

From SaaS to AI-Native Software

When Software Becomes a Producer: Pricing, Incentives and Scale under Variable Inference Costs

Jorge Menéndez-Pidal

Project Frontier, Madrid

ABSTRACT. Generative AI changes the production function of software: every answer, document or completed task consumes inference compute, so software acquires a positive marginal cost that scales with how intensively each customer uses it. I derive the consequences in three steps. First, flat access breaks. Heavy users become the least profitable, flat pricing's share of attainable profit falls monotonically with inference cost, and it collapses once a unit of inference costs half the value of the first unit of use. Second, and centrally, the unit of sale follows control of compute. When the customer controls usage, as with copilots, usage pricing disciplines consumption and pass-through of inference cost exceeds Borch's risk-sharing benchmark. When the provider controls the process, as with agents, usage pricing is cost-plus, and the efficient contract moves towards paying for outcomes. Outcome pricing is worth its verification cost when the compute bill per task is large. Third, with constant inference cost scale economies vanish; they survive only through firm-specific learning in inference. Common falls in compute prices raise margins without raising gross profit, and because AI also lowers the cost of building software, free entry pushes each firm's gross profit towards its entry cost. Financial statements of 307 US-listed software firms are consistent with these predictions. Early generative-AI adopters saw gross margins fall four to five points relative to other firms after 2023 while growing faster, and market value tracks gross profit more closely than revenue. The learning rate in inference cannot be identified from public accounts.

Preliminary draft. Sections 3–6 are theory, with all propositions proved in Appendix A; Section 8 illustrates them numerically; Section 9 reports descriptive evidence from SEC filings. Comments welcome; please do not cite without permission.

Keywords: software economics, generative AI, inference costs, nonlinear pricing, risk sharing, economies of scale, free entry, firm valuation. **JEL:** D42, D43, D86, L11, L13, L86, O33.

1 Introduction

For forty years the economics of software rested on one fact: once a program is written, serving more use of it costs almost nothing. Software firms do incur costs that scale with customers, such as hosting, support and billing, but a seat price covers those. What was close to free was *intensity*: a customer who used the product ten times as much as another cost the firm about the same. That fact explains why software is sold as access, by subscription and by seat. It also explains why software firms enjoy extraordinary operating leverage and why investors value them on revenue multiples that would be absurd in other industries.

Generative AI breaks that fact. An AI product does not deliver a copy of something already written; it produces something new each time it is used, and every answer, document or completed task consumes inference compute. AI does not merely add a cost to software. It changes the production function of software: software becomes a producer, and its output has a positive marginal cost that varies with how intensively each customer uses it.

This paper derives the consequences of that single change in three steps.

Pricing: why flat access breaks. Under a flat subscription, every customer consumes to satiation and the heaviest users become the least profitable. In conventional software, a heavy user was simply a better customer. With inference costs, a heavy user is both more valuable and more expensive to serve. Flat pricing's share of attainable profit falls monotonically as inference cost rises, and flat pricing collapses entirely once a unit of inference costs half the value of the first unit of use (Proposition 1). Usage-based and tiered contracts take its place, and AI plans become screens on computational intensity (Propositions 2–3).

Incentives: the unit of sale follows control of compute. This is the paper's central result. With uncertain usage, a contract decides who bears inference risk and who has an incentive to control it. When the customer controls consumption, as with a copilot, usage pricing disciplines it, and the optimal pass-through of inference cost lies above Borch's risk-sharing benchmark (Proposition 4). When the provider controls the process, as with an agent completing a task, usage pricing is cost-plus and removes the incentive to economise, so the optimal contract moves towards paying for outcomes (Proposition 5). Outcome pricing is worth the cost of verifying outcomes when the compute bill per task is large (Proposition 6). The observed migration from seats to usage, tasks and outcomes is therefore not a fashion in pricing. It follows from who controls the variable input, and it adapts the classic procurement trade-off between cost-plus and fixed-price contracts to a setting where product architecture determines which party controls cost.

Scale and rents. With constant inference cost, scale economies vanish asymptotically. They survive only through the firm's own learning in inference, with a long-run scale index of $1/(1 - b)$ for learning rate b (Proposition 7). Industry-wide falls in compute prices raise reported margins without raising gross profit; only firm-specific cost advantages create rents (Proposition 8). And because AI lowers the cost of building software while raising the cost of running it, free entry drives each firm's gross profit towards its entry cost (Proposition 9). This

is a benchmark rather than a forecast. It implies that cheaper software is worth less unless a firm has a source of advantage that entry cannot copy.

Evidence. Section 9 confronts these predictions with the financial statements of 307 US-listed software firms. Firms that discussed generative AI in their annual reports by 2023 saw gross margins fall four to five points relative to other software firms, while growing about nine points a year faster. Holding revenue fixed, market value rises with gross profit. The learning rate b cannot be identified from these data: software firms' cost of revenue has long moved one-for-one with revenue, and early adopters show no detectable change in that elasticity. The evidence is descriptive and the margin result rests on a small group of early adopters, but the patterns are those the theory predicts.

Roadmap. Section 2 places the paper in the literature. Section 3 sets out the model. Sections 4, 5 and 6 develop the three steps. Section 7 sketches two extensions, labour anchoring and make-or-buy. Section 8 illustrates the results numerically, Section 9 presents evidence, and Section 10 concludes. Proofs are in Appendix A.

2 Related literature

Information goods. The economics of software has rested on near-zero marginal cost. Shapiro and Varian (1999) describe how that cost structure shapes pricing, versioning and competition in information goods. Bakos and Brynjolfsson (1999) show that it makes large-scale bundling profitable: with negligible marginal cost, a seller can bundle heterogeneous goods and sell access at a flat price. This paper asks what happens to these results when marginal cost is no longer negligible, and finds that the flat-access logic breaks down in a specific way (Proposition 1).

Flat rates and nonlinear pricing. The two-part tariff of Oi (1971) and the screening models of Mussa and Rosen (1978) and Maskin and Riley (1984) underlie Propositions 2 and 3; Wilson (1993) surveys the field. Flat-rate versus metered pricing has a long empirical history. Train et al. (1987) model the choice between flat and measured local telephone service, Lambrecht and Skiera (2006) document that consumers often prefer flat rates even when metered plans would be cheaper, and DellaVigna and Malmendier (2006) show that gym members on flat contracts over-estimate their usage. In these markets marginal cost was low throughout, and the interest lay in demand and behavioural biases. What is new here is a change in the cost side within one industry: marginal cost moves from roughly zero to positive and, crucially, becomes proportional to usage intensity, so customer heterogeneity becomes cost heterogeneity. The adverse-selection logic of all-you-can-eat contracts in Proposition 1 is familiar; the contribution is to show when it binds and what replaces flat access. Bala and Carr (2010) study usage-based pricing of software services under competition.

