

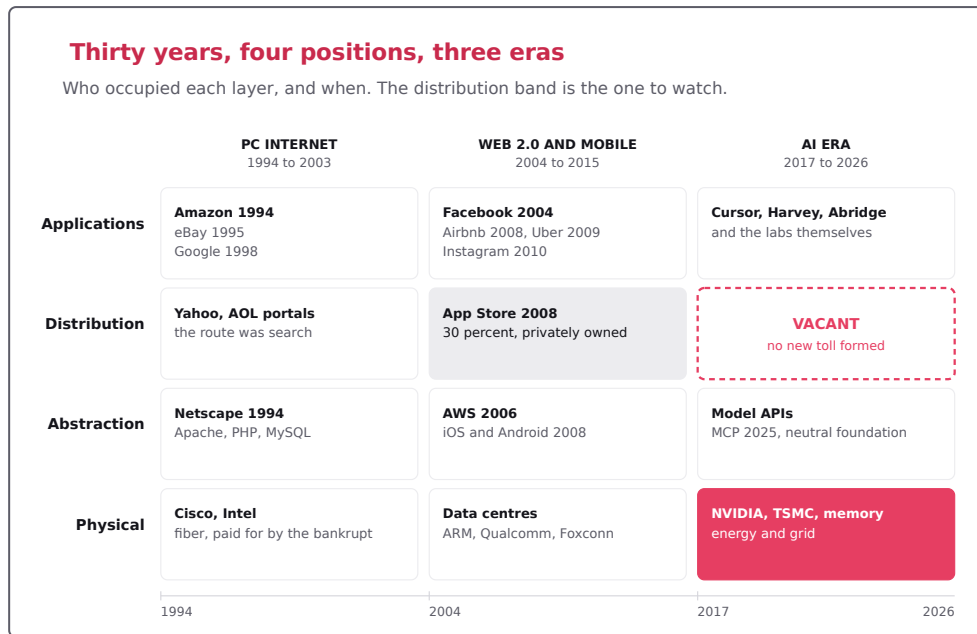
What the AI Era Broke

*How AI changed the value structure Web 2.0
built.*

ABOUT 5,000 WORDS · SIX EXHIBITS · AUGUST 2026

Every technology era since 1994 has produced the same four positions. A scarce physical input at the bottom. An abstraction layer above it. A route to the user. Applications on top. AI has all four, and behaves nothing like the era that produced the playbook everyone is still running.

One difference is worth flagging before anything else, because it is the only one nobody can argue with. In every previous era, infrastructure capacity could be improved with capital. For AI, that is not the case. Only a technology breakthrough moves it, and inventions do not have delivery dates.



Every era filled all four positions. In the AI era no new distribution position has formed, and the bottom band has become a physical gate rather than a purchasing decision. Tolls are still being paid, but to booths built in the previous era. Those two changes drive everything that follows.

The disparity is not a matter of degree. In Web 2.0, the platform needed developers and therefore protected them. In the AI era, this is not the case. Adding your best feature costs the platform a few weeks of engineering rather than an acquisition.

Everything below follows from that.

01

What each era was actually for

Web 2.0 had a single job: move the physical world online.

Shopping moved from a shop to a page. Documents moved from a filing cabinet to a server. The transport of information moved from letterbox to fax to email.

Money, transport, dating, work, all of it went from something you did in a place to something you did on a screen. Every large company of that era is an instance of one migration.

AI is not doing that, because it is already done. The information is already online. What changed is the retrieval.

In Web 2.0 you searched and chose. In the AI era a machine searches, chooses, and hands you one answer.

Exposure is now selective, and the selection is no longer manual.

The obvious objection is that social media already did this. A feed is a filter, and nobody chose what was in it either. The difference is what the filter returns.

A feed returns a ranked list of sources. You can see who published each one, click through, disagree, and notice that something is an advertisement because it is labelled. An AI answer returns a single synthesised claim with the sources removed and the discarded material invisible. A feed hides the ranking. An answer hides the corpus.

