Applied AI

A position on how organisations should build with AI, and who should own what.

COVELENT · JULY 2026

BUILDING IS NO LONGER THE BOTTLENECK

Since the first software ran on a stored-program computer in the late 1940s, building it was slow and expensive. Because building was the constraint, the valuable work was deciding what to build. Strategy, analysis and advice commanded a premium because execution was costly and uncertain, and much of the consulting industry grew up on that fact.

That constraint has now shifted. A single practitioner working with frontier models can stand up a working system in days that would once have taken a team a quarter, and we've done it repeatedly, across defence, property and capital markets. The length of task an AI can complete on its own has roughly doubled every seven months for six years,[1] and the benchmarks that tracked the last decade of progress now saturate faster than researchers can write new ones.[2]

When the cost of building collapses, so does the premium once paid to decide what to build. The scarce inputs are no longer analysis and headcount, which AI now supplies in quantity, but judgement about what's worth building and ownership of the result once it exists. An organisation that rents both has bought motion without control.

When building is cheap, the value moves to judgement and ownership.

This paper sets out what we think follows. AI is an accelerant, not a strategy. It widens the gap between what a customer will pay and what it costs to serve them, but it doesn't invent that gap. The economics of the firms most organisations rely on are badly suited to it, and durable advantage comes from owning the stack that intelligence runs on, not from renting the intelligence. Applied AI is the discipline of making that stack reliable, governed and owned; Build-to-Own is the delivery model we use to do it.

> **We build on the frontier. We think you should own what results.**

THE ARGUMENT IN BRIEF

1. **Building is no longer the bottleneck.** A single practitioner on frontier models ships in days what once took a team a quarter, and the length of task AI can do unaided has **doubled every seven months** for six years. Value moves to judgement about what to build, and ownership of the result.
2. **The frontier is commoditising.** Capability is roughly **280× cheaper** and **142× smaller** in two years, and the best open models now trail the closed frontier by about one release cycle. It's a rented input, not a moat.
3. **The incumbents are structurally exposed.** Advice-by-the-hour and software-by-the-seat sell exactly the work AI automates, and in the first half of 2026 the market marked the most exposed names down, several by **more than half**.
4. **Renting is temporary on every axis.** Access sits on the provider's and the government's terms, and the price itself is investor-subsidised, by **40 to 70×**, with one leading provider spending **$1.69 for every dollar** it earns. It's set to rise.
5. **Build-to-Own is the durable position.** In our survey of senior leaders, roughly **seven in ten** of the biggest blockers to production AI were about control, not capability. Build fast on the rented frontier, then hand over an owned stack, the harness, the ontology, local models and data inside your perimeter, so a shift at the frontier is an upgrade, not an outage.

1 METR, *Measuring AI Ability to Complete Long Tasks*, Mar 2025. metr.org

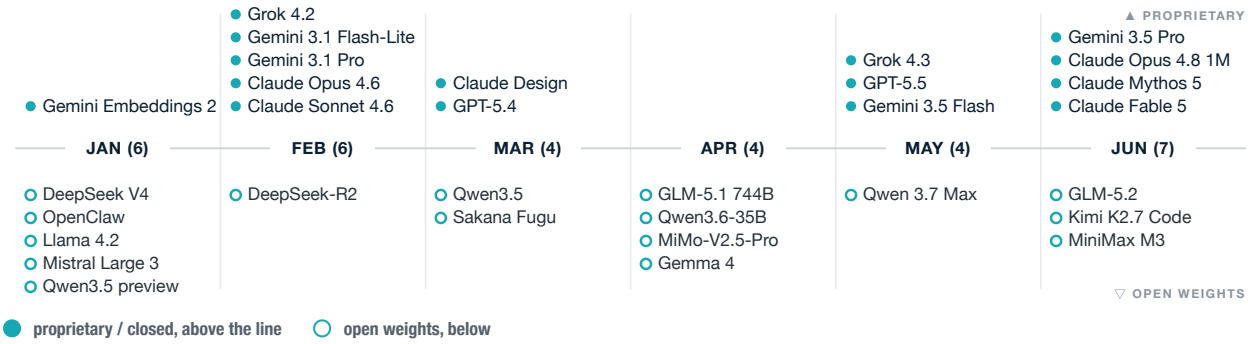
2 Stanford HAI, *AI Index 2025*, Technical Performance. hai.stanford.edu

THE LANDSCAPE

Capable new models now ship almost every week, faster than any enterprise can adopt

This pace isn't anecdotal, and the six-year record shows it. The length of task an AI agent can complete unaided has grown exponentially, doubling roughly every seven months.[1] The benchmarks that tracked progress, from MMLU to GSM8K, have saturated, so researchers now build harder ones because models exhaust the old ones.[2] Driving all of it is the sheer volume of release. Capable models ship almost every week, and no team can keep pace. Below is a fraction of what landed in the first half of 2026 alone.

FIGURE 1 · FRONTIER MODEL RELEASES, JANUARY TO JUNE 2026



Frontier and notable open-weight models released between January and June 2026, listed by month; the bracketed figure is that month's count. A partial view: hundreds more quantised, distilled and fine-tuned open models shipped alongside them, and the pace has not slowed since, with GPT-5.6 Sol among the flagships landing in July. Source: public launch announcements.

No organisation can evaluate all of this, let alone adopt it. But three shifts sit beneath the noise, and each changes who can build and at what cost. Capability is getting cheaper, smaller and more open. The direction is clear, and it's the premise of everything that follows. The frontier is a fast-moving, commoditising input, not a moat you can rent forever. The advantage isn't in having access to it, because soon everyone will. It's in what you build on top, and whether you own it.

1 METR, *Measuring AI Ability to Complete Long Tasks*, 2025.

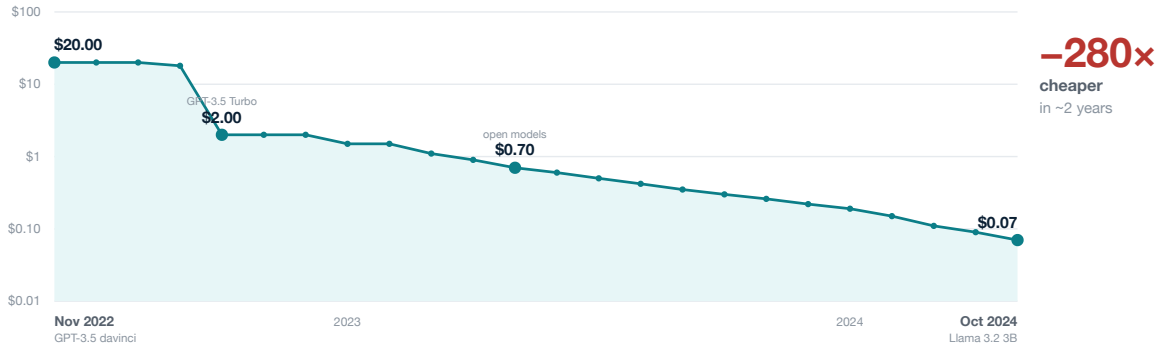
2 Stanford HAI, *AI Index 2025*, Technical Performance.

THE LANDSCAPE · IN NUMBERS

Capability got 280 times cheaper and 142 times smaller in two years

Three forces pull the same way. Capability that once cost a fortune, filled a data centre and lived behind a single provider's API is becoming cheap, small and open, each on the trajectory below.

FIGURE 2 · COST TO QUERY GPT-3.5-LEVEL INTELLIGENCE, PER MILLION TOKENS



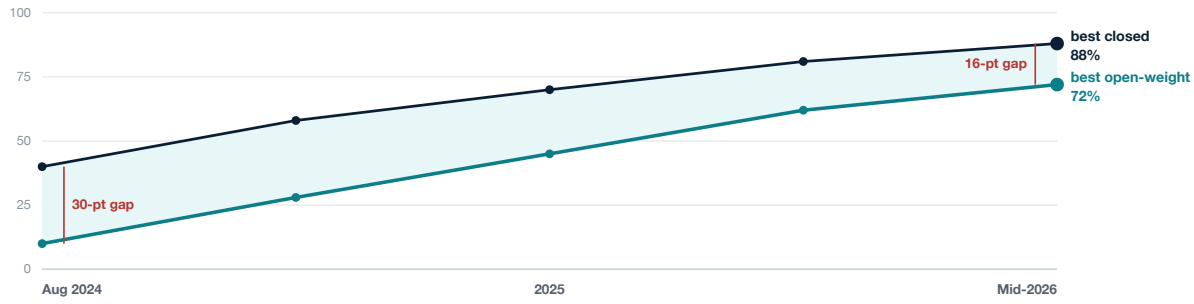
A 280-fold fall in about two years, from $20.00 to $0.07 per million tokens. The line tracks the cheapest price to reach GPT-3.5-class quality, month by month, as each new tier of models undercut the last.[3] Independent analysis puts the broader trend near fifty times cheaper per year.[4] Log scale.