Risk sharing, incentives and procurement. Proposition 4 applies Borch's rule for efficient risk sharing (Borch, 1962) to inference-cost risk. Proposition 5 applies the linear trade-off between insurance and incentives of Holmström and Milgrom (1987). The resulting choice between

usage and outcome pricing is the procurement choice between cost-plus and fixed-price contracts (Laffont and Tirole, 1993; Bajari and Tadelis, 2001), with outcome pricing playing the role of a fixed price per task. The contribution is to show that the two cases correspond to two product architectures, copilots and agents, and to derive from them a theory of the unit of sale.

Entry and the value of firms. Propositions 8 and 9 use the circular city of Salop (1979). The result that free entry dissipates rents until profit equals entry cost is classic (Mankiw and Whinston, 1986). Applied to software, it implies that a technology lowering the cost of building software lowers the value of the typical firm. Sutton (1991) shows how endogenous sunk costs can sustain concentration despite free entry, which bears on the ways an AI-native firm can escape the benchmark.

AI, automation and pricing. Acemoglu and Restrepo (2018) and Acemoglu and Restrepo (2019) model automation as AI taking over tasks and creating new ones; Section 7 uses their task framework to study pricing anchored to wages. Agrawal et al. (2018) frame AI as a fall in the cost of prediction, which here is the fall in c . Recent work studies the pricing of AI services directly. Lobel et al. (2026) study the pricing of AI provided as a service, and Li et al. (2024) study how a downstream generative-AI application should price when the upstream model provider charges per use, finding that competing applications may split between subscription and usage pricing. This paper differs in tying pricing, risk allocation, scale and valuation to a single change in the production technology, the marginal cost of inference.

3 A software firm with a variable cost of production

3.1 Technology

A software firm incurs a fixed cost F to build its product and a marginal cost c for every unit of output it delivers:

$$C(Q) = F + cQ. \quad (1)$$

For conventional software, $c \approx 0$: once the code exists, serving another user costs almost nothing. For AI-native software, every unit of work the product performs, whether an answer, a document or a completed task, requires inference, and $c > 0$. More generally $c = c(\text{model}, \text{context}, \text{steps})$. For an agent that completes a task in a random number of steps K , each using a units of compute, the cost of the task is $c a K$. I take c as the price of a unit of compute, whether bought from an upstream model provider or produced in-house. Conventional software is not literally costless to serve: hosting, support and payment processing scale with the number of customers, and Section 9 shows that software firms' cost of revenue has long moved one-for-one with revenue. Those costs scale with *seats*, which the seat price covers. What is new with AI is a cost that scales with *intensity of use within a seat*, which a seat price does not cover. The parameter c is that cost.

The ray economies-of-scale index, average cost over marginal cost, summarises the difference:

$$S(Q) = \frac{AC(Q)}{MC(Q)} = 1 + \frac{F}{cQ}. \quad (2)$$

With $c \rightarrow 0$, S is unbounded: scale economies never run out. With $c > 0$, $S \rightarrow 1$ as Q grows, and the firm converges to constant returns, like a services firm. Section 6 returns to this, allowing c to fall with the firm's own volume.

3.2 Customers

Customers buy access for a fixed number of seats, which I normalise to one; seats are observable and contractible. What differs across customers is how intensively each seat uses the product. A customer of intensity X who consumes q units of AI work obtains

$$u(q; X) = \theta \left(q - \frac{q^2}{2X} \right), \quad 0 \leq q \leq X, \quad (3)$$

where θ is the value of the first unit and X is the satiation level. Facing a marginal price p , the customer consumes

$$q(p; X) = X \left(1 - \frac{p}{\theta} \right), \quad \text{with surplus} \quad CS(p; X) = \frac{X(\theta - p)^2}{2\theta}. \quad (4)$$

Both consumption and surplus scale with X . Heavy users value the product more *and* cost more to serve, so customer heterogeneity is at once heterogeneity in demand and in cost. This is the feature that distinguishes AI-native from conventional software. Under $c \approx 0$ a heavy user is a pure gain; under $c > 0$ the heaviest user may be the least profitable.

To obtain closed forms, there are two intensities: light users with $X = 1$ and heavy users with $X = r > 1$, a share ϕ of the market. Mean intensity is $\mu \equiv 1 - \phi + \phi r$, and r measures usage heterogeneity. Section 8 confirms that the qualitative results survive richer distributions.

3.3 Contracts

The firm chooses among three families of contracts, which correspond to the pricing models observed in software.

1. *Flat (seat) pricing*: a fee A per seat, with unlimited use. Every subscriber consumes to satiation, $q = X$.
2. *Two-part (hybrid) pricing*: a fee A per seat plus a price p per unit of usage.
3. *A menu of plans*: bundles $\{(q_L, T_L), (q_H, T_H)\}$ of usage allowances and prices, the tiered plans of AI products. This is the optimal nonlinear tariff.

Any contract with a usage component requires metering, billing and explaining a variable bill. I summarise these costs, borne by the firm, as $m \geq 0$ per customer. Flat pricing is the special case $p = 0$ without metering, so with $m = 0$ the two-part tariff can never do worse than flat pricing. The interesting question is when the gain is large enough to justify m .

4 Pricing: why flat access breaks

4.1 Flat pricing: adverse selection and collapse

Under a flat fee every subscriber consumes to satiation, so a customer of intensity X costs cX and is worth $\theta X/2$ to herself. The firm either prices for both types, $A = \theta/2$, or only for heavy

users, $A = \theta r/2$.

Proposition 1 (Flat pricing under variable cost). *Under flat pricing:*

- (i) *The marginal subscriber is always the lightest user, and the per-customer margin $A - cX$ is decreasing in intensity: heavy users are infra-marginal and the least profitable.*
- (ii) *Relative to efficient use, each subscriber over-consumes, with deadweight loss $c^2X/(2\theta)$, increasing in the square of the inference cost.*
- (iii) *The optimal flat profit is $\Pi_F = \max\{\theta/2 - c\mu, \phi r(\theta/2 - c), 0\}$. It is non-increasing in c and zero for $c \geq \theta/2$: flat pricing cannot be profitable once a unit of inference costs half the value of the first unit of use.*

Part (i) is the adverse-selection logic of all-you-can-eat contracts. Raising the fee drives out the cheapest customers first, so the firm cannot reprice its way out of heavy usage. Part (iii) is a collapse condition. Since every subscriber consumes to satiation, total surplus under a flat contract is $\theta X/2 - cX$ per customer, which is negative for every customer when $c \geq \theta/2$. Usage-based contracts have no such threshold: they remain profitable for any $c < \theta$.