That is a meaningfully larger amount of control, and it sits with whoever owns the model.

Four rules in Web 2.0

The migration produced a playbook. For a decade it worked. That is why it became instinct.

- 1 **Acquire users, monetize later.** One more user cost you nothing to serve, so there was no penalty for waiting to charge.
- 2 **Growth is an asset before revenue.** A user who paid nothing was still worth something, because they might bring revenue later and cost nothing to keep in the meantime.
- 3 **Build on the platform.** The chain compounded in your favour. More iPhones sold meant more App Store accounts, which meant more people browsing the store, which meant more downloads of your app. The

growth of the platform pulled everything inside its ecosystem up with it, whether or not you did anything.

- 4 **The application layer holds the most value.** It was the execution layer, where the work actually got done. The server was plumbing. The internet was plumbing. The application was the product.

All four depended on conditions that no longer hold.

The toll booth, and what it was actually for

Apple charged thirty percent on the App Store. It could do so because three conditions held at once. There was one route onto the device. The money passed through Apple's hands. The account belonged to Apple. Remove any one and the toll collapses.

The part that gets ignored is what the toll did for the people paying it.

Microsoft owned Windows for two decades and never built the winning search engine, marketplace, or messenger. Apple owned the phone and never built ride hailing, delivery, or photo sharing. This is usually called restraint. It was arithmetic. A platform that taxes an ecosystem cannot compete with that ecosystem, because developers leave, and a platform with no developers has no tax base.

*The toll booth was not a tax on the ecosystem.
It was the reason the ecosystem was left
alone.*

02

None of that logic survives the AI era

Run Apple's three conditions against the AI era and all three fail in the middle of the chain. There is no single route to the user. Money does not pass through one set of hands. Identity is fragmenting rather than consolidating. The Model Context Protocol, the closest thing this era has produced to a distribution standard, was donated to a neutral foundation under

the Linux Foundation in December 2025, with every major lab on the board. A layer that is neutral from birth has nothing to withhold.

So there is no App Store equivalent in this value chain, and there is not going to be one. It is not late. The position cannot form.

Two objections arrive immediately, and both are correct.

Apple still takes thirty percent. Every AI subscription sold through an iOS app pays the same toll that existed in 2008. The old booth did not close when the new era arrived. It kept charging, and AI companies are among its better customers.

In enterprise, the route to the user belongs to whoever already had the install base. Microsoft ships Copilot into roughly a billion Office seats without anyone downloading anything. That satisfies all three of Apple's conditions at once: one path, custody of payment, ownership of identity. It is a functioning toll booth.

So the precise claim is narrower than it first sounds, and it is the one that matters. **No toll booth native to this era has formed.** Nobody has become the App Store of AI. The tolls being collected are either inherited from the previous era or held by incumbents who already had distribution before this one started. Nothing new opened up in the middle of this chain for a new company to occupy.

Which should mean nobody is collecting a toll, and that anyone can build on top of these models without paying rent to a gatekeeper. That is how most people currently read the situation, and it is why so much capital is aimed at the application layer.

It is wrong. The toll is being collected. It just moved.

The API is the toll booth, and it sits at the bottom

An API key is the ticket you buy at a toll booth, except the booth sits underneath everything rather than beside the road, and it charges on every trip rather than on entry. OpenAI and Anthropic are paid three ways at once: consumer subscriptions, business API usage, and, increasingly, advertising. The API line is the one that matters structurally, because it is the only one that scales with somebody else's success.

