gamma$ structure. Zendesk's bundled agreements also increasingly include minimum commitment floors that function as γ .

The pattern is the same across all four cases. Providers start at a single-component model, hit either CFO resistance or margin compression, and add the missing components piece by piece. None of them coordinated. Each discovered the structural gap separately. Our view is that providers who start with the framework can get to the convergent structure faster, with less revenue volatility, and with the right metering infrastructure in place from day one.

7. Risk and Implementation

7.1 The Computational Circuit Breaker

The tail risk under the framework is the runaway-agent event from Section 2.3. Under $\alpha \cdot C$, this creates large cost exposure in a window shorter than any human review cycle. The solution is an automated circuit breaker in the metering layer, not the application layer. Application-layer rate limits are bypassable and do not have cost visibility. A metering-layer breaker has direct access to the running $\sum_i (\alpha_i \cdot C_i)$ total and can trigger on statistical deviation from expected cost patterns. A simple rule works: halt if hourly $\alpha \cdot C$ exceeds the running mean by three standard deviations.

The calibration matters. Setting the threshold too tight produces false positives that disrupt normal operations and train customers to ignore alerts. Setting it too loose fails to catch real runaway events. In practice, the threshold should be calibrated separately per customer, based on their historical usage distribution, and should adapt to diurnal and weekly patterns. A customer whose usage runs 1,000 inference calls per hour on weekdays and 200 per hour on weekends should have different thresholds in those windows. The metering system needs to maintain enough history (typically 30 to 60 days) to compute these adaptive thresholds reliably.

When triggered, the breaker suspends the agent, sends alerts to engineering (anomalous usage) and finance (projected bill impact), and requires explicit re-authorization to resume. This turns a black swan billing scenario into a known-maximum billing scenario. The CFO can model worst-case monthly cost as a function of the breaker threshold, not as a function of unbounded agent behavior.

7.2 S-Curve Volume Pricing

Providers with volume commitments from LLM vendors see actual costs per token fall below the contract α . The framework should reflect this through a volume-dependent modification that shares the discount with the customer while keeping a positive floor. LNG contracts use an identical mechanism through S-curve pricing bands, where the linear slope flattens above and below certain Brent thresholds to limit bilateral exposure. The mechanism creates a natural volume incentive and prevents the provider from pricing below cost at any volume.

7.3 Implementation Sequencing

Adoption needs three workstreams running in parallel.

The first is coefficient calibration. This means historical inference cost attribution by provider and model version to set baseline α_i values, outcome attribution analysis linking AI actions to measurable business outcomes (resolution rates, lead conversion, CSAT) to set β , and fixed cost accounting to set γ . This is not a modeling exercise. It requires production data. Most providers discover at this stage that their outcome attribution infrastructure is missing.

The second is sales motion transformation. Sales compensation tied to seat counts creates internal resistance to any model that replaces seats with agent capacity. The change is structural: commission moves to γ floor value (committed baseline ARR) and $\beta \cdot V$ expansion (outcome

value growth), not seat count. Sales engineers also need to be able to present ICR projections to buyers, showing that an ICR of 3.0x means \$3.00 in delivered value for every \$1.00 of total cost, instead of walking through token-cost spreadsheets that just invite meter-watching.

The third is continuous recalibration. LLM API pricing changes on six to twelve month cycles with 20 to 50% price movements. Keeping margins intact requires quarterly α_i recalibration.

Outcome attribution infrastructure also matures over time, which enables more granular β measurement and vertical-specific β . ICR should be reviewed at every board meeting as the primary AI health metric.

8. Limitations and Boundary Conditions

The framework has three boundary conditions worth stating clearly.

First, it assumes outcome value V is measurable. For some agentic applications this is easy (resolved tickets, qualified leads, approved claims). For others it is harder (creative work, complex judgment tasks, strategic analysis). Where V is unmeasurable, the $\beta \cdot V$ term collapses and the framework reduces to a two-component structure ($\alpha + \gamma$) that provides provider solvency but not full value capture. This is a meaningful limitation for providers in those verticals. A partial answer is to use proxy metrics (user engagement, time saved, downstream conversion) as V , but proxies are less defensible than direct outcome measurement and should be adjusted through lower β values to account for attribution uncertainty.