Source: Covelent analysis of Stanford HAI and Epoch AI data.

FIGURE 3 · SMALLEST MODEL TO CLEAR 60% ON MMLU, BY PARAMETER COUNT

The same capability, 142× smaller, in two years: PaLM at 540 billion parameters in 2022, Microsoft's Phi-3-mini at 3.8 billion by 2024.[5] Capability that required a data centre now runs on commodity hardware. Log scale.

FIGURE 4 · CODING CAPABILITY, BEST OPEN-WEIGHT VS BEST CLOSED MODEL, SWE-BENCH VERIFIED



On coding, scores have soared and the gap has closed at the same time. The best open-weight model has climbed from around ten per cent to the low-70s on SWE-bench Verified in two years, against a closed frontier now in the high-80s, and the lead has narrowed from roughly thirty points to the mid-teens.[6] Open models such as GLM-5.1 and DeepSeek V4 already top individual coding and reasoning benchmarks.[7] Percentage of issues resolved.

Source: Covelent analysis; trajectory illustrative, on SWE-bench data.

3 Stanford HAI, *AI Index 2025* ($20.00 → $0.07 / M tokens).

4 Epoch AI, *LLM inference price trends*, 2025.

5 Stanford HAI, *AI Index 2025* (PaLM 540B → Phi-3-mini 3.8B).

6 SWE-bench Verified, 2024 to 2026; best open-weight vs best closed, % resolved (illustrative). swebench.com.

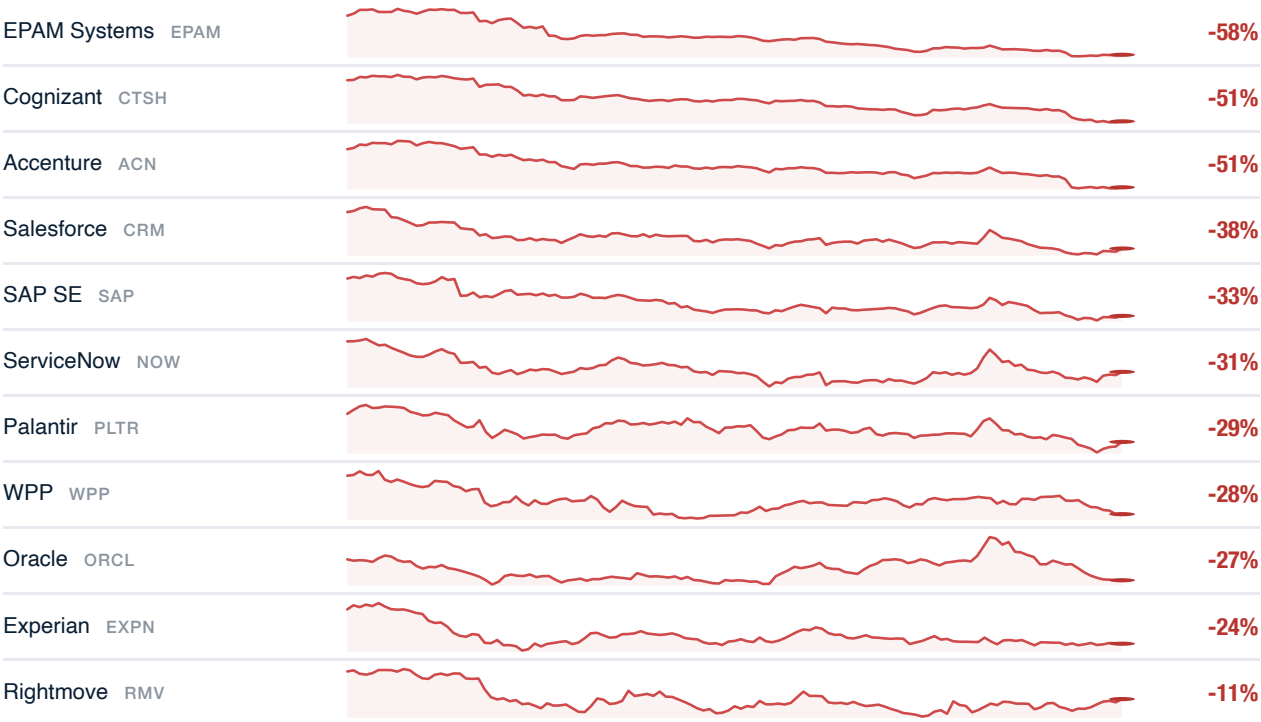
7 SWE-bench Verified & SWE-bench Pro leaderboards, 2026 (open-weight low-70s vs closed high-80s; GLM-5.1 and DeepSeek V4 lead individual benchmarks). benchlm.ai; openrouter.ai.

THE MARKET

In the first half of 2026, the market marked down the AI-exposed incumbents, several by more than half

The clearest independent read on where value is moving is the market itself. Through the first half of 2026, investors sold down the enterprise-software and IT-services providers whose revenue AI most directly threatens. These are the firms whose business is selling seats, licences and billable delivery of exactly the work that's now automatable, and they aren't fringe names. They're the sector's bellwethers (the CRM, the ERP suite, the systems integrator, the ratings and data provider), so for most organisations the names below read as a fair approximation of their own vendor roster, and the multi-year contracts most firms hold are with exactly the names now under the most pressure.

FIGURE 5 · YEAR-TO-DATE SHARE PRICE, ENTERPRISE SOFTWARE AND IT SERVICES, 2026



Year-to-date share price, 1 January to 1 July 2026; each chart on its own scale. Source: Yahoo Finance. Names shown are widely held enterprise software, IT services and data providers.

We don't read a single half-year as destiny, and share prices are noisy. But the de-rating fits the argument of this paper. The market is repricing the model of renting people and software to do work that can increasingly be built once and owned, and it is doing so before the substitution has even fully arrived. Two operating models sit behind those names, and the pages that follow take each in turn.

OPERATING MODELS · ENTERPRISE SOFTWARE

When one agent does the work of ten people, software priced by the seat stops adding up

Two operating models sit between most enterprises and their own systems, and AI is repricing both. The first is the software an organisation rents to run itself, and its economics are simple. Sell access by the seat and the licence, bill it every year, and design the product so that leaving is expensive. Revenue depends on two things holding, seat count and renewal rate, and for two decades both have.

AI puts pressure on both at once.

An agent that does the work of ten people doesn't need ten seats, and a capability you build and own doesn't need a licence at all. The same technology that lets a business do more with less also lets it rent less, and a model priced by the seat was never designed for a world where the work itself can be automated and the tool can be owned.

The incumbent response is rarely to cut the price. It's to deepen the moat. Bundle AI into the suite so it looks free, raise switching costs, and use the account relationship to steer customers back towards the approved, integrated, rented option and away from open or self-hosted alternatives. The pressure to stay is presented as prudence (sound architecture, security, support). Often it's protection of the vendor's own position, and it's worth naming as such.

Put the two prices side by side.

The arithmetic isn't subtle. A licence is priced per person who might touch the workflow, bought across the headcount whether or not each seat is used, and renewed every year whatever the usage. An agent that does the same work is a one-off build and then a small, falling cost per task on hardware you own. The vendor's revenue is a function of your headcount. Your cost, once you own the thing, is a function of the work actually done. Those two lines were always going to diverge, and what has changed is that the technology to separate them is now in general release.

It also explains why the incumbent response is a bundle rather than a discount. A discount concedes the new price. A bundle defends the old one, and buys time to move the customer onto terms that survive the transition.

The rented model has a texture the buyer knows well.

Anyone who has run a large software estate knows the shape of it. The licence audit that surfaces just as the renewal is negotiated, the seat true-up when headcount ticks up, the modules bought in a bundle and never switched on, the price that steps up once a migration has made leaving impractical.[1] None of this is a failure of the product. It's the model working as designed, and the same account teams now arrive with an AI upsell bolted onto the same terms.

1 The Register, *Oracle ULA audits are a license to bill*, May 2024; Competition and Markets Authority, cloud services market investigation, 31 Jul 2025 (licensing, egress and switching-cost dynamics).