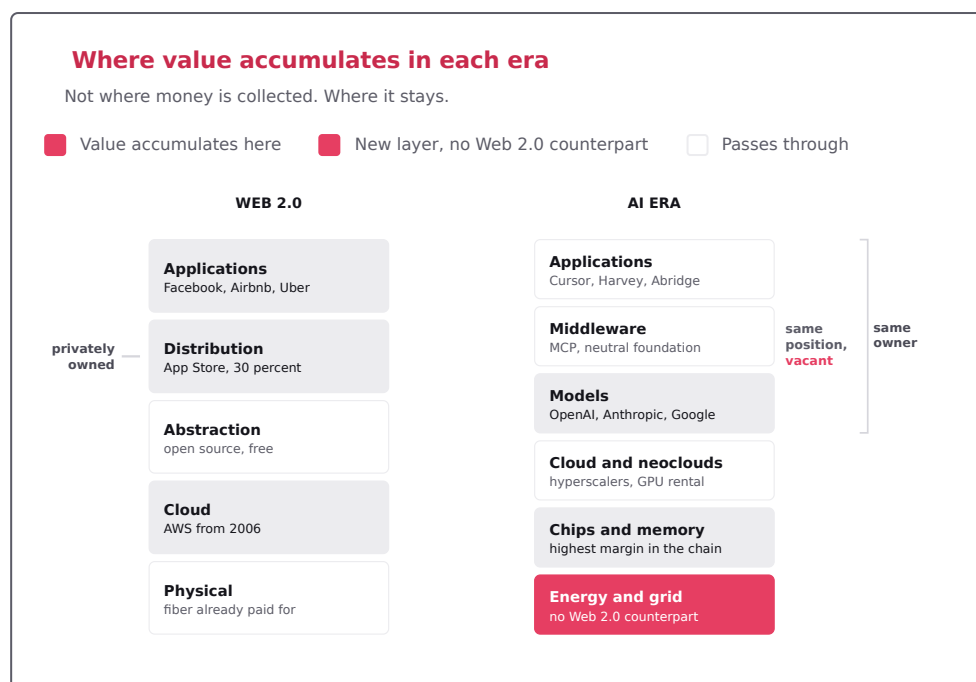
The exclusion is narrower than it looks

Enterprise traffic being kept out of training does not mean the supplier learns nothing from it. Content and metadata are different things, and only one of them is contracted away.

What stays visible without reading a single prompt: which categories consume the most tokens, how fast that consumption is growing, which ones renew, and which ones expand from a pilot into a real contract. That is a market map with a revenue column attached, updated monthly, covering every company in a category at once.

Web 2.0 platforms had a version of this. Apple could see App Store revenue. But that arrived after a product had already worked, a lagging measure of a finished thing. Consumption is a leading one. It is visible while the product is still being built, and it is visible to the party best placed to build the same thing in a few weeks.

The companies on top are not only customers. They are a distributed, self-funded programme for discovering which uses are worth owning.



Web 2.0 accumulated value at two points and left the middle open for anyone. AI accumulates at the bottom, where physics limits supply, and at the model layer, which also occupies the application layer above it.

There is a second difference in how the two eras charge, and it is easy to miss because it looks like a pricing choice rather than a structural one.

TikTok never charged you for using the platform. Nor did Facebook, Instagram, or WeChat. They gave the product away, accumulated hundreds of millions of users, and monetized afterwards through advertising, creator fees, and marketplace take rates. Users were free to acquire because they were free to serve.

AI charges on day one, like enterprise software. A ChatGPT subscription is priced the way Notion and Slack are priced, because the cost structure is the same: every use consumes something. There is no version of this business where the user is free.