4.2 Two-part and nonlinear pricing

Proposition 2 (Two-part pricing). *If the optimal two-part tariff serves both types, the per-unit price and profit are*

$$p^* = c + \frac{(\mu - 1)(\theta - c)}{2\mu - 1}, \quad \Pi_T = \frac{\mu^2(\theta - c)^2}{2\theta(2\mu - 1)} - m.$$

- (i) *With homogeneous usage ($\mu = 1$) the firm prices usage at cost, $p^* = c$, and extracts all surplus through the fee (the classic two-part tariff of Oi, 1971).*
- (ii) *With heterogeneous usage ($\mu > 1$) it marks usage up above cost, using the per-unit price to extract more from heavy users.*
- (iii) *Ignoring metering costs, the share of attainable profit captured by flat pricing, $(\theta/2 - c\mu)/(\Pi_T + m)$, is strictly decreasing in c .*

Part (iii) gives the comparative static in its robust form. As inference becomes more expensive, flat pricing leaves an ever larger share of the available profit on the table. The absolute gap between the two contracts is not monotone over the whole range of c . As c approaches θ every contract's profit vanishes, so the gap eventually closes. For this reason the result is stated in shares.

Proposition 3 (Plans as screening on computational intensity). *The optimal menu gives heavy users their efficient allowance, $q_H = r(1 - c/\theta)$, and distorts the light users' allowance downwards:*

$$q_L = \frac{r(1 - \phi)(\theta - c)}{\theta(r - \phi)} < 1 - \frac{c}{\theta}.$$

Heavy users earn an information rent $\theta q_L^2(1 - 1/r)/2$. The menu weakly dominates the two-part tariff, which weakly dominates flat pricing whenever $m = 0$.

Tiered AI plans therefore screen customers on computational intensity, not only on willingness to pay, and the tiers differ in compute allowances because compute is what drives cost. In conventional SaaS, tiers differentiate features. In AI-native software they also ration the variable input.

When is flat pricing still optimal?. With metering costs, flat pricing is optimal if and only if $\Pi_F \geq \max(\Pi_T, \Pi_M)$, where Π_M is menu profit net of m . At $c = 0$ and moderate heterogeneity, flat pricing wins whenever m exceeds the small extraction gain from usage pricing. As c rises, flat pricing's share of attainable profit falls (Proposition 2) and eventually reaches zero. In the numerical exploration of Section 8 the set of costs for which flat pricing is optimal is an interval $[0, \bar{c}(m, r, \phi)]$ in 99.8% of parameter combinations, with $\bar{c}$ increasing in the metering cost and decreasing in usage heterogeneity. The exceptions occur near the kink where the flat contract switches from serving both types to serving only heavy users.

5 Incentives: the unit of sale follows control of compute

This section contains the paper's central result. Section 4 treated usage as known when the contract is signed. In practice a customer's usage, and the compute needed to complete a task, are uncertain, so a contract also decides who bears the variance of inference cost, and who has an incentive to control it. Write a linear contract as a fixed fee plus a share $p \in [0, c]$ of each unit of compute passed through to the customer. Then $p = 0$ is a flat contract, $p = c$ is cost-plus usage pricing, and the *pass-through rate* p/c measures how much inference risk the customer bears. Buyer and seller have mean-variance preferences with risk-aversion coefficients ρ_B and ρ_S .

The trade-off is the one familiar from procurement, between cost-plus contracts, which insure the supplier, and fixed-price contracts, which give it incentives (Laffont and Tirole, 1993; Bajari and Tadelis, 2001). What is new is that in AI-native software the party who controls the variable input differs across product architectures. In a copilot the customer decides how much compute to consume; in an agent the provider does.

5.1 Copilots: the customer controls usage

Suppose the customer chooses how much to use, as with a coding assistant or a chat seat. Usage is $q = X(1 - p/\theta) + \varepsilon$, with $\text{Var}(\varepsilon) = \sigma^2$. The fixed fee transfers surplus, so the contract maximises joint surplus. Pricing usage below cost induces over-use, while passing cost through loads risk on the buyer.

Proposition 4 (Pass-through when the customer controls usage). *The surplus-maximising pass-through rate is*

$$\frac{p^*}{c} = \frac{X/\theta + \rho_S \sigma^2}{X/\theta + (\rho_B + \rho_S) \sigma^2} \in (0, 1].$$

It equals one when usage is certain or the buyer is risk-neutral. It falls as the buyer's risk aversion or the volatility of usage rises, and rises with the seller's risk aversion and with the price-sensitivity of usage X/θ .

Without the usage term, $p^*/c = \rho_S/(\rho_B + \rho_S)$, Borch's rule for sharing a risk between two

parties (Borch, 1962). The usage term pushes pass-through above Borch's level, because a price below cost invites over-consumption by the party who controls consumption. The proposition also implies that large, diversified vendors (low ρ_S) can absorb inference risk and offer flat or capped plans, while young firms (high ρ_S) pass it through.

5.2 Agents: the provider controls the process

Now suppose the product is an agent that completes a task, and the *provider* controls how much compute the task consumes. A task takes a random number of steps K with mean μ_K and variance σ_K^2 , each using a units of compute. By exerting effort e , at cost $e^2/(2k)$, the provider reduces the expected number of steps to $\mu_K - e$. Engineering effort of this kind includes better prompting, routing, caching and verification. Here k measures how productive such engineering is. Under pass-through p , the provider bears $c - p$ of each unit of compute.

Proposition 5 (Pass-through when the provider controls compute). *The provider chooses $e^* = k a (c - p)$, and the surplus-maximising pass-through rate is*

$$\frac{p^*}{c} = \frac{\rho_S \sigma_K^2}{k + (\rho_B + \rho_S) \sigma_K^2} < \frac{\rho_S}{\rho_B + \rho_S}.$$

It is strictly below the Borch benchmark and tends to zero, pure outcome pricing, as the return to efficiency effort k grows.

Usage pricing at $p = c$ is cost-plus. It removes the provider's incentive to economise on compute, and can even reward padding. Outcome pricing makes the provider the residual claimant on efficiency, as a fixed-price contract makes a contractor the residual claimant on cost.

5.3 The unit of sale

Propositions 4 and 5 imply a simple organising principle:

Corollary 1 (Unit of sale). *Charge for usage when the customer controls how much compute is consumed, and charge for outcomes when the provider does. Holding risk preferences fixed, the optimal pass-through is higher for copilots than for agents.*

As products move from assisting a user to completing tasks autonomously, control of compute shifts from customer to provider, and the efficient unit of sale shifts from usage to tasks and outcomes. The familiar sequence

$$\text{seat} \rightarrow \text{usage} \rightarrow \text{task} \rightarrow \text{outcome}$$

is therefore not a fashion in pricing. It follows from who controls the variable input.

Outcome contracts have a cost of their own: someone must verify that the task was completed, and done well. Let verifying an outcome cost κ per task.