Consulting sells hours, and AI is deleting them, a third already gone at some firms

The second model is the people an enterprise rents, and it works the same way one level up. The economics of a professional-services firm are straightforward. Profit comes from leverage (a wide base of junior staff billed out beneath a thin layer of seniors) and from utilisation (keeping those juniors on billable engagements as long as possible).[1] Compensation tracks the hours billed,[2] and for decades it has worked.

The model was never only about the advice.

The defining programmes of the last two decades (digital transformation, cloud migration, ERP rollouts) were sold as three-to-five-year, multi-million-pound engagements. Their real product was rarely a single recommendation, but a standing team of external consultants embedded inside the client. With that team went the knowledge the work created, product knowledge, domain knowledge and infrastructure knowledge alike. When the engagement ends, the understanding of how the enterprise actually runs leaves with the supplier. The advice was outsourced, and so was the knowledge, which is the harder dependency to unwind.

That's the design of the model, not a flaw in it. A client that can't see how its own systems work can't easily change course, or change supplier, and the next phase always feels necessary. Letting a client build the answer itself, in weeks, and keep the knowledge is **anathema** to a business built on embedding and hours.

AI turns this from lucrative to untenable.

When work that took a team a quarter can be done by one person in a week, the billable-hour engine runs in reverse. The very tools that raise productivity shrink the revenue those hours produce.[3] Some firms already report their own AI tools saving close to a third of a consultant's time,[4] yet compensation still tracks hours, and the model still depends on the client never owning what was built. This isn't a criticism of the people. It's arithmetic.

It's worth being precise about what's under pressure. Strategy and advisory (genuine judgement about what to do and why) is a distinct and durable discipline. It can stand on its own, it's often the precursor to what follows, and it's work we do ourselves. What the economics above describe is something else. It's the multi-year transformation, migration and rollout programmes that are long by design and slow to deliver on their original promise. It's that model AI turns from lucrative to untenable, not the counsel that precedes it. Building with AI needs neither a standing team nor an open-ended programme, but a small number of senior people who carry a problem all the way to a working, owned system (knowledge included) and who are paid for the outcome rather than the hours. If the firms that sold slow building are exposed, though, the organisations that buy from them have barely moved either.

1 David Maister, *Managing the Professional Service Firm*, 1993.

2 D. Duncan, T. Anderson & J. Saviano, *AI Is Changing the Structure of Consulting Firms*, Harvard Business Review, Sep 2025 (the pyramid giving way to the "obelisk").

3 Thomson Reuters Institute, *The $2,000-hour problem*, 2025.

4 Consultancy.uk, *Is AI about to kill the billable hour?*, Jan 2026.

The technology has arrived, yet almost nine in ten enterprises have not scaled it

Everything so far describes the supply of capability, and it is plentiful, cheap and getting cheaper. The demand side tells a different story. To test it, we ran a pulse survey of senior leaders at large enterprises, and the picture is consistent. The technology has arrived, and almost nobody has scaled it. The bottleneck has moved from building to adopting, and adoption has stalled.

> **Covelent AI Adoption Pulse.** n=113 senior decision-makers (CIO, CTO, CDO, CISO, General Counsel and PE operating partners) at enterprises of $200m to $80bn revenue; United States (46), Europe (35), United Kingdom (24) and Middle East (8); fielded 1 June to 6 July 2026.

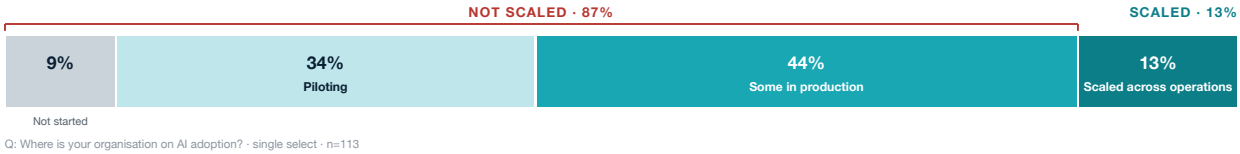
87%
have **not** scaled AI across operations

62%
have been stuck in pilots for six months or more

75%
had an AI initiative paused by legal, compliance or risk in the last year

70%
judge themselves behind their own 12-month ambition

FIGURE 6 · WHERE ENTERPRISES ACTUALLY ARE ON AI ADOPTION



Q: Where is your organisation on AI adoption? · single select · n=113

Only one enterprise in eight has scaled AI across operations. The rest are piloting, running isolated production use, or have not begun. Nearly nine in ten sit short of scale, and most of them are stuck: three in five have been at the pilot or evaluation stage for six months or more.

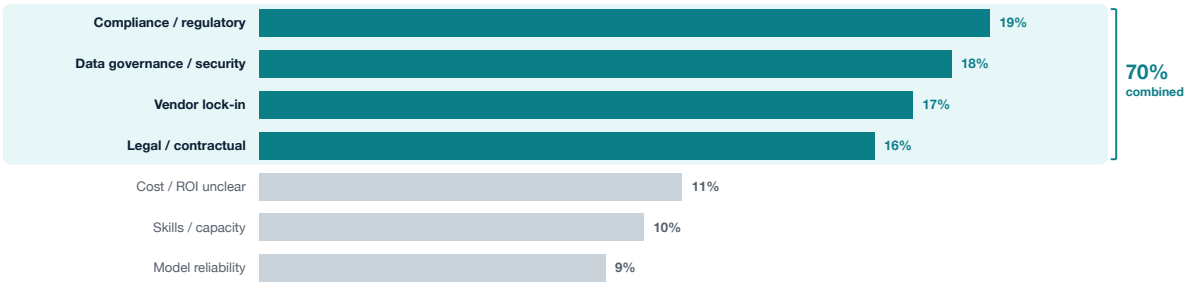
Source: Covelent AI Adoption Pulse, 2026 (n=113).

This is the gap between what the frontier now makes possible and what organisations have actually captured, and it isn't a capability gap.

For seven in ten enterprises, the blocker is control and governance, not the technology

Asked for the single biggest blocker to moving AI into production, the leaders didn't point at the models. They pointed at control. Compliance, data governance, vendor lock-in and legal exposure (four faces of the same question of who governs the system) together account for roughly seven in ten of the biggest blockers named,[1] far ahead of cost, model reliability or skills. What stands between a large enterprise and production AI isn't whether the technology works, but whether the organisation can run it on terms its own risk, legal and compliance functions will accept.

FIGURE 7 · THE SINGLE BIGGEST BLOCKER TO PRODUCTION AI



Q: single biggest blocker to moving AI into production? · single select · n=113, of those answering · rounded to whole percentages

Control, not capability, is the binding constraint. Compliance, data governance, vendor lock-in and legal concerns, all questions of who governs the system, are the biggest blocker for roughly seven in ten enterprises, far ahead of cost, model reliability and skills.

Source: Covelent AI Adoption Pulse, 2026 (n=113).

The blockers are not separate. Lock-in feeds the rest.

Vendor lock-in doesn't sit apart from the governance concerns. It sits upstream of them. The software and services incumbents an enterprise already depends on have every reason to keep it inside their ecosystem, and they use the compliance conversation to do it, steering customers towards the approved, integrated, rented option and warning that open-weight or self-hosted models are risky, unsupported or non-compliant, often out of concern for their own market position as much as the client's security. The stakeholders advising on how to adopt AI safely are frequently the ones with the most to lose if the enterprise owns it. Compliance, legal exposure and lock-in reinforce one another, and together they keep the system rented.

Nobody else has measured this.

Three quarters of the leaders we surveyed had an AI initiative paused or blocked by legal, compliance or risk in the last twelve months. We went looking for someone else's figure to check ours against, and there isn't one. No government statistical series, no regulator survey and no independent study we could find puts a number on how often governance stops an AI project, or on how long it holds one up. Adoption is measured exhaustively. The friction that determines adoption is not measured at all, which is its own comment on how the problem is understood.

Ask them what would unlock it, and they answer with control.

The single most-requested unlock isn't better models or a clearer business case. It's a **clear compliance and governance framework**, named first, with sovereign or private infrastructure and approved access to owned data close behind, even as most admit they couldn't say what they'd pay if a provider repriced or withdrew a model tomorrow. The enterprise isn't asking for more frontier. It's asking to own and govern what it runs.

1 Covelent AI Adoption Pulse, 2026. n=113 enterprise decision-makers; fielded 1 Jun to 6 Jul 2026. Percentages use populated responses as the denominator.