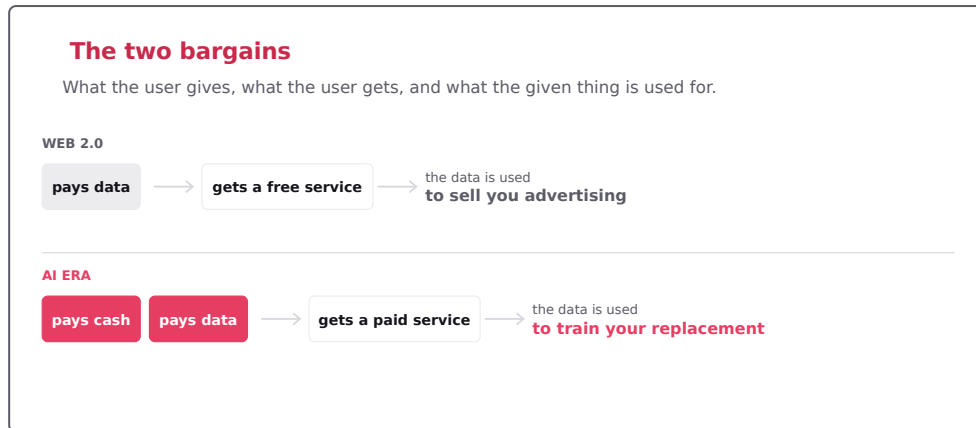
03

Users are being taxed twice

The Web 2.0 bargain was legible and, on its own terms, fair. You paid in data. You received a free service. Your data was used to sell you advertising.

Both sides understood the trade.

The AI bargain has two payments.



The structural difference is not that AI charges money. It is what the second payment is used to build.

You pay cash for the subscription. You also pay in data, because your prompts, your corrections, your workflows and your judgement about which answer was right are among the highest quality training signals that exist.

The second payment is not universal, and the distinction matters more than it is usually given credit for. Contracted enterprise traffic is generally excluded from training by default, and zero retention terms are available on request. A bank running an API contract pays once, in cash. Consumer subscriptions and free tiers are where the data payment happens, and that is also where the volume is.

So the two payments land on different people. Businesses pay cash. Individuals pay cash and data. And the return on the individual's second payment is not a better advertising experience.

Web 2.0 users paid in data and got a free service. AI users pay in cash and in data, and what the data trains is the thing that replaces them.

This is not a moral complaint. It is the reason the old rules break. Under Web 2.0 logic, a user was an asset because a user was future revenue at zero present cost. Under AI logic, a user costs money to serve on every interaction, and simultaneously contributes to a system whose improvement reduces what that user can charge for their own labour.

The common formulation is that users no longer equal profit. That is imprecise. Users still equal revenue. What has changed is that the value a user generates flows to the model, not to the product sitting on top of it.

04

Which means the only question about data is whether it evaporates

If data is the second currency, the obvious response is to accumulate it. Most of what gets accumulated is worthless. One question separates the two.

THE TEST

Would this data exist if you had not captured it?

If yes, it is worth nothing. Everyone can get it, and a stronger model will extract more from it than you will.

If no, it is worth a great deal. It was evaporating, you caught it, and no model however strong can regenerate a moment that is gone.

Two examples make the line visible.

A browser captures browsing behaviour. Browsing behaviour already leaves a complete digital trace across servers, logs and analytics. Building a new browser to capture it adds cost and no information, which is why OpenAI shut down Atlas in July 2026 and Google shut down Project Mariner in May 2026 after seventeen months. Both were well built. Both were capturing something that already existed.

A recorder in a medical examination room captures a conversation that has never been written down anywhere and disappears the moment the

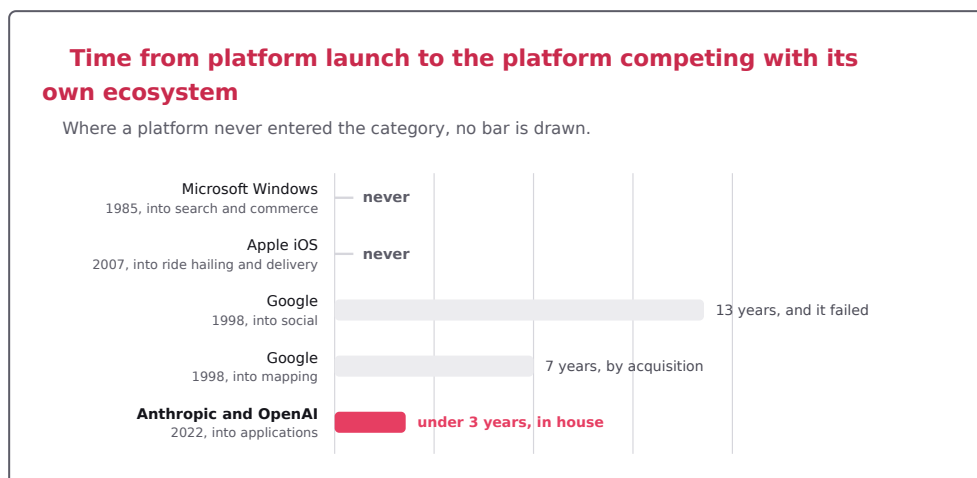
appointment ends. Abridge, which does exactly this, is valued in the billions. The product is not a better interface. The product is the only copy.