Proposition 6 (When outcomes are worth verifying). *For a provider-controlled task, the optimal*

outcome-based contract yields higher surplus than cost-plus usage pricing if and only if

$$\kappa < G \equiv \frac{(ac)^2 (k + \rho_B \sigma_K^2)^2}{2 (k + (\rho_B + \rho_S) \sigma_K^2)}.$$

The gain G rises with the square of the expected compute bill per step, ac , and with the return to efficiency effort k .

Outcome pricing is worth its verification cost when the compute bill per task is large, when engineering can reduce it substantially, and when buyers are risk-averse. It is not worth it for cheap, routine tasks, or where outcomes are hard to verify. This gives the sequence above a natural order. Tasks with clear, cheaply verifiable outcomes and heavy compute, such as resolving a support ticket, completing a code change or processing a claim, move to outcome pricing first. Tasks with diffuse or hard-to-judge outcomes stay priced by usage.

6 Scale and rents

6.1 Do AI-native firms scale like software?

Equation (2) shows that a constant inference cost exhausts scale economies. Gross margin, $1 - c/P$, does not improve with volume, and operating margin is capped at $1 - c/P$. Scale returns only if the firm's *own* cost of serving a unit falls with its volume, through routing, caching, distillation or its own models. Let $c(Q) = c_0 Q^{-b}$, with $b \in [0, 1)$ the firm's learning rate in inference.

Proposition 7 (Scale economies with inference learning). *With $C(Q) = F + c_0 Q^{1-b}$,*

$$S(Q) = \frac{1}{1-b} \left(1 + \frac{F}{c_0} Q^{b-1} \right) \xrightarrow{Q \rightarrow \infty} \frac{1}{1-b}.$$

Long-run scale economies are governed entirely by the firm's learning rate in inference: none if $b = 0$, unbounded only as $b \rightarrow 1$, the conventional-software limit.

Whether an AI-native firm “scales like software” is therefore an empirical question about b , not about the level of c . That leads to the question of what kind of cost reduction a firm can keep.

6.2 Common versus idiosyncratic cost deflation

Inference prices have fallen rapidly. Much of the decline is *common*: an upstream model provider cuts its price for everyone. Some is *idiosyncratic*: one firm learns to serve a task with less compute.

Proposition 8 (Who captures falling inference costs). *Let N application firms compete on a circle with transport cost t (Salop, 1979), all buying compute at price c , with market size L .*

- (i) *In symmetric equilibrium $P = c + t/N$. A common fall in c passes one-for-one to customers. Gross profit per unit, t/N , is unchanged, while gross margin $(t/N)/(c + t/N)$ rises mechanically.*
- (ii) *In a homogeneous-product (Bertrand) market, a firm whose cost is δ below its rivals' earns a rent of δ per unit served. Only firm-specific reductions in c create rents.*

Part (i) separates two quantities that are often conflated. A common fall in inference prices raises reported gross margins without raising gross *profit*. Investors who value gross profit should not reward it; investors who value revenue multiples adjusted for margin may be misled by it. Part (ii) says where durable advantage comes from: the firm-specific learning rate b of Proposition 7, not the industry-wide price of compute.

6.3 Free entry and the cost-structure rotation

Generative AI lowers the cost of *building* software, F , at the same time as it raises the cost of *running* it, c . Call this the cost-structure rotation. Its consequences for the value of a typical firm are stark.

Proposition 9 (Free entry: the typical firm is worth what it cost to build). *In the circular market of Proposition 8 with free entry, the number of firms is $N^* = \sqrt{Lt/F}$, and each firm's gross profit equals its entry cost:*

$$\text{gross profit per firm} = \frac{Lt}{N^{*2}} = F.$$

Gross profit per firm is independent of the inference cost c and of market size L . If the firm is valued at a multiple M of gross profit, its value is MF , and a fall in F lowers the value of the typical firm one-for-one.

Proposition 9 formalises the intuition that AI makes software “cheaper to build but harder to defend”. Lower F invites entry, entry erodes the markup t/N , and in equilibrium each firm earns back only what it spent to enter. Neither a larger addressable market nor cheaper compute changes this: both are dissipated by entry. A unicorn, a firm with value $V_U = \$1\text{bn}$, must therefore be an *exception* to the symmetric equilibrium. It needs a firm-specific cost advantage (Proposition 8(ii)), a larger differentiation parameter t than its rivals, or demand-side scale economies that the benchmark omits.

6.4 The unicorn threshold

For a firm valued at a multiple $M(g, \delta)$ of gross profit, where g is growth and δ the durability of its advantage, the scale needed to reach value V_U is

$$Q_U = \frac{V_U}{M(g, \delta)(P - c)}, \quad (5)$$

which rises with c and falls with durability. In terms of observable quantities, the unicorn frontier is the hyperbola $\text{ARR} \times \text{GM} = V_U/M$. If revenue grows exponentially from Q_0 at rate g , the time to a \$1bn valuation is $T_U = g^{-1} \ln(Q_U/Q_0)$. An AI-native firm reaches it faster than a conventional software firm if and only if

$$\frac{g_{AI}}{g_{SaaS}} > \frac{\ln(Q_U^{AI}/Q_0)}{\ln(Q_U^{SaaS}/Q_0)}. \quad (6)$$

This race between growth and margin reconciles two observations. AI-native firms can reach very large revenue unusually fast, yet markets decline to value them at the revenue multiples of classic software.

7 Extensions: labour anchoring and firm boundaries

Two extensions connect the framework to questions developed elsewhere. When AI completes work that people would otherwise do, the competitive benchmark becomes the wage rather than other software; a fuller treatment belongs with the study of AI and labour. And the make-or-buy decision for inference links pricing to the boundaries of the firm.

7.1 Labour-anchored pricing

When AI completes a task that a worker would otherwise do, the customer's outside option is no longer competing software but labour. Consider a continuum of tasks, task j costing w_j to perform with labour and c with AI. A task is automated only if $c < w_j$, so the share of tasks automated is $H(c) = \Pr(w_j > c)$, and it rises as inference becomes cheaper, the extensive margin of Acemoglu and Restrepo (2018).

Proposition 10 (Labour anchoring requires market power). *For an automated task, the provider's price is $P_j = \min\{w_j, c + t/N\}$.*

- (i) *With a single provider ($t/N \geq w_j - c$), the price is anchored to the wage: $P_j = w_j$ and gross margin $1 - c/w_j$ rises as c falls.*
- (ii) *With competing providers ($t/N < w_j - c$), the price is anchored to cost: $P_j = c + t/N$, and falling c passes to customers as in Proposition 8.*
- (iii) *Revenue from automation is $\int_{w>c} \min\{w, c + t/N\} dH$. The addressable market expands as c falls, but under competition each automated task contributes only the markup t/N to gross profit.*

Pricing work at “a fraction of the cost of a person” is therefore a property of market structure, not of contract design. Outcome-based contracts make the comparison with labour salient, but where several providers can complete the same task, competition drives the price towards the cost of compute plus a differentiation margin. Labour anchoring is the payoff to being the only credible provider of a task, which links this section back to the sources of advantage in Proposition 8.