THE DISCIPLINE

The scarce skill is making AI reliable, compliant and owned, not knowing what it can do

Almost everything written about AI is about what AI can do, which is the easy part now, and not where the difficulty or the value sits. Every serious field has this divide. In physics, one half works out what's theoretically possible, and the applied half makes it hold up in the world (at cost, under real constraints, in something a person actually uses). These are different disciplines, and the second is where things get built.

Applied AI is our name for that second half. It's not a strategy deck, and not a pilot that never leaves the lab. It's the engineering work of taking what the frontier makes possible and turning it into systems a business runs on, connected to real data, inside real governance, doing real work, and owned by the organisation that depends on them.

This matters because the market is full of the first half and short of the second. Knowing that a model could, in principle, automate a workflow is now common. Standing that workflow up so it's reliable, compliant, auditable and yours is the scarce skill, and most of the job.

Most of that work sits outside the model, in two places. One is the **harness**, the code around the model that gathers context, calls tools, checks the output and decides when to stop; it now moves measured performance more than the choice of model does, and it has a section of its own later. The other is the **ontology**, the connected layer underneath that gives an agent something reliable to reason over, and it is the centrepiece of what Build-to-Own hands you. We have done exactly this on our own account, in one case turning a business's scattered records into a connected knowledge layer of several thousand entities in two days.

And building this way is no longer exceptional.

AI has set off a measurable boom in how quickly software is made. In the past twelve months the number of new AI projects created on the largest public code platform grew by 178 per cent year on year, close to seven hundred thousand of them, after a 98 per cent rise the year before, an acceleration rather than a one-off spike.[1] Around 85 per cent of developers now build with AI tools, on two independent surveys.[2] The barrier to turning an idea into a working, owned system has fallen through the floor. What the rest of this paper sets out is why the thing you build should be owned, what owning costs against renting, and how we deliver it.

1 GitHub Octoverse, 2024 and 2025 (+98% then +178% year on year; ~694,000 new AI projects in twelve months).

2 Stack Overflow Developer Survey 2025; JetBrains State of Developer Ecosystem 2025 (~85% building with AI tools).

You cannot own what you rent

Building with frontier models is now within reach of any organisation, but depending on them is a different matter. Access to an external model is access on someone else's terms, and those terms are set by the provider, the market, and increasingly by governments with competing priorities. What renting really costs you is sovereignty, the ability to decide for yourself, on your own timeline, what your business runs on.

Renting is not one option of two. It is the default, overwhelmingly. In our own survey, **69 per cent of senior decision-makers said their AI runs all or mostly on third-party and cloud APIs, against 4 per cent mostly on models they own.**[1] The question is not whether to move from owning to renting. It is whether an enterprise ever intends to move the other way.

The risk isn't hypothetical. In July 2026, ByteDance and Alibaba disabled user-created humanlike AI agents to comply with new Chinese rules on anthropomorphic AI,[2] and capabilities that businesses had built on those platforms disappeared on a regulator's timetable. External providers retire models on their own schedule too, so forced migration is a routine feature of renting, not an exception.[3]

The intervention cuts from both directions.

In June 2026 the US Commerce Department ordered Anthropic to suspend its two most capable models three days after launch, under export-control powers, and access went dark worldwide for eighteen days.[4] They returned only once the directive was withdrawn, and then with a new blocking classifier that fenced off some of the very capabilities that had set them apart.[5] Now the pressure runs the other way. As this paper goes to press, the administration is weighing restrictions on domestic firms' use of Chinese open-weight models, through procurement rules and Entity List pressure rather than an outright ban.[6] Since most leading open-weight models are now Chinese, that would push American enterprises back onto a handful of closed providers. A closed model can be switched off on a government's timetable, and the open one can be closed off by the same hand. Whichever way policy turns, the renter's options are chosen for it.

The everyday version is already familiar.

The core systems enterprises rent show the same grip. Oracle's licensing and audit regime is built to keep customers renewing rather than leaving,[7] and it can reprice access on those already locked in. When its Java terms moved to a per-employee model, analysts put the rise at two to five times the old cost.[8] Salesforce customers sit inside an ecosystem projected to earn its partners over six dollars for every dollar Salesforce makes.[9] Even cloud carries egress fees and switching barriers the UK competition regulator found to harm competition across a £9 billion market.[10]

This is why ownership matters, and why the real word for it is sovereignty. Ninety per cent of organisations already regard data held locally as inherently safer,[11] and regulation from the EU AI Act onward is moving sensitive workloads inside the perimeter.[12] Breaches involving unsanctioned, external AI cost around $670,000 more than the average.[13] A business that owns what it runs, on its own hardware and inside its own network, keeps control whatever an administration decides next. A business that rents has handed that decision away. The frontier can be rented to build. What runs the business should be owned.

1 Covelent AI Adoption Pulse, 2026 (n=113). Q: how much of your AI today runs on third-party or cloud APIs versus models you own or host?

2 South China Morning Post, 5 Jul 2026 (China Interim Measures on anthropomorphic AI).

3 OpenAI model deprecations. developers.openai.com/api/docs/deprecations

4 Forbes, 16 Jun 2026; National Law Review, Jun 2026 (Commerce Dept export-control directive; Fable 5 and Mythos 5 launched 9 Jun, suspended 12 Jun).

5 Axios, 27 Jun 2026; TechCrunch, 1 Jul 2026 (directive withdrawn 30 Jun; redeployed 1 Jul with a new blocking classifier).

6 TechCrunch and CNBC, 24 Jul 2026 (US weighs restrictions on Chinese open-weight models; Nvidia, Microsoft and Meta warn against premature restrictions); US NTIA, *Dual-Use Foundation Models with Widely Available Weights*, 2024.

7 The Register, *Oracle ULA audits are a license to bill*, May 2024.

8 The Register, *Oracle's new Java license terms*, Jul 2023.

9 IDC (Salesforce-commissioned), *The Salesforce Economy*, 2021.

10 Competition and Markets Authority, cloud services market investigation, final decision summary, 31 Jul 2025 (egress fees and technical barriers found to harm competition in a £9bn market).

11 Cisco, *2025 Data Privacy Benchmark Study*.

12 European Commission, EU AI Act, GPAI obligations (from 2 Aug 2025).

13 IBM, *Cost of a Data Breach 2025* (shadow-AI breach premium).

The frontier lead you are told is permanent is already down to a single release cycle

The usual objection to owning your stack is that the frontier is too far ahead and too fast-moving to keep up with in-house. That's true today, and it's exactly why we rent it to build. But the frontier you can see isn't the real one. Labs never stop iterating, and anything released publicly implies something better already in training, so the public models are a lagging indicator of where the frontier actually is. And the gap that makes renting worthwhile is closing on three fronts at once.

The models are commoditising.

The best open-weight model trailed the best closed one by eight per cent in early 2024, and by under two per cent a year later.[1] On a single capability index, the best open models now lag the closed frontier by roughly four months (about one release cycle), a gap that's shrinking, not widening.[2] The frontier also leaks. In June 2026 Anthropic told the US Senate Banking Committee that operators tied to a Chinese lab had run some 28.8 million exchanges through about 25,000 fraudulent accounts to distil cheaper models from Claude's own answers, even though the leading labs had themselves trained on the public internet without consent.[3] What's frontier and rented today is open and local tomorrow.

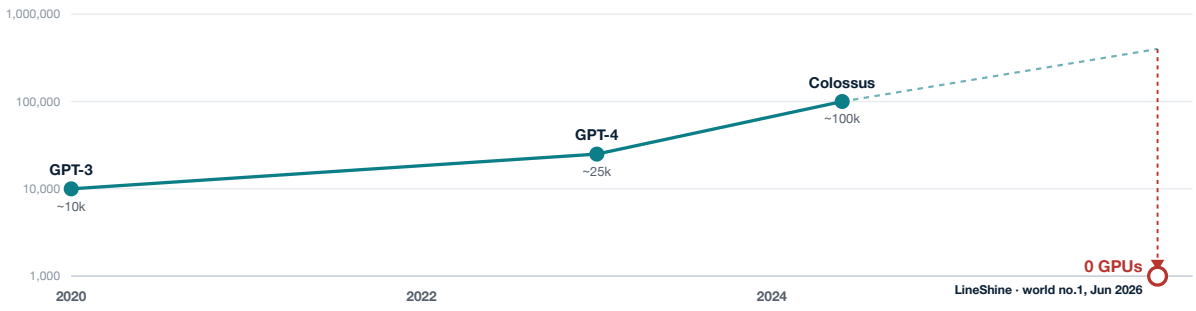
And they are getting cheaper and smaller.