Everything else sorts against that line. Code in a repository, tickets in a system, emails in an inbox: already recorded, worth little. A contractor's verbal instruction on a site, a shift handover, a sales conversation in a customer's office: evaporating, and therefore worth a great deal. Permission to record those is hard to obtain for exactly the reason they are valuable.

05

**Absorption used to cost an acquisition.
Now it costs a few weeks of
engineering.**

This is the variable that explains the rest.



Google needed seven years and an acquisition to enter mapping, and thirteen years to fail at social. The model labs arrived in under three, building in house.

Google was founded in 1998 and did not ship Maps until February 2005, and only after acquiring Where 2 Technologies the year before. It attacked social in 2011 with Google+ and lost. Apple has never built ride hailing or delivery.

The sharpest case is Microsoft and Notion. Microsoft has owned the document category since the 1980s. It has Word, OneNote, SharePoint, Teams, a billion seats of distribution, and more engineers than Notion has customers. Notion's core ideas, blocks, databases and linked pages, were not secret. Microsoft could have absorbed them at any point in ten years and did not, and Notion became a multi billion dollar company inside the gap.

Not because Microsoft was slow. Because absorbing a category meant a new product, a new team, a new codebase and a multi year roadmap fight, weighed against a market that was small next to Office. The arithmetic said no.

Not because these companies lacked ambition. Because entering an adjacent category meant building a different product, with a different team, on different technology, against an incumbent with a head start.

For a model lab, entering an adjacent category means writing a system prompt, defining a few tools, and putting a surface on a model it already owns. Anthropic shipped Claude Code and then Claude for Legal. OpenAI shipped Codex. When a popular third party client called OpenClaw became a meaningful surface on top of Claude Code, Anthropic absorbed the useful behaviour into its own product and then, on 4 April 2026, cut the subscription quota those users were relying on.

The mechanism is not aggression. It is cost.

THE COMPARISON THAT MATTERS

For Google in 2004, adopting a feature was more expensive than acquiring the company that had it. For Anthropic in 2026, adopting a feature is cheaper than an acquisition, and cheaper than the meeting about the acquisition. When the price of absorption falls below the price of negotiation, nothing gets bought. It gets absorbed.

Which raises the question the entire application layer is currently avoiding.

The boundary, before this goes too far

The obvious objection is that the companies supposedly being absorbed are not dying. Cursor is one of the fastest growing software companies in history, and Anthropic ships a competing product. Harvey keeps raising at higher marks, and Claude for Legal exists.

Both things are true, because they run on different clocks.

TWO CLOCKS

The feature can be absorbed in weeks. The account cannot. Procurement, security review, on premise deployment, forty integrations into internal systems, and a relationship with the person who signs the contract: none of that is compressed by a model release.

That gap is where a company can still be built. It is also closing, because every year the labs get better at enterprise sales, and every year more of the integration work that used to take humans months is done by agents in an afternoon.

If absorption is cheap, what is the moat for anyone who does not own a frontier model?

One boundary still holds, and it is the only reliable one observed so far. **Platforms absorb what makes them look smarter and decline what makes them liable.** Code generation was absorbed because it is horizontal, consumes enormous volumes of tokens, and doing it well is a direct demonstration of model quality. Legal work was absorbed halfway: retrieval, drafting and review were taken, standing behind a lawyer's error was not. A significant share of Harvey's valuation is not software. It is that somebody is accountable.