7.2 Make or buy

An application firm can buy inference at price p_{API} or serve its own models at fixed cost F_M and marginal cost $c_M < p_{API}$. It integrates vertically when volume exceeds

$$q^* = \frac{F_M}{p_{API} - c_M}. \quad (7)$$

Falling API prices raise q^* and keep firms on external models; falling in-house costs lower it. Through Proposition 8, the threshold also governs which deflation a firm can keep. Below q^* the firm rides the common price path of its supplier. Above it, its own efficiency gains are idiosyncratic and become rents.

8 Numerical illustration

This section illustrates the propositions with $\theta = 1$, so that c is the cost of a unit of inference relative to the value of the first unit of use. All parameter values are illustrative.

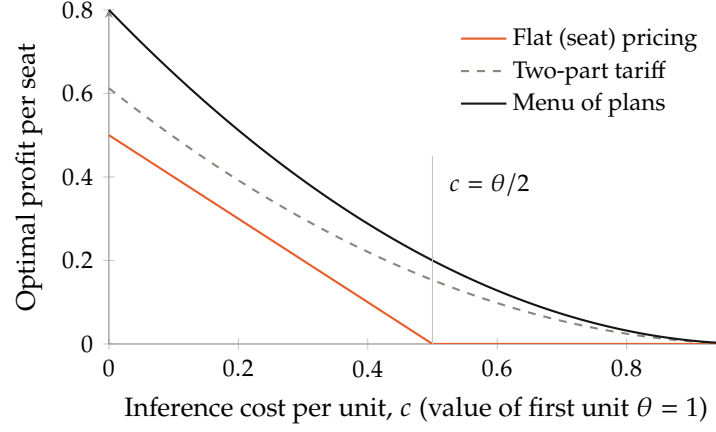


FIGURE 1. Optimal profit by contract as inference cost rises ($r = 4$, $\phi = 0.25$, $m = 0$). Flat pricing earns nothing once $c \geq \theta/2$.

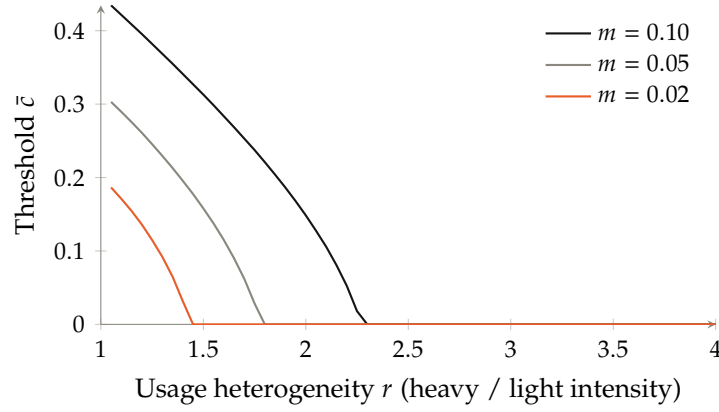


FIGURE 2. Inference cost below which flat pricing is optimal, by usage heterogeneity and metering cost m ($\phi = 0.25$).

Contracts and inference cost. Figure 1 plots optimal profits under each contract when heavy users consume four times as much as light users ($r = 4$) and make up a quarter of the market ($\phi = 0.25$). Flat pricing collapses at $c = 0.5$. Before that point its share of attainable profit falls steadily, from 62% at $c = 0$ to 51% at $c = 0.3$. The menu and the two-part tariff remain profitable throughout.

When is flat pricing still optimal?. Figure 2 plots the threshold $\bar{c}$ below which flat pricing beats the best usage-based contract, for three metering costs. The threshold rises with the metering cost and falls with usage heterogeneity, as the discussion after Proposition 3 states. Heterogeneity matters even at $c = 0$. With heavy users consuming several times as much as light ones, a menu of plans already beats a flat fee by more than plausible metering costs. Conventional SaaS also sold tiers, but differentiated them by features rather than by consumption. Flat access survives at low inference cost only when usage is fairly homogeneous or metering is costly. Across 5,000 combinations of (r, ϕ, m) the region where flat pricing is optimal is an interval starting at zero in 99.8% of cases.

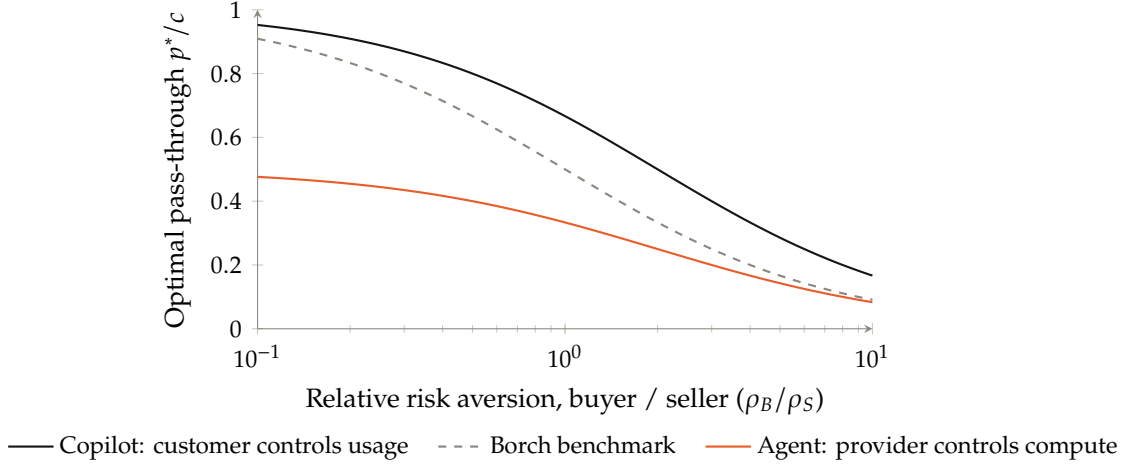


FIGURE 3. Share of inference cost passed through to the customer. Usage pricing corresponds to $p^*/c = 1$, outcome pricing to $p^*/c = 0$. Parameters: $X/\theta = k = \sigma^2 = \rho_S = 1$.

Richer heterogeneity. With a continuum of intensities drawn from a lognormal distribution (log-standard deviations of 0.5 and 1), the optimal flat fee and two-part tariff computed numerically behave as in the two-type model. Flat pricing's share of attainable profit falls monotonically with c , and flat pricing becomes unprofitable at or before $c = \theta/2$. Greater dispersion lowers flat pricing's share even at $c = 0$, from 77% to 57%.