Capability is getting cheaper by roughly fifty times a year[4] and smaller by more than a hundred times for the same benchmark,[5] to the point where one-bit models now run efficiently on ordinary CPUs, with no specialised accelerator required.[6]

The hardware is diversifying off a single vendor.

NVIDIA still supplies the overwhelming majority of AI accelerators, but the largest labs are deliberately spreading their bets. Anthropic says it runs Claude across three chip platforms (Google TPUs, Amazon Trainium and NVIDIA GPUs),[7] and OpenAI, NVIDIA's largest customer, is designing its own accelerators with Broadcom and sourcing memory from Samsung and SK Hynix.[8] When the suppliers of compute are themselves reducing their dependence, an enterprise's dependence on a single rented stack looks less like prudence and more like concentration risk.

FIGURE 8 · GPUS PER FRONTIER TRAINING RUN, AND THE MACHINE THAT USED NONE



Reported and estimated accelerator counts for notable training runs and clusters. In June 2026 China's LineShine took the world number-one spot from El Capitan at 2.2 exaflops with no GPUs at all, on 13.8 million CPU cores;[9] NVIDIA still powers more than 400 of the world's 500 fastest machines. Log scale.

Source: Covelent analysis of reported and estimated cluster sizes.

Put together, these trends point one way. The rented layer thins, and over time even the build comes in-house. The one thing that doesn't change is ownership. Nothing rented is ever owned, so the durable position is to own what runs, and to have built it so that a change at the frontier is an upgrade rather than an outage.

1 Stanford HAI, *AI Index 2025* (Chatbot Arena, 8.04% → 1.70%).

2 Epoch AI, Capabilities Index, 2026 (open models ~4 months behind the closed frontier).

3 Anthropic, letter to the US Senate Banking Committee, 10 Jun 2026; Forbes and Tom's Hardware, 25 Jun 2026 (~28.8m exchanges via ~25,000 accounts, late Apr to early Jun 2026).

4 Epoch AI, *LLM inference price trends*, 2025.

5 Stanford HAI, *AI Index 2025* (142× parameter reduction).

6 Ma et al., Microsoft Research, *The Era of 1-bit LLMs*, arXiv:2402.17764; CPU inference arXiv:2410.16144.

7 Anthropic, *Expanding our use of Google Cloud TPUs*, 23 Oct 2025. anthropic.com

8 OpenAI & Broadcom, 10 GW collaboration, 13 Oct 2025; Samsung/SK Hynix memory, Oct 2025.

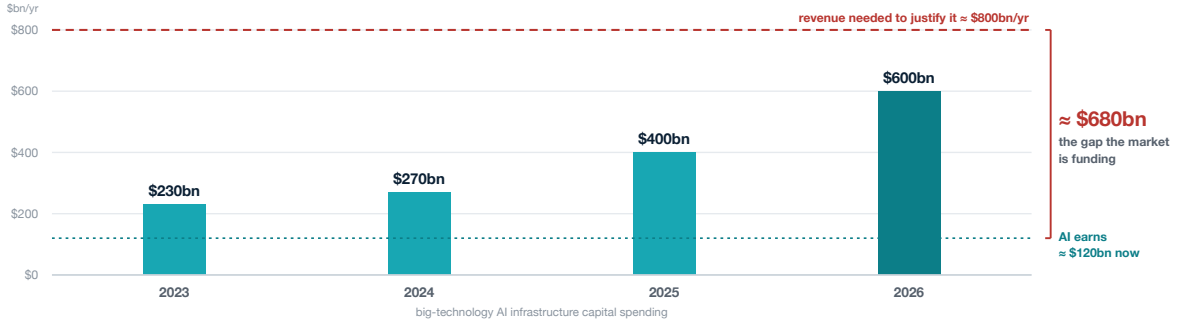
9 TOP500, 67th list, Jun 2026 (LineShine, 2.198 exaflops HPL, 13.79m CPU cores, CPU-only); Al Jazeera and Network World, 24 Jun 2026.

AI is spending about $680 billion a year it has not yet earned

There's a deeper reason to own rather than rent, and it's the price itself. The rented price isn't merely temporary, it isn't even real. Behind today's cheap compute sits a build-out with no precedent, and an arithmetic that doesn't yet close. In 2026 the largest cloud providers are guiding to roughly $600 billion of capital expenditure between them, most of it for AI, up from about $230 billion three years earlier.[1] Nvidia, which supplies most of the chips, booked around $194 billion of data-centre revenue in its last financial year, a run-rate growing above sixty per cent.[2] This is the fuel that lets compute be sold cheaply today.

The problem is what it implies. A widely cited framework from one venture investor takes the chip run-rate, doubles it for the rest of a data centre, and doubles it again for a normal software gross margin, to estimate the annual revenue the build-out must eventually earn to pay for itself. In mid-2024 that produced a gap of roughly half a trillion dollars a year between what was being spent and what AI was earning.[3] Since then the spending has climbed and the gap has widened, financed, for now, by investors buying a market.

FIGURE 9 · AI INFRASTRUCTURE SPENDING, AND THE REVENUE IT IMPLIES



Spending has multiplied; the revenue to justify it has not arrived. Big-technology AI infrastructure capital spending by year, against the annual revenue that build-out implies it must eventually earn: roughly $800bn, against about $120bn earned today. The distance between the two, close to $680bn a year, is what today's cheap prices are borrowing from.

Source: Covelent analysis of company filings, venture-capital estimates and IEA data.

None of this predicts a collapse. It's a statement about price. Compute sold below cost is compute sold on someone else's balance sheet, and balance sheets get repriced. The organisations most exposed are the ones that have built their operations on the assumption that today's rented price is the real one.

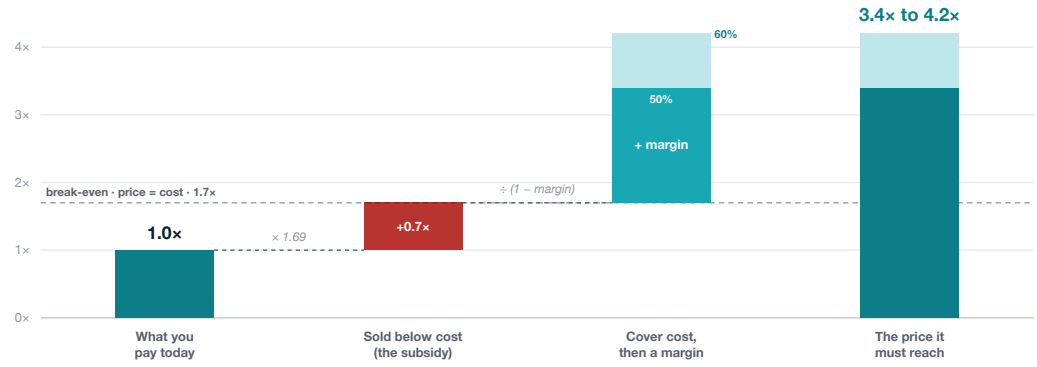
1 Company earnings guidance, 2023 to 2026 (aggregate hyperscaler capital expenditure).
2 NVIDIA, FY2026 results, data-centre segment revenue.
3 D. Cahn, Sequoia Capital, *AI's $600B Question*, Jun 2024.

You rent at a price subsidised 40 to 70 times over, and it has to rise

There's one more reason the rented layer isn't the safe default it looks like. The price isn't real. The subscriptions and API rates that make frontier AI look cheap are being sold below cost, funded by investors buying market share, wooden dollars underwriting the sticker price. In 2025 the market leader spent about **$1.69 for every dollar of revenue** it earned, has told investors it expects to burn on the order of $115 billion through 2029, and has conceded it loses money on its $200-a-month plan because customers use it far more than the price assumes.[1] A product sold below the cost of serving it is a promotion, not a price.

Put a number on the gap. Independent analysis of the leading consumer tiers found a $200-a-month plan delivering $8,000 to $14,000 of API tokens, a subsidy of forty to seventy times the sticker price.[2] Read the other way, and the chart below builds it up. Simply covering cost lifts today's price to about 1.7 times, and a normal software gross margin of 50 to 60 per cent lifts it to between three and four.[3]

FIGURE 10 · COVELENT ANALYSIS: BUILDING THE UN-SUBSIDISED PRICE



From what you pay to what it costs. Indexed to today's price of 1.0×, the market leader's $1.69 of spend per $1 of revenue means simply covering cost lifts the price to about 1.7×, and adding a normal software gross margin of 50 to 60% lifts it to between three and four. The venture "double for the data centre, double for margin" rule agrees, implying the build-out must earn roughly four times the chip run-rate.[3] Illustrative; the ratio and the assumed margin will move, the direction and scale do not.