Second, the framework assumes a bilateral relationship between provider and customer. In multi-sided platforms where a provider mediates between multiple customer groups, the framework needs extension. The α , β , and γ terms may need to be computed separately for each

side of the platform, with the understanding that value capture and cost pass-through run through different economic channels. We leave this extension for future work.

Third, the framework applies to providers with direct contractual relationships to inference suppliers. Providers running open-source models on owned infrastructure face a different cost structure, with lower variable cost but higher fixed capital. For them, the α_i coefficients should reflect amortized compute cost plus electricity, not API pricing. The formula still applies, but the calibration differs.

A fourth consideration, though not strictly a limitation, concerns regulatory environments. AI pricing that captures a share of outcome value may trigger regulatory scrutiny in certain verticals, particularly healthcare, financial services, and legal services, where outcome-linked compensation structures are sometimes restricted. Providers in these verticals need to confirm that $\beta \cdot V$ revenue is structured as a service fee rather than a contingent payment that could be mischaracterized as fee-splitting or regulatory arbitrage.

A fifth concern is measurement integrity. The framework relies on accurate reporting of both C (compute consumption) and V (outcome value). If either is misreported (by the provider, by the customer, or by a third-party attribution service), the framework produces wrong prices. LNG contracts address this with independent meter reading, third-party assays, and dispute resolution clauses. AI contracts need analogous provisions: independent logging of inference events, audit rights over outcome attribution methodology, and a defined process for resolving disputes about outcome counts. Providers should assume that audit rights will be exercised and build their metering infrastructure to a forensic standard from day one.

Finally, the framework assumes a relatively stable technology baseline. If a disruptive shift in inference technology (for example, much cheaper locally-hosted models, or a new class of

specialized AI hardware) dramatically changes the α_i landscape, contracts will need renegotiation. This is not a flaw of the framework. LNG contracts have price review points specifically to handle analogous shifts in the gas market. AI contracts should include similar provisions.

9. Broader Implications

The arithmetic of AI automation applied to the U.S. software sector is stark. U.S. software revenues are roughly \$850 billion annually^[9]. Per-seat licensing accounts for a substantial share. Credible estimates suggest generative AI could automate 20 to 30% of current knowledge work within the decade^[10]. If per-seat licensing stays dominant, that automation implies \$150 to \$250 billion in annual software revenue at risk, destroyed not by competitive failure but by the success of the technology the software companies themselves built.

Our framework is a structural hedge. Decoupling software revenue from headcount and linking it to the value of AI-delivered work lets providers maintain, and probably grow, revenue even as the human workforce they serve becomes more agent-augmented. The γ floor provides ARR stability through the transition. The $\beta \cdot V$ term provides revenue growth proportional to expanding AI scope.

A secondary implication is for capacity planning. Treating inference as a commodity, with the same pricing discipline applied to LNG, produces the billing data needed for aggregate forecasting of AI compute demand, energy consumption, and infrastructure investment. The multi-provider α_i decomposition, aggregated across providers and customers, is the right data structure for answering sector-level capacity questions. These are infrastructure planning

questions, and they do not have answers today because the pricing models in use do not generate the necessary data.

10. Conclusion

Our core argument is that the AI inference pricing problem has the same structure as the long-term LNG contract problem, and that the solution the energy industry developed over four decades applies directly to agentic software. The framework we propose is not an analogy used for effect. It is a structural mapping with specific consequences for how SaaS providers should price, meter, and report AI-delivered value.

Three contributions come out of this. The three-term formula gives a pricing structure that protects provider cost, captures outcome value, and produces CFO-legible forecasts at the same time. The multi-provider α decomposition extends the formula to the architecture of real enterprise systems, where inference cost sits across multiple providers. The Inference Capture Ratio gives a single forward-looking metric, analogous to Net Revenue Retention in traditional SaaS, that boards and CFOs can use to benchmark AI monetization health.