Wrapper versus layer

Both words get used as insults and neither is well defined. They fail differently and it matters which one you are.

Two positions, two failure modes

	WRAPPER	LAYER
What it sells	A feature the model does not have yet	Access to something it does not own
Why it works	The model is inconvenient to use directly for this task	The owner finds arranging access tedious
How it dies	The lab ships the feature. The users leave the same week.	The owner decides arranging access is worth doing
Cost control	None. Your input price is set by the company that will replace you.	Partial. You can multi source, but the margin is thin.
If inference gets cheap	Worse, not better. A thin layer over free inference has no cost advantage left to sell.	Worse. Arranging access to something abundant is not a business.

The last row is the one founders get backwards. Falling API prices are read as good news for anyone building on top of a model. For a wrapper the opposite is true.

A wrapper's implicit product is that using the raw model directly is annoying and expensive. Cheaper API prices remove the expense, so half the pitch is gone. Every model release removes some of the annoyance, so the other half goes too. A layer dies the same way for a different reason: arranging access to something abundant is not a business.

Neither position is a mistake to build. Both are a mistake to confuse with a company.

The real trap is subtler than building the wrong thing. A wrapper makes money. Money feels like proof that you have something defensible. It is not proof of anything except that the model does not have your feature yet. And while the money is coming in, nobody feels any urgency to build the thing that would still be standing after the model ships it.

There is a third box, and it is where the survivors are

Cursor is not a wrapper and it is not a layer. Neither is Harvey, or Abridge, or Sierra. They sell things a model could reproduce, and they are not dying, which means a two box taxonomy is incomplete.

The third position is a company whose defensibility is the account rather than the feature. What it owns is not something the model lacks. It owns the deployment: the integrations, the compliance posture, the workflow the customer rebuilt around it, the data the customer has already put inside it, and the person who would have to explain the switch to their board.

THE TEST

If the lab shipped your exact feature tomorrow, free, how long would it take your largest customer to leave?

A wrapper's answer is a week. A layer's answer is a quarter. An account company's answer is measured in years, and by then it has built something else.

This position is real and available today. It is also the one that decays fastest as models get better at doing integration work themselves, which is the part most people building it are not pricing in.

But what about open weights?

The table above says your input price is set by the company that will replace you. That is only true if you must use a frontier API.

Open weight models from Meta, DeepSeek, Alibaba and Mistral are adequate for a large share of production workloads, and they can be self hosted or rented from a dozen inference providers competing on price. A company running its own weights controls its cost, cannot have its quota cut, and cannot be repriced overnight. Those are real advantages, and an essay that ignored them would deserve to be ignored.

They solve the wrong problem.

Self hosting fixes input price and supplier risk. It does nothing about absorption, which is the thing that actually kills you. If a lab ships your feature into a product that hundreds of millions of people already have open, your inference bill is not what you lost. And self hosting adds back the costs the API was hiding: GPU capacity, serving reliability, evaluation infrastructure, and a capability gap against the frontier that widens every few months and now has to be closed by you rather than by your supplier.

Open weights are a cost strategy. They are not a moat. The two get confused constantly, usually on the slide where a defensibility argument should be.

07

The labs are making money. They are not optimising for it.

Subscriptions, API contracts and now advertising all generate real revenue. What the labs are not doing is maximising it, and two decisions make that visible.

OpenAI shut down Sora. The consumer video product reached roughly a million users and was closed in 2026 while it was still popular, taking a large media partnership with it, on the stated grounds that

compute was better spent elsewhere. A company optimising for revenue does not close a product with a million users. A company optimising for a research race reallocates the GPUs.

DeepSeek says so directly. Founder Liang Wenfeng, in an interview with 暗涌 Eras: "Our goal isn't quick profits but advancing the technological frontier" and "Providing cloud services isn't our main goal, achieving AGI is." The company has kept its frontier weights open and priced aggressively while saying plainly that neither is a monetisation strategy.