Source: Covelent analysis of OpenAI spend-to-revenue disclosures, SemiAnalysis subscription economics and venture-capital estimates.

Almost nobody has priced this. Asked whether they knew what they would pay if their AI providers repriced or withdrew a model tomorrow, **only 13 per cent of the senior decision-makers in our survey said they could answer fully. Fifty-one per cent said flatly that they could not, and a further 35 per cent only roughly.**[5] Seven in eight enterprises are carrying an exposure they have never sized, on a price the provider has already said is below cost.

When the labs turn from land-grab to returns, and they're visibly preparing to,[4] the levers are obvious. They can raise prices, meter usage, or cut the limits that make today's plans feel unlimited, and everything rented reprices with them. A system that runs on models and hardware you own has a cost you set and can forecast. A workflow that rents its intelligence inherits whatever price the market needs once the subsidy ends. This isn't an argument against the frontier. It's an argument against building the things that run your business on top of a price that was never real.

1 The Information; SemiAnalysis, 2025 to 2026 (OpenAI spend vs revenue; ~$1.69 per $1; ~$115bn cumulative burn to 2029); S. Altman, Jan 2025 (loss-making on the $200 plan).

2 SemiAnalysis, consumer AI subscription economics, 2026 ($200 plan ≈ $8,000 to 14,000 API value; 40 to 70× subsidy).

3 D. Cahn, Sequoia Capital, *AI's $600B Question*, Jun 2024; standard data-centre cost-of-ownership arithmetic.

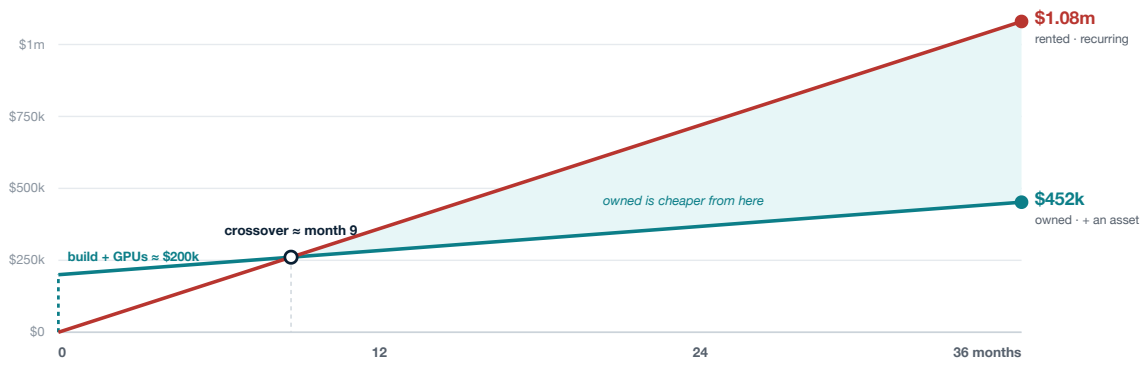
4 The Information, 14 Apr 2026 (Anthropic Enterprise moved to $20/seat plus API pricing); OpenAI Codex pricing aligned to API token usage, 2 Apr 2026, extended to all Enterprise plans 23 Apr 2026. Both removed the enterprise discount to list rates.

5 Covelent AI Adoption Pulse, 2026 (n=113). Q: do you know what you would pay if your AI providers repriced or withdrew a model tomorrow? Fully 13%, roughly 35%, no 51%.

At $30,000 a month rented, an owned build pays for itself in nine months

Every argument so far has a number attached, and for most boards it's the deciding one. Renting and owning aren't just different levels of control, they're different cost structures. Renting is pure operating expense (opex), per-seat software and per-token inference, billed every month, rising with use, and leaving nothing owned when the contract lapses. Building and owning converts most of that into a one-off capital cost (capex, the build and the infrastructure it runs on), then a low, predictable running cost, and it leaves an asset on the balance sheet. On a total-cost-of-ownership basis the two lines diverge quickly, and once you discount the recurring rental the net present value favours owning well inside the first year.

FIGURE 11 · CUMULATIVE COST OF ONE WORKLOAD: RENTED VERSUS BUILT AND OWNED



Illustrative, on public unit costs. A workload renting frontier models and seats at roughly $30,000 a month accumulates cost in a straight line that never flattens. The same capability built and owned, assuming the existing data-centre or cloud footprint most enterprises already have, so the incremental cost is the build and GPUs, about $200,000 then roughly $7,000 a month to run, costs more on day one and less within the first year, with the gap widening every month after. The figures scale with the workload; the shape does not.

Source: Covelent analysis, on public unit costs.

The unit economics are not close.

Rented, the flagships cost $5 to $30 per million tokens.[1] The same inference on an owned open-weight model is a few cents of electricity once the hardware is bought,[2] where the real expense is the server, an eight-GPU machine at $200,000 to $400,000, or reserved cloud at a few dollars an hour.[3] Past a threshold that published analyses put at around $20,000 of API spend a month, or a few hundred million tokens, owning is simply cheaper, and the saving compounds.[4]

Renting buys motion, owning builds an asset.

The rented line is operating expense. It recurs, it rises, and it leaves nothing behind. The owned line is capital plus a small running cost, and what it buys can be depreciated, adapted and reused (an asset, not a subscription). A bespoke tool that retires a six-figure or seven-figure software estate pays back inside a year and then costs almost nothing.[5] And because today's rented prices are subsidised, as we set out earlier, the real crossover arrives sooner than list prices suggest and moves further towards owning as that support unwinds. This isn't an argument to own everything. For spiky, low-volume or non-sensitive work, renting elastic capacity is cheaper, and we'd advise exactly that. Owning is for the systems the business runs on every day, where the meter never stops.

1 Published API list prices, Jul 2026, per 1M tokens (GPT-5.5, Claude Opus 4.8, Gemini 3.1 Pro). pricepertoken.com.

2 Marginal inference cost of a self-hosted open-weight model (energy and compute per 1M tokens).

3 NVIDIA H100 8-GPU server, $200k to 400k; reserved cloud ≈ $1.5 to 10 / GPU-hour, 2026.

4 Self-host vs API break-even ≈ $20k/mo API spend, or ~100 to 256M tokens/mo on premium models, 2026.

5 Salesforce published list pricing, 2026 (per user, per month); saving depends on scope and seats.

An owned stack is 60% chips and 7% power, a depreciating asset, not a runaway bill

Owning is often imagined as a permanent utility bill, but the composition says otherwise. On an independent build of a modern AI data centre, roughly sixty per cent of the lifetime cost is the servers, about thirty per cent the building and cooling, and only around seven per cent the electricity.[1] At the level of a single machine, the accelerators alone are close to sixty per cent of the bill of materials.[2] A full year of power for an eight-GPU server, even at UK industrial rates, is under a tenth of the price of the box. Owning AI is buying a depreciating asset you control, not feeding a meter.

And you don't buy a frontier data centre to run a business. Nobody is proposing to stand up a trillion-parameter generalist in the server room. Most operational work (classification, extraction, retrieval, routing and structured drafting) is handled by small or fine-tuned models, with a larger model orchestrating and stepping in only on the hard residual. The same discipline runs through every layer. On one of our own builds the storage settled on a provider roughly five times cheaper than the default cloud, with no egress fees, chosen because it did the job, not because a preferred-supplier list already approved it.

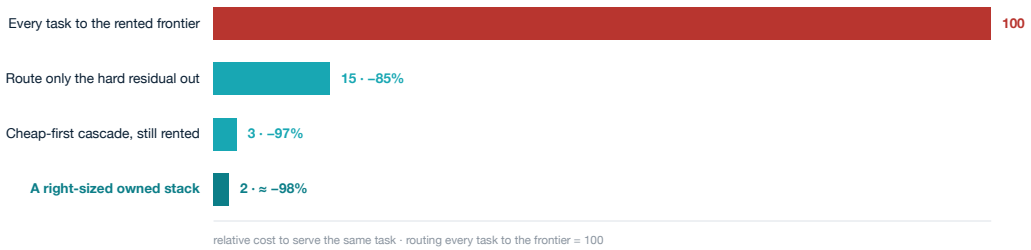
FIGURE 12 · WHAT THE INFRASTRUCTURE COST ACTUALLY IS

Energy is roughly 7% of the total. The cost is the silicon, an asset you own and depreciate.