The Salesforce trajectory, along with the partial trajectories at Microsoft, Intercom, and Zendesk, suggests the market is converging on this structure independently. Providers who understand why that convergence is happening can deploy it faster, calibrate it more precisely, and build the metering and attribution infrastructure that makes it commercially robust.

One last point worth stating. The LNG industry did not invent long-term commodity-indexed contracts because its practitioners were theoretically sophisticated. They did it because they were operationally exposed. Capital at risk was too large and price volatility too severe for simpler contracts to hold. The AI software industry is approaching a comparable inflection point. GPU

infrastructure capital, LLM API price volatility, and agentic deployment scale are all moving toward a regime where per-seat and pure-outcome pricing fail structurally. This paper is an attempt to offer the response before that threshold is crossed, not after.

Declarations

Funding: No funding was received for conducting this study.

Competing interests: The author has no competing interests to declare that are relevant to the content of this article.

Ethics declaration: not applicable.

Clinical trial number: not applicable.

Notes

1. Hinterhuber, A. “Is Innovation in Pricing Your Next Source of Competitive Advantage?” *Business Horizons* 57, no. 3 (2014): 413 to 423.
2. Nagle, T. T., and G. Müller. *The Strategy and Tactics of Pricing*, 6th ed. (New York: Routledge, 2017). See also Lambrecht, A., and B. Skiera. “Paying Too Much and Being Happy About It.” *Journal of Marketing Research* 43, no. 2 (2006): 212 to 223.
3. Choudhary, V. “Software as a Service: Implications for Investment in Software Development.” *Proceedings of the 40th Hawaii International Conference on System Sciences* (2007). See also Bala, R., and S. Carr. “Usage-Based Pricing of Software Services Under Competition.” *Journal of Revenue and Pricing Management* 9 (2010): 204 to 216.

4. International Gas Union. World LNG Report 2023 (IGU Publications, 2023). On commodity contract theory broadly, see Yergin, D. *The Prize: The Epic Quest for Oil, Money, and Power* (New York: Simon & Schuster, 1991).
 5. Salesforce Inc. “Agentforce: The Evolution of Flexible AI Credits.” Salesforce Help Documentation (2026). Pricing trajectory reconstructed from Salesforce investor communications and public product announcements 2024 to 2026.
 6. Microsoft Corporation. “Microsoft 365 Copilot Pricing and Licensing.” Microsoft Learn (2024 to 2025). Copilot Studio consumption pricing was introduced in November 2024.
 7. Intercom. “Fin AI Agent Pricing” (Intercom Product Documentation, 2024 to 2025). See also Poyar, K., “A New Framework for AI Agent Pricing,” *Growth Unhinged* (2026).
 8. Zendesk Inc. “AI and Automation Pricing.” Zendesk Product Documentation (2024 to 2025).
 9. Gartner. “Worldwide IT Spending Forecast.” Gartner Research (2025).
 10. McKinsey Global Institute. *The Economic Potential of Generative AI: The Next Productivity Frontier* (McKinsey & Company, 2025). See also Bain & Company. “Pricing AI: How SaaS Companies Are Reinventing Monetization in the Agentic Era” (Bain Technology Practice, 2025).
- Additional scholarship engaged implicitly: Hagiu, A., and J. Wright. “Multi-Sided Platforms.” *International Journal of Industrial Organization* 43 (2015): 162 to 174. Bergemann, D., and J. Välimäki. “Dynamic Pricing of New Experience Goods.” *Journal of Political Economy* 114, no. 4 (2006): 713 to 743. Hart, O. *Firms, Contracts, and Financial Structure* (Oxford University Press, 1995). Schlereth, C., and B. Skiera. “Optimization of Two-Part Tariffs.” *Management Science* 58, no. 9 (2012): 1698 to 1716. Iyer, G., and J. Soberman. “Product Design and Pricing in the Presence of Risk.” *Management Science* 62, no. 8 (2016): 2264 to 2279.