Pass-through. Figure 3 plots the optimal pass-through rate against relative risk aversion ρ_B/ρ_S for a copilot and an agent with the same risk parameters. The copilot lies above the Borch benchmark and the agent below it at every level of risk aversion, which is Corollary 1.

Scale and entry. Under Proposition 7, a firm with no inference learning ($b = 0$) sees its scale index fall towards one as it grows. A firm with $b = 0.5$ retains a long-run index of two, while conventional software's index grows without bound. Under Proposition 9, halving the cost of building a product raises the number of firms in a free-entry market by a factor of $\sqrt{2}$ and halves the value of each. For a typical firm in that benchmark to be a unicorn at a gross-profit multiple of 10, it must have cost \$100m to build.

9 Evidence from SEC filings

The theory has testable implications for the financial statements of software firms. If AI raises the marginal cost of serving customers, firms that build it into their products should see cost of revenue rise relative to revenue, so their gross margins should fall relative to other software firms (Proposition 1 and Section 6). If the race between growth and margin of equation (6) operates, the same firms should grow faster. And if markets value the gross profit that survives competition rather than revenue that passes through compute costs (Proposition 8), market value should track gross profit more closely than revenue. This section reports descriptive evidence on all three. None of it identifies causal effects.

TABLE 1. Early generative-AI adopters versus other software firms, before and after 2023

	Gross margin (1)	Rev. growth (2)	R&D share (3)	S&M share (4)
Early adopter × post	−4.56** (2.22)	9.19** (3.96)	−1.23 (4.76)	−0.68 (2.22)
Firms	307	307	270	248
Firm-years	2,013	1,870	1,749	1,613
<i>Robustness (columns 1–2)</i>				
Any GenAI mention by 2024	0.44 (1.01)	5.45* (3.16)		
Revenue ≥\$100m in 2022	−1.94 (1.21)	4.10 (3.27)		
Excluding fiscal 2025	−4.36* (2.24)	11.21*** (3.71)		

Notes: US-listed software firms (SIC 7370–7374) with at least \$10m of revenue in fiscal 2022, fiscal years 2018–2025, from SEC XBRL financial data. Early adopters are the 32 firms whose 10-K filed by the end of 2023 mentions “generative AI” or “large language model(s)”. Coefficients in percentage points; firm and year fixed effects; standard errors clustered by firm. Variables winsorised at the 1st and 99th percentiles. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

9.1 Data

I assemble the universe of firms reporting revenue in the SEC’s XBRL financial-statement data and keep those classified as software and computer services (SIC 7370–7374), 718 firms. From each firm’s annual 10-K filings I extract revenue, cost of revenue or gross profit, research and development, and selling and marketing expenses for fiscal years 2018–2025, and the public float reported on the cover page. The estimation sample keeps firms with at least \$10m of revenue in fiscal 2022: 307 firms and 2,013 firm-years.

Exposure to generative AI is measured from the text of annual reports. Using the SEC’s full-text search, I flag 10-K filings that mention “generative AI” or “large language model(s)”. *Early adopters* are the 32 sample firms whose 10-K filed by the end of 2023 contains these terms. In the months after ChatGPT’s release, such mentions were rare and plausibly signal firms whose products rely on the technology. By 2024 the terms had spread to the risk-factor boilerplate of 141 sample firms, so later mentions discriminate far less. A narrower measure, mentions of inference or compute costs, flags only 11 firms and is too sparse to use.

9.2 Gross margins and growth

Table 1 compares early adopters with other software firms before and after 2023, with firm and year fixed effects. Early adopters’ gross margins fall by −4.6 percentage points relative to other firms (s.e. = 2.2), while their revenue grows 9.2 points a year faster. The research and development and sales and marketing shares of revenue show no relative change, so the fall in fixed development cost that the theory also predicts is not visible in these accounts.

Figure 4 shows the timing. Before 2023 the gap between early adopters and other firms is stable and statistically indistinguishable from its 2022 level. From 2023 it opens, reaching about five points in 2024. The divergence is relative: median gross margins of other software firms rose from 64% in 2022 to 70% in 2025, as many firms cut hosting and service costs, while early adopters’ margins stayed flat at around 73–74%.

The result is fragile in ways the table reports. Defining treatment as any mention by 2024 removes it, which is consistent with later mentions being boilerplate but also means the effect

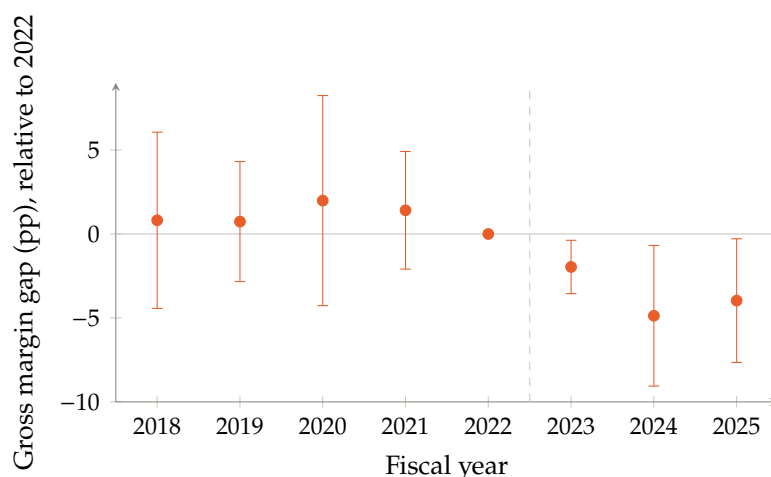


FIGURE 4. Gross margins of early generative-AI adopters relative to other software firms, by fiscal year (event study, 2022 = 0). Bars are 95% confidence intervals; firm and year fixed effects, standard errors clustered by firm.

rests on 32 firms. Among firms with at least \$100m of revenue the margin effect halves to -1.9 points and loses significance. And early adopters differ from other firms in level: their margins were about nine points higher before 2023. The pattern is what the theory predicts for firms whose products embed inference, but selection into early adoption could produce it too.

9.3 Is cost of revenue becoming more variable?

Proposition 7 makes the learning rate b central, but firms do not disclose inference volumes or unit costs. A feasible proxy is the within-firm elasticity of cost of revenue to revenue, ε , estimated from annual log changes with year effects. With constant prices, $1 - \varepsilon$ approximates b , although it also absorbs fixed components of cost of revenue and price changes. Before 2023, cost of revenue moved one-for-one with revenue among other software firms ($\varepsilon = 1.00$, s.e. 0.04), consistent with costs that scale with seats. After 2023 the elasticity fell for both groups, to 0.83 (0.05) for other firms and 0.71 (0.13) for early adopters, a period of widespread cost-cutting. The difference-in-differences in the elasticity is 0.04 (0.16). These data cannot identify b , nor detect a shift towards more variable costs among adopters.