Chips dominate; power is a sliver. Where the money goes in an owned AI stack: at the data-centre level as lifetime cost of ownership, and at the level of a single eight-GPU server as bill of materials. Energy is roughly seven per cent of the total.

Source: Covelent analysis of Epoch AI and SemiAnalysis data.

FIGURE 13 · COVELENT ANALYSIS: THE SAME WORK, ROUTED DIFFERENTLY



You do not pay the frontier price for work a smaller model can do. Relative cost to serve the same operational task. Sending only the hard residual to a frontier model cuts cost by about 85%, and a cheap-first cascade by up to 97%.[3] Both still meter every tier. An owned stack takes the cheap tiers to zero, a fine-tuned open model on its own task costing little more than the electricity to run it, and meters only the small residual that still goes out. Nothing here needs anything you cannot run inside your perimeter.

Source: Covelent analysis of RouteLLM, FrugalGPT and open-model fine-tuning results.

1 Epoch AI, AI data-centre cost breakdown, 2026 (~60% servers, ~30% facility, ~7% energy).

2 SemiAnalysis, AI server bill-of-materials (accelerators ~59% of an 8-GPU system).

3 RouteLLM (UC Berkeley, 2025); FrugalGPT (Stanford); open-model fine-tuning results, 2024 to 2025.

Rent for breadth. Own for everything else.

Cost is where the argument usually starts, but it isn't where it ends. Set the two side by side and the pattern holds across every axis a business actually cares about, from where your data sits to whether your IP ever leaves the building, what happens when a provider changes its terms or withdraws a model, and whether an auditor can reproduce a decision months later. On each, owning what runs the business is the lower-risk position. The frontier's one decisive advantage, raw general-purpose breadth, is the part you rent to build with, not the part you bet your operations on.

WHAT MATTERS TO A BUSINESS	RENTED · CLOUD LLM	OWNED · RIGHT-SIZED, IN YOUR PERIMETER
Your data	Leaves your perimeter; logged and retained	▸ **Stays in your perimeter**
IP & secrets	Code, prompts and trade secrets sent off-device	▸ **Nothing is transmitted**
Cost at scale	Recurring, per seat and per token, and rising	▸ **One-off hardware, low and fixed to run**
Compliance	Residency a grey zone under GDPR and HIPAA	▸ **In-jurisdiction by design**
Control	Vendor sets price, terms and deprecations	▸ **Your model, your data, your rules**
Continuity	Outages, forced migration, withdrawal	▸ **Runs offline; no external dependency**
Latency	A network round-trip on every call	▸ **Local inference, no round-trip**
Auditability	Black box; changes between one call and the next	▸ **Reproducible and logged**
Breadth	▸ **Broad, general-purpose, frontier-grade**	Focused; fine-tuned to your tasks
Setup	▸ **Instant; no hardware**	Hardware and tuning needed upfront

The edge marks where each option leads. Rent the frontier for breadth and speed. Own the right-sized stack that runs the business.

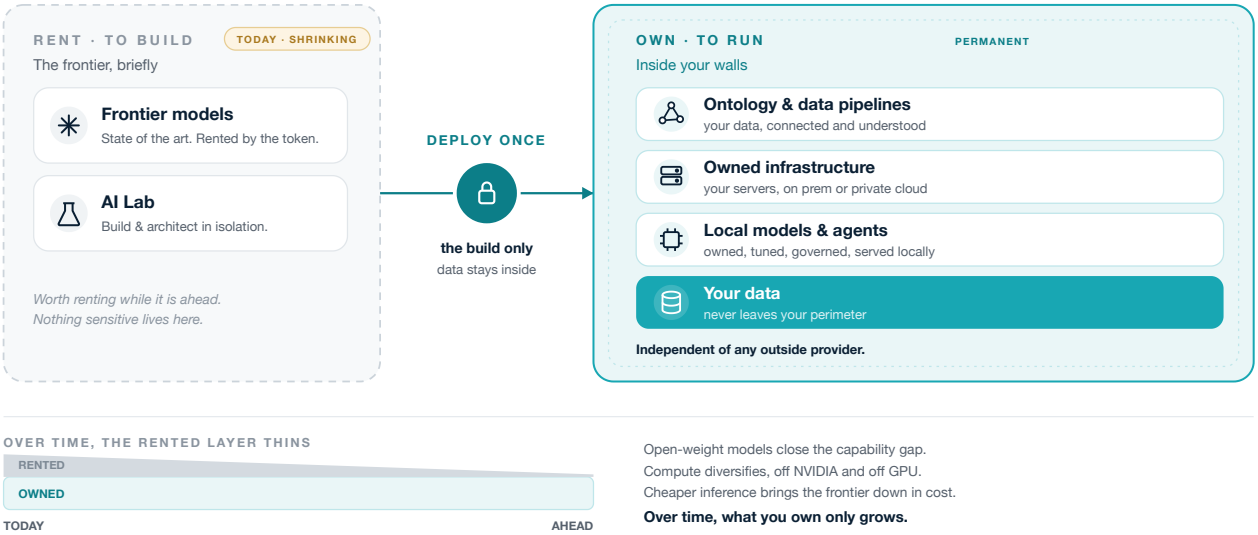
This isn't a case against the frontier. Read the marks and the split is honest. Renting leads where a rented tool should, on raw breadth and on standing something up in minutes, and those are real reasons to rent it while you build. Owning leads on everything you then run the business on. The two are complementary, and the point is to use each for what it's good at, not to bet your operations on the one that answers to someone else.

Several of the rows above rarely reach the cost spreadsheet, yet they decide the outcome. Two are operational. On **latency**, a rented model answers only after a network round-trip, where an owned one runs locally, at conversation speed, with no dependency on a link staying up. On **continuity**, the rented model works only while the provider is available and willing, and models are withdrawn, deprecated and rate-limited on the provider's timetable, not yours, where an owned system keeps running offline through all of it. Two are matters of governance. On **determinism and auditability**, a rented endpoint is a black box that can change underneath you between one call and the next, where an owned model gives reproducible, logged behaviour an auditor can stand behind months later. The quiet one is **exposure**. Every prompt to a rented model carries your code, your data and your intent off your premises, to be logged and, increasingly, used to train the next version. For a regulated business, that isn't a convenience. It's a disclosure.

Build-to-Own builds fast on the rented frontier, then hands over a system you own outright

The case is now made. The frontier is a commoditising rented input, sold at a price that has to rise, and what runs the business should be owned. This is how we do it. **Build-to-Own** is how we practise applied AI, and it sits deliberately outside the two options an enterprise is usually offered. A consultancy advises and leaves, or worse, embeds a dependency it has every incentive to deepen. An internal team builds slowly, inside existing governance and rented software, and rarely ships. We do neither. We build fast on the frontier, and hand over something you own outright.

FIGURE 14 · THE BUILD-TO-OWN MODEL: BUILD RENTED, RUN OWNED



Rent the frontier to build; own everything that runs. The rented side is deliberately thin and temporary, a state-of-the-art model billed by the token and a sealed lab to build in, with nothing sensitive living there. The owned side is permanent and sits inside your perimeter: the harness that drives the models, the ontology that connects your data, the infrastructure it runs on, the local models and agents that serve it, and the data itself. The band below shows the direction of travel, as open models close the gap, compute diversifies and inference gets cheaper, the rented layer thins and what you own only grows.

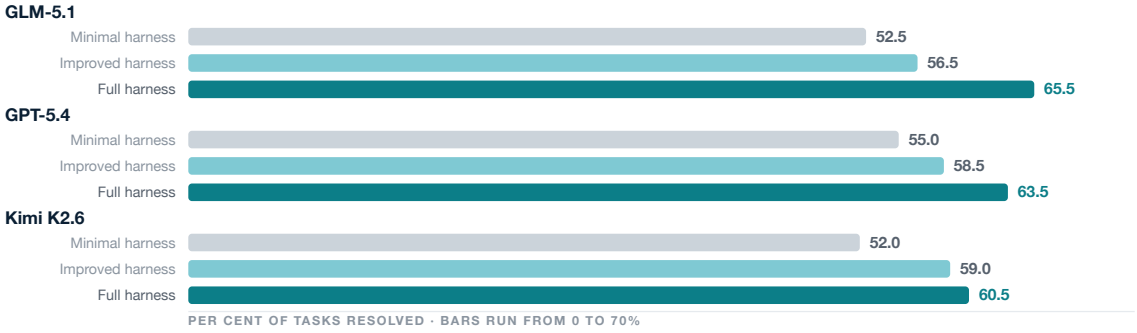
Source: Covelent, the Build-to-Own delivery model.