The following is an assumption and is stated as one, because it cannot be verified from outside. The labs appear to believe that whoever reaches general intelligence first captures a disproportionate share of everything downstream. If that belief is correct, optimising for revenue today is irrational, because any feature, product or vertical can be absorbed in a few weeks by whoever gets there, and the profit follows the absorption.

Read through that assumption, four otherwise confusing decisions become obvious:

- Shipping products that compete with your best paying customers is fine. Those customers are a rounding error against the prize.
- Donating the protocol layer to a foundation is fine. Owning distribution matters far less than owning the model.
- Pricing below cost is fine. Usage is training signal, and training signal is progress.
- Frontier prices rising rather than falling is not a broken cost curve. It is revenue being redirected into the race instead of into margin.

For anyone building on top, the practical consequence is that the platform's roadmap is not a business decision you can negotiate with. It is a bet,

and you are inside the blast radius of it.

08

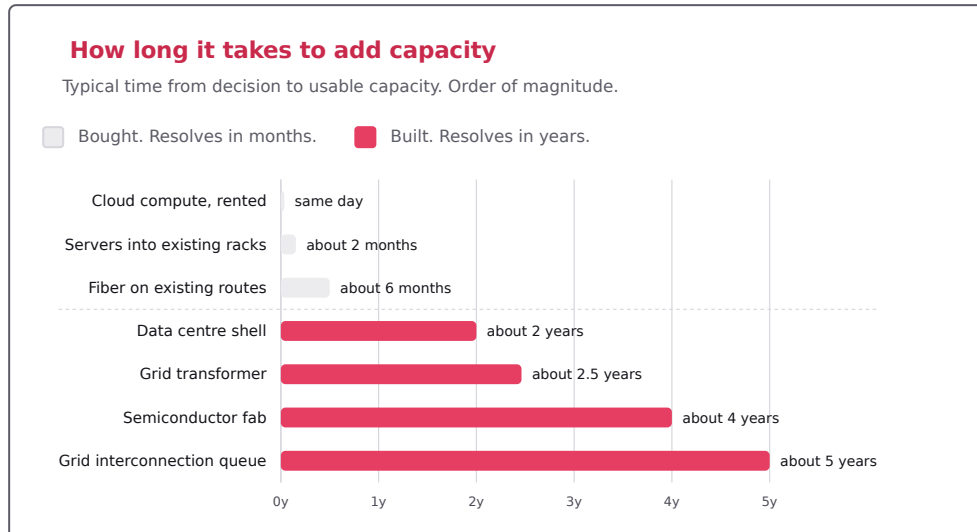
Three things worth forecasting

Where the labs' business model goes

Subscriptions and API usage are the present. Advertising has already started. The one that would change the structure of the industry is a take rate on transactions completed by agents.

If a model completes a purchase on a user's behalf and the lab retains a percentage, the toll booth reappears in the middle of the chain after all, and it is owned by the same company that owns the bottom. That is the single development most worth watching, and it would invalidate the claim in section two rather than confirm it.

What happens when the physical constraint is solved



Grid transformer lead times do not shorten when a buyer offers to pay double, because the constraint is winding capacity, steel grades and trained labour. The exit is not capital. It is a technology breakthrough, and inventions do not have delivery dates.

Web 2.0 scaled because money solved infrastructure. Need more servers, buy more servers. This era cannot do that, and software has never faced this before. Every other industry has. Railways waited on steel.

Electrification took four decades. Fabs have always been gated by construction schedules. Software is close to the only major industry in history that was exempt from physics, and its entire vocabulary, ship fast and scale on demand, was written during that exemption.

If the constraint breaks, whether through efficiency per watt, packaging yields, small modular reactors, or algorithmic gains, the effect is not gradual. Everything currently blocked by arithmetic rather than by capability becomes viable at once: continuous observation instead of sampling, twenty models running one task with a judge above them, generating twenty options so a person selects rather than specifies, reading everything instead of retrieving fragments.