9.4 What markets value

Table 2 regresses the log of public float on log revenue and log gross profit for 2023–2024. Each explains market value about equally well on its own. When both are included, the coefficient on gross profit is 0.72 and that on revenue 0.41: holding revenue fixed, firms with higher gross profit, that is higher gross margins, are worth more. This is what Proposition 8 implies if investors value the profit that survives competition rather than revenue that includes pass-through of compute costs. It is also consistent with the race between growth and margin. Revenue growth carries a large premium of its own, so a firm that trades margin for growth can still gain value, as long as growth outruns the margin it gives up.

TABLE 2. What markets value: revenue or gross profit

Dependent variable: log public float	(1)	(2)	(3)	(4)
log revenue	1.15*** (0.05)		0.41* (0.23)	0.41** (0.20)
log gross profit		1.10*** (0.05)	0.72*** (0.23)	0.72*** (0.20)
revenue growth				2.20*** (0.37)
R^2	0.670	0.685	0.692	0.726
Observations	465	465	465	464

Notes: Software firms, fiscal years 2023–2024, with public float, revenue and gross profit above \$10m. Public float is the market value of shares held by non-affiliates reported on the 10-K cover page. Year fixed effects; standard errors clustered by firm. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

10 Conclusion

Generative AI turns software into a producer whose output has a positive marginal cost that scales with use. This paper has traced what follows from that single change.

Flat access, the business model that near-zero marginal cost made natural, becomes fragile: it selects the heaviest users, over-supplies compute and collapses once inference is expensive enough. What replaces it depends on who controls compute. When the customer does, usage pricing disciplines consumption. When the provider does, as with agents, usage pricing is cost-plus, and paying for outcomes gives the provider a reason to economise, provided outcomes can be verified at a cost below the gain. The shift from selling seats to selling outcomes is the procurement choice between cost-plus and fixed-price contracts, made by product architecture.

The same change alters what software firms are worth. Scale economies survive only through firm-specific learning in inference. Industry-wide falls in compute prices raise reported margins without raising gross profit. And by lowering the cost of building software, AI invites entry until, in the benchmark, the typical firm earns back only what it cost to build. The firms that become very valuable are exceptions to this benchmark: those with a proprietary cost advantage, unusual differentiation, or a position that lets them price work at the wage it replaces.

Financial statements of US-listed software firms are consistent with these predictions, though they do not establish them. Firms that discussed generative AI early saw their gross margins fall by four to five points relative to other software firms after 2023, while growing faster, and markets value software firms more closely on gross profit than on revenue. The margin result rests on a small group of early adopters and weakens among larger firms, and public accounts cannot reveal how fast firms learn to economise on inference.

Limitations. The model is deliberately simple. Customers differ along one dimension; the learning rate b , which governs whether AI-native firms scale like software, cannot be identified from public accounts; the firm is a monopolist in the pricing sections and symmetric in the entry sections; risk preferences are mean-variance; and the model of agents treats verification

as a fixed cost per task and abstracts from the quality of outcomes. Inference cost is treated as linear in usage, while batching, caching and routing make it concave in practice, which is the learning rate b in reduced form. The evidence is descriptive.

Next steps. The theory points to data that would test it directly. Contract-level data on AI pricing would test whether pass-through is higher for copilots than for agents and for smaller vendors. Firm-level data on inference spending would measure the learning rate b and separate common from idiosyncratic cost declines. And the entry result predicts falling gross profit per firm in segments where AI has lowered development costs most.

References

- Daron Acemoglu and Pascual Restrepo. The race between man and machine: Implications of technology for growth, factor shares, and employment. *American Economic Review*, 108(6): 1488–1542, 2018.
- Daron Acemoglu and Pascual Restrepo. Automation and new tasks: How technology displaces and reinstates labor. *Journal of Economic Perspectives*, 33(2):3–30, 2019.
- Ajay Agrawal, Joshua Gans, and Avi Goldfarb. *Prediction Machines: The Simple Economics of Artificial Intelligence*. Harvard Business Review Press, Boston, 2018.
- Patrick Bajari and Steven Tadelis. Incentives versus transaction costs: A theory of procurement contracts. *RAND Journal of Economics*, 32(3):387–407, 2001.
- Yannis Bakos and Erik Brynjolfsson. Bundling information goods: Pricing, profits, and efficiency. *Management Science*, 45(12):1613–1630, 1999.
- Ram Bala and Scott Carr. Usage-based pricing of software services under competition. *Journal of Revenue and Pricing Management*, 9:204–216, 2010.
- Karl Borch. Equilibrium in a reinsurance market. *Econometrica*, 30(3):424–444, 1962.
- Stefano DellaVigna and Ulrike Malmendier. Paying not to go to the gym. *American Economic Review*, 96(3):694–719, 2006.
- Bengt Holmström and Paul Milgrom. Aggregation and linearity in the provision of intertemporal incentives. *Econometrica*, 55(2):303–328, 1987.
- Jean-Jacques Laffont and Jean Tirole. *A Theory of Incentives in Procurement and Regulation*. MIT Press, Cambridge, MA, 1993.
- Anja Lambrecht and Bernd Skiera. Paying too much and being happy about it: Existence, causes, and consequences of tariff-choice biases. *Journal of Marketing Research*, 43(2):212–223, 2006.
- Liaoliao Li, Yali Zhang, Jun Sun, and Zhaojun Yang. Generative artificial intelligence application pricing: Navigating upstream strategy, consumer behavior, and market competition. Working Paper 4956548, SSRN, 2024.

- Ilan Lobel, Y. Sun, and J. Wu. Pricing AI as a service. Working paper, 2026.
- N. Gregory Mankiw and Michael D. Whinston. Free entry and social inefficiency. *RAND Journal of Economics*, 17(1):48–58, 1986.
- Eric Maskin and John Riley. Monopoly with incomplete information. *RAND Journal of Economics*, 15(2):171–196, 1984.
- Michael Mussa and Sherwin Rosen. Monopoly and product quality. *Journal of Economic Theory*, 18(2):301–317, 1978.
- Walter Y. Oi. A Disneyland dilemma: Two-part tariffs for a Mickey Mouse monopoly. *Quarterly Journal of Economics*, 85(1):77–96, 1971.
- Steven C. Salop. Monopolistic competition with outside goods. *Bell Journal of Economics*, 10(1):141–156, 1979.
- Carl Shapiro and Hal R. Varian. *Information Rules: A Strategic Guide to the Network Economy*. Harvard Business School Press, Boston, 1999.
- John Sutton. *Sunk Costs and Market Structure*. MIT Press, Cambridge, MA, 1991.
- Kenneth E. Train, Daniel L. McFadden, and Moshe Ben-Akiva. The demand for local telephone service: A fully discrete model of residential calling patterns and service choices. *RAND Journal of Economics*, 18(1):109–123, 1987.
- Robert B. Wilson. *Nonlinear Pricing*. Oxford University Press, New York, 1993.