The principle is in the name. **You cannot own what you rent.** We rent the frontier only to build. For as long as closed models on scarce hardware are ahead, renting them speeds up delivery, so we do our building in a lab, a sealed environment on your own infrastructure where the work graduates from synthetic data to real, and only then deploy once. Everything that runs the business (the harness that drives the models, the ontology that connects your data, the infrastructure it runs on, the local models and agents that serve it, and the data itself) is owned outright and runs inside your perimeter. How it is built, the sealed lab, the agents that do the building, the connected data layer they produce, and how the models are selected and tuned, is set out in a separate delivery pack.

The harness now moves the numbers more than the model does

A model on its own does nothing. What turns weights into work is the **harness** around them: how context is gathered and pruned before the model sees it, which tools it may call, how its actions are executed and checked, when it retries, when it escalates, and when it stops. Recent work decomposes that into six runtime jobs (observation, context, control, action, state and verification) and finds the bottleneck in long-horizon work now sitting at least as much there as in the model itself.[1]

FIGURE 15 · CHANGE THE HARNESS, AND THE RANKING CHANGES



Change the harness, and the ranking changes. Same benchmark subset, three frontier models, three harness configurations. Each model gains 8.5 to 13 points from the harness alone, against the 2.5 to 5 points separating the models inside any one configuration. The strongest model on a minimal harness (55.0) finishes below the weakest on a full one (60.5).

Source: Y. Zhang et al., controlled factorial experiment, 2026.

Same weights, different harness, a different answer.

In a controlled experiment across three frontier models and three harness configurations, changing the harness moved results by 8.5 to 13 percentage points, while changing the model moved them by 2.5 to 5. Six of nine model comparisons reversed their ranking when the harness changed.[2] Published results show the same effect: one frontier model resolves 45.9 per cent of a hard software benchmark under a standardised research scaffold and 55.4 per cent under its own vendor's harness.[2] A benchmark number is a property of a model and a harness together, and published alone it says less than it appears to.

And the gains cost almost nothing.

Because harness work is engineering rather than training, it lands immediately and at almost no marginal cost. Automated tuning of the harness alone lifted a frontier model from 69.7 to 77.0 per cent on one terminal benchmark, with the weights untouched.[2] This is the cheapest performance in the stack, and it accrues to whoever owns the harness.

And a disclosed harness still tells you nothing about your work.

Correcting for the harness fixes the comparison, not the relevance. A public benchmark is somebody else's questions. Only a private set of tasks from your own work, graded against a definition of correct you wrote first, shows which is the smallest permitted system that clears your bar.

Which is why you build your own.

The objection is that the labs' own harnesses are better, and today they often are. The agent frameworks from Anthropic, OpenAI and Microsoft ship more options than any single enterprise would write for itself, and we use them in the lab for exactly that reason. Three things argue against making one of them the thing you run on. First, the orchestration doesn't travel: tool integrations written to the open protocol survive a change of provider, but the workflows, permissions and routing built into a vendor's harness do not.[3] Second, the harness is where your business goes, in which systems an agent may touch and which check must pass before a payment run is released. That is your governance expressed as code, and the last thing to hand to a third party. Third, harnesses are small: the team behind the standard software benchmark publishes an agent of roughly a hundred lines of Python that resolves over 74 per cent of it.[4]

So the harness is the one layer where owning is barely a trade. It is cheap to build, it is where domain knowledge accumulates, and it keeps the model swappable. Rent the frontier to build. Own the harness that runs.

1 J. Guo et al., *From Question Answering to Task Completion: A Survey on Agent System and Harness Design*, Jun 2026. arXiv:2606.20683.

2 Y. Zhang et al. (Tulane, Rutgers, Virginia Tech), *Stop Comparing LLM Agents Without Disclosing the Harness*, May 2026. arXiv:2605.23950 (factorial experiment on a SWE-bench Verified subset; SWE-bench Pro and TerminalBench 2.0 figures).

3 Model Context Protocol specification, 2025 to 2026; OpenAI Agents SDK portable sandbox manifests, Apr 2026; Microsoft Agent Framework, Build 2026.

4 SWE-agent team (Princeton and Stanford), *mini-swe-agent*, 2026 (~100 lines of Python, >74% SWE-bench Verified); All Hands AI, *OpenHands*, MIT licence (~72%, model-agnostic).

THE CASE AGAINST

Take the argument's four weakest points, and none of them says rent

A position paper that only makes its own case is marketing. Here are the strongest arguments against ours, and where they change our advice.

What if the frontier keeps pulling away?

It's possible that closed models stay far enough ahead that renting them remains worthwhile indefinitely, and open weights never quite catch the frontier for the hardest work. If so, the build side of Build-to-Own stays rented for longer than we expect. It doesn't change the conclusion. You still run owned, because the reasons to own (data residency, continuity, and freedom from a provider's terms) are independent of how good the rented model is.

What if owning the stack is simply more expensive?

For some workloads it is. Renting elastic cloud capacity for spiky, non-sensitive tasks can be cheaper than owning idle hardware, and we'd advise exactly that where it applies. But today's rental prices flatter the comparison, because they're subsidised, as we set out earlier, and as that support unwinds the balance tips further towards owning. Build-to-Own isn't a demand that everything move on-premise tomorrow, but a principle about what must be owned (the data, the models and agents that touch it, and the systems the business depends on), and a recognition that the cost of ownership is falling as fast as anything else in this paper.

What if the regulatory picture reverses?

Rules may loosen as easily as they tighten. But the events of the last two years, models withdrawn by governments on both sides of the Pacific, point to more intervention, not less, and an owned system is robust either way. Ownership is the low-regret choice. It costs a little optionality today and removes a large tail risk tomorrow.

What if AI does not even make us faster?

Sometimes it doesn't. In a controlled trial, experienced developers working in their own mature codebases were about nineteen per cent slower with early-2025 AI tools than without, while believing they'd been faster.[1] A field experiment with several hundred consultants found large quality gains on tasks inside the models' reach, but materially more wrong answers on tasks beyond it.[2] This doesn't argue against building. It's the argument for judgement. The value is conditional on knowing which work to hand the machine, which is exactly the scarce input ownership is built around.

We'd rather state these openly than have a prospect find them for us. On balance, none of them recommend renting what runs your business.

1 METR, controlled study of experienced developers, Jul 2025 (~19% slower on their own repositories). arXiv:2507.09089.

2 F. Dell'Acqua et al., *Navigating the Jagged Technological Frontier*, Harvard Business School Working Paper 24-013, Sep 2023 (field experiment, n=758: quality gains inside the frontier, more errors beyond it).

IN CLOSING

The question was never whether to adopt AI, but whether you own what you adopt

Pull the threads together. The frontier is a fast-moving, commoditising input, cheaper and smaller every year, sold today at a subsidised price on someone else's balance sheet, and available or withdrawn on a government's timetable. The firms most enterprises rent from, software by the seat and advice by the hour, sell exactly the work it now automates, and the market has already started to mark them down. Yet almost nobody has scaled AI, and the reason is control, not capability.

The answer is not to rent harder. It is to build with the frontier while it leads, and to own what runs, the models, the data, and the system that ties them together, inside your own perimeter. Owning wins on the axis every board starts with, cost over time, and on the ones they discover later. Continuity when a provider changes its mind, an audit trail an auditor can stand behind, and sovereignty, the ability to decide for yourself, on your own timeline, what your business runs on.

So the real question was never whether to adopt AI. It is whether you own what you adopt, or rent it on terms set by others. The first move is small and specific, one problem worth owning, built once and kept. What that problem is, and what owning versus renting costs at your scale, are specific to you, and where any real conversation starts.

> **Adopt AI, yes. The only question left is whether you own it.**

Building used to be slow. So advice was the product.

That constraint is gone. What stays scarce is judgement about what to build, and ownership of what you build. Rent the frontier. Own what runs.

HOW WE SOURCED THIS

Every figure is drawn from primary or independent sources, government and official statistics, peer-reviewed research, company filings and first-party disclosures, and specialist data providers. Where a number is our own synthesis across more than one source, it's labelled Covelent analysis and its inputs are cited. Figures dated 2026 reflect the most recent data available at the time of writing.

Covelent

COVELENT.COM · +44 (0) 20 8881 7767

15 STRATTON STREET, MAYFAIR, LONDON W1J 8LQ