Value then moves toward whoever owns demand rather than supply, because supply stops being scarce. Wrappers and layers both get worse rather than better, for the reason in section six: a wrapper sells relief from an expense that has gone away, and a layer sells access to something that is no longer scarce.

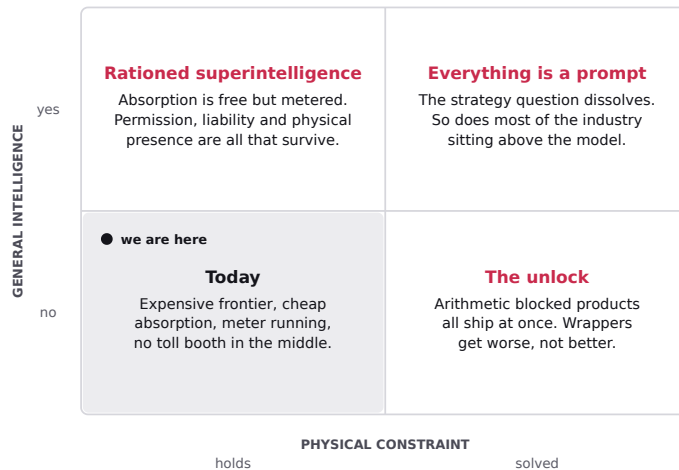
What happens if general intelligence arrives

The cost of absorption goes to zero rather than merely low. Every product becomes a prompt away. Section five's question stops being a strategy problem and becomes the only problem.

What survives is what a model cannot obtain regardless of how strong it is: permission that has to be negotiated one counterparty at a time, liability somebody has to accept in person, data that evaporates unless captured, and physical presence in a room.

Two variables, four futures

Both axes are observable. Neither is predictable.



The upper right quadrant is the one nobody has a strategy for, because it is the one where the strategy question dissolves.

09

The question this leaves open

Two facts sit uncomfortably together.

Innovation in software has historically come from small companies, because small companies were cheaper to run and faster to try things than incumbents. And the cost of an incumbent copying a small company has now fallen from an acquisition to a few weeks of engineering, and is still falling.

There are two honest answers to what happens next and no way to choose between them yet.

The first is that innovation relocates. It moves to where models cannot follow: regulated markets, physical settings, permissioned data, work that requires somebody to be accountable. Slower, smaller, more defensible, and a much less interesting industry than the one that produced Instagram in a year.

The second is that innovation stops being a company. It becomes a feature request. The returns concentrate entirely in whoever owns the model, and

the thousands of people currently building on top are performing unpaid product research for four laboratories.

If being copied costs a competitor a few weeks of engineering, and that competitor is also your supplier, who also sets your input price, what exactly is a startup in this era?

Appendix

Assets that appreciate as models improve

The test for any asset is whether a ten times stronger model makes it worth more or worth less. Four survive it, and each carries a cost that is usually left unstated.

Adoption capacity, meaning the ability to change how an organisation actually works, was on this list and has been removed. It appreciates for a while, then stops. It is headcount shaped, it carries a low multiple, and the integration and change management work it consists of is precisely what agents are getting good at. It is a business, not an asset that outruns the meter.

ASSET	WHY IT APPRECIATES	WHAT IT COSTS
Permission and data agreements	Negotiated one counterparty at a time, so it cannot be bought at speed by anyone in a hurry	Growth capped by negotiation. Naturally regional.
Accepted liability	The stronger the model, the more consequential the work, and the more valuable a party willing to be accountable	Insurance, legal capacity, surviving one bad event
Evaporating data	No model can regenerate a moment that is gone. Time is the only way to accumulate it.	High front loaded collection cost. Stop and it decays.
Verification and provenance	Answers now arrive without sources. Checking became expensive exactly as it became skippable.	Unclear who pays. No unit of account.

Kaus AI · London · August 2026. Figures are drawn from public filings, company announcements and named interviews; approximate values are marked as such in the text.