A Proofs

Demand and surplus. Maximising $u(q; X) - pq$ gives $\theta(1 - q/X) = p$, hence equation (4). Substituting, $u - pq = X(\theta - p)^2/(2\theta)$.

Proposition 1. (i) A customer subscribes if $\theta X/2 \geq A$, so the subscriber set is an upper interval in X and the marginal subscriber has the lowest participating X . The margin $A - cX$ is decreasing in X . (ii) Efficient use solves $\theta(1 - q/X) = c$, so $q^e = X(1 - c/\theta)$. The welfare loss from consuming X instead is

$$\int_{q^e}^X \left[c - \theta \left(1 - \frac{q}{X} \right) \right] dq = \frac{c^2 X}{2\theta}.$$

(iii) Serving both types requires $A \leq \theta/2$, and profit is $A - c\mu$, maximised at $A = \theta/2$. Serving only heavy users requires $A \leq \theta r/2$, giving $\phi(\theta r/2 - cr)$. Both expressions are linear and decreasing in c , so their maximum with zero is non-increasing. For $c \geq \theta/2$, $\theta/2 - c\mu \leq \theta/2 - c \leq 0$ and $\phi r(\theta/2 - c) \leq 0$.

Proposition 2. Serving both types, the light type's participation binds: $A = (\theta - p)^2 / (2\theta)$. Profit is $\Pi(p) = (\theta - p)^2 / (2\theta) + (p - c)(1 - p/\theta)\mu - m$. The first-order condition $-(\theta - p) + \mu(\theta + c - 2p) = 0$ gives

$$p^* = \frac{\mu c + \mu\theta - \theta}{2\mu - 1} = c + \frac{(\mu - 1)(\theta - c)}{2\mu - 1},$$

and substitution gives Π_T . The heavy type's participation is slack because her surplus is r times the light type's. (i) and (ii) follow from $p^* - c \propto \mu - 1$. For (iii), the ratio $R(c) = (\theta/2 - c\mu)2\theta(2\mu - 1)/[\mu^2(\theta - c)^2]$ has derivative with the sign of

$$-\mu(\theta - c)^2 + 2(\theta - c)(\theta/2 - c\mu) = (\theta - c)[\theta(1 - \mu) - c\mu] < 0 \quad \text{for } \mu \geq 1, c > 0.$$

Proposition 3. With usage contractible, the firm offers (q_L, T_L) and (q_H, T_H) . The light type's participation and the heavy type's incentive constraint bind. The heavy type's rent from mimicking is $u(q_L; r) - u(q_L; 1) = \theta q_L^2(1 - 1/r)/2$. Profit is

$$(1 - \phi)[u(q_L; 1) - cq_L] + \phi[u(q_H; r) - cq_H] - \phi \theta q_L^2(1 - 1/r)/2 - m.$$

Maximising pointwise, $q_H = r(1 - c/\theta)$ and $(1 - \phi)[\theta(1 - q_L) - c] = \phi \theta q_L(1 - 1/r)$, which gives the stated q_L . Since $r(1 - \phi) < r - \phi$ for $r > 1$, $q_L < 1 - c/\theta$. The two-part tariff is a special menu, and flat pricing is a two-part tariff with $p = 0$, which gives the dominance ordering when $m = 0$.

Proposition 4. With a fixed fee transferring surplus, the contract minimises the deadweight loss from mispriced usage plus the two risk premia:

$$\mathcal{L}(p) = \frac{X(c - p)^2}{2\theta} + \frac{\rho_B}{2}p^2\sigma^2 + \frac{\rho_S}{2}(c - p)^2\sigma^2.$$

The first-order condition gives $p^* = c(X/\theta + \rho_S\sigma^2)/(X/\theta + (\rho_B + \rho_S)\sigma^2)$. Writing $a = X/\theta$ and $s = \sigma^2$, the derivatives of p^*/c have the signs of $-\rho_S s - a$ for ρ_B , $\rho_B s$ for ρ_S , $\rho_B s$ for a and $-a\rho_B$ for s .

Proposition 5. The provider bears $(c - p)aK$ and chooses e to maximise $-(c - p)a(\mu_K - e) - e^2/(2k)$, so $e^* = ka(c - p)$. Joint surplus is $v - ca(\mu_K - e^*) - e^{*2}/(2k) - \frac{1}{2}(\rho_B p^2 + \rho_S(c - p)^2)a^2\sigma_K^2$. Its derivative in p is $a^2[-kp - \rho_B p \sigma_K^2 + \rho_S(c - p)\sigma_K^2]$, which vanishes at the stated p^* . Comparing with the Borch level, $\rho_S\sigma_K^2/(k + (\rho_B + \rho_S)\sigma_K^2) < \rho_S/(\rho_B + \rho_S)$ for $k > 0$.

Proposition 6. Write $B = \rho_B\sigma_K^2$ and $S = \rho_S\sigma_K^2$. Substituting e^* into joint surplus and comparing $p^* = cS/(k + B + S)$ with $p = c$ gives

$$W(p^*) - W(c) = \frac{a^2c^2(k + B)^2}{2(k + B + S)} = G,$$

so the outcome-based contract is preferred if and only if $\kappa < G$. G is increasing in ac , and $\partial G/\partial k$ has the sign of $2(k + B)(k + B + S) - (k + B)^2 > 0$.

Corollary 1. For equal risk parameters, the copilot pass-through exceeds the Borch level because $X/\theta > 0$, and the agent pass-through falls short of it because $k > 0$.

Proposition 7. $AC = F/Q + c_0Q^{-b}$ and $MC = (1 - b)c_0Q^{-b}$, so $S = (1 + FQ^{b-1}/c_0)/(1 - b)$. For $b < 1$, $Q^{b-1} \rightarrow 0$.

Proposition 8. (i) In the symmetric equilibrium of Salop (1979), each firm's best response gives $P = c + t/N$. Hence $dP/dc = 1$, and gross profit per unit $P - c = t/N$ does not depend on c . (ii) With homogeneous products, the low-cost firm prices at its rivals' cost c and serves the market, earning δ per unit.

Proposition 9. Each of N symmetric firms serves L/N customers at markup t/N , so gross profit is Lt/N^2 . Free entry sets $Lt/N^2 = F$, so $N^* = \sqrt{Lt/F}$ and gross profit per firm equals F , independent of c and L .

Proposition 10. The customer's reservation price for a task is w_j , its labour cost. Competition among providers sets the price at $c + t/N$, so the equilibrium price is the smaller of the two. Gross margin under anchoring is $1 - c/w_j$, which is decreasing in c . Revenue integrates the price over automated tasks.