

The Hardware Moat: Why Owning GPUs Beat Renting Them Through a Macro Shock

A multi-year subsurface AI programme makes a compute-sourcing decision on day one that looks like a finance question and turns out to be a risk question. Rent the GPUs and the programme's largest recurring cost is pinned to a market you do not control; own them and that cost is fixed the day you sign. We made the owning choice in 2020, before the 2022 energy shock, through a multi-year hardware partnership that put the cards on our own books and a local data-center deal that gave us full cost visibility.

When the shock arrived, the consequences were measurable and asymmetric: the DGX and HPC hosting market climbed from USD 12-15K per month pre-war to past USD 20K post-war, energy prices rose about 400 percent, and electricity bills rose 394 percent, while the programme carried a single bounded, transparent contingency of OMR 20K, roughly USD 51K, across a twelve-month window. This paper reconstructs the decision as a board would: it reads the three-way cost model behind the choice (public cloud near EUR 17,200 a month, private cloud near EUR 4,000, on-premise near EUR 20,000, with a mid-size AI server at 50-120K and two servers at 100-200K amortised), it shows why the public-cloud and on-premise sticker prices land alike while their risk profiles diverge, and it frames the contingency as the single approvable number that ownership put on the agenda instead of an open-ended rental bill. We argue that owned compute is best evaluated as a hedge with a known premium, not as a line of capital spending, and we end with the concrete questions a CIO should ask before committing to a multi-year AI programme's infrastructure.

Tarry Singh

Founder & CEO

Narendra Patwardhan

Research Collaborator

COMPUTE-ECONOMICS / GPU-OWNERSHIP / OWNED-VS-RENTED / TOTAL-COST-OF-
OWNERSHIP / ON-PREMISE-AI / PRIVATE-CLOUD

CONTENTS

01	The decision was a hedge, not a purchase
02	What the shock actually did to the two paths
03	The cost model behind the choice
04	What the utilisation ledger says about the crossover
05	The board only ever saw one number
06	Where owning stops being the right answer
07	The questions to ask before a multi-year commitment
08	Limitations
09	References

Most infrastructure decisions on a multi-year AI programme are reversible. You can move a pipeline, swap a framework, renegotiate a storage tier. The compute-sourcing decision is not one of them, and it is usually made first, on day one, when the programme's risk profile is least understood and the pressure to minimise upfront spend is highest. Rent the GPUs and you defer the capital, keep the option to walk away, and pin your largest recurring cost to a market you do not control. Own them and you commit the capital, lose the walk-away option, and fix that same cost the day you sign. Framed as a finance question, renting almost always wins the first meeting.

We made the owning choice in 2020 for a multi-year subsurface AI programme with a mid-sized Middle East carbonate operator. The programme ran borehole-image interpretation at scale: detection models for fractures and bedding planes, a computer-vision track for vugs, and the compute-heavy training and retraining that a research-grade programme consumes over years. The decision to own the compute, through a multi-year hardware partnership that put the cards on our own books and a local data-center deal that gave us full cost visibility, was not made because we forecast a war. Nobody did. It was made because the alternative left the programme's largest cost exposed to something we could not bound, and bounding cost was worth a premium.

Two years later the premium paid for itself in a way that was easy to measure. The 2022 energy shock moved the DGX and high-performance-compute hosting market from USD 12,000-15,000 per month before the war to past USD 20,000 after it, compounded by energy prices rising about 400 percent and electricity bills rising 394 percent. A programme renting its compute would have watched its single largest line item climb without a ceiling for the duration of the shock. The programme we ran carried a different number entirely: a single bounded, transparent contingency of OMR 20K, roughly USD 51K, over a twelve-month window, and that was the whole of the shock's effect on our infrastructure cost. This paper is the reconstruction of that decision as a board would examine it, and the argument that owned compute is best understood as risk management with a known premium rather than as a line of capital spending.

The decision was a hedge, not a purchase

The instinct when a team proposes buying hardware is to treat it as a capital-allocation question: what is the asset, what is the payback period, what is the residual value. That framing is not wrong, but it misses what the decision actually does. Buying the compute for

a multi-year programme is functionally taking out an insurance policy against the price of rented compute. The premium is the difference between the amortised cost of the owned asset and the rental rate you would otherwise pay in calm conditions. The payout is the divergence between the two when conditions stop being calm.

In 2020, before the shock, that premium looked unremarkable. Owned hardware amortised to roughly the same monthly figure as renting the equivalent capacity; on paper the two paths were close, and a purely cost-minimising analysis would have leaned toward renting to keep the capital free and the commitment reversible. What the cost-minimising analysis could not price was the variance. Rented compute has a cost with a wide distribution: cheap when the market is slack, punishing when it is tight, and pinned to energy and supply conditions that a subsurface AI programme has no influence over. Owned compute has a cost with almost no variance once the capital is committed. The choice between them is a choice about how much variance you are willing to carry on your largest recurring line item for the life of a multi-year programme.

We chose to carry as little as we could. The hardware partnership locked in 2020 meant the cards were bought and on our books before the shock, so the rental market's move simply did not reach the programme's cost base. The local data-center deal meant we had full visibility into and control over what the owned fleet cost to run, rather than an opaque hosting bill that bundled hardware, power, and margin into a single number that moved with the market. Those two moves together are the moat: not a clever piece of engineering, but a structural decision that took the programme's biggest cost off the table before the table caught fire.

It is worth being precise about why a rented compute bill moves at all, because the transmission mechanism is what makes the variance real rather than theoretical. A DGX or HPC hosting rate is not just the amortised price of the hardware; it is that price plus power, plus data-center overhead, plus margin, and power is a large and volatile share of it. When energy prices rose about 400 percent and electricity bills rose 394 percent through the 2022 shock, those increases fed almost directly into hosting rates, which is exactly why the market moved from a USD 12-15K per month band to past USD 20K in the span of the crisis. A renting programme is therefore not merely exposed to the GPU market; it is exposed to the energy market through the GPU market, with no way to separate the two. An owning programme with its own data-center arrangement still pays for power, but it pays the metered cost directly and transparently rather than a marked-up, bundled, market-set rate,

and the capital cost of the cards themselves is already fixed. The bounded contingency was sized to cover that direct power increase on the owned fleet, which is why it could be a single capped number rather than an open-ended one.

What the shock actually did to the two paths

The abstract argument becomes concrete when you integrate the two cost paths over the twelve-month contingency window. The rented path is the DGX and HPC hosting market, which moved from a pre-war band of USD 12,000-15,000 per month to a post-war floor past USD 20,000, driven up month after month by the energy and electricity rises that fed straight into hosting rates. The owned path is the bounded contingency: OMR 20K, about USD 51K, spread across the same twelve months, and nothing more, because the hardware itself was already paid for and did not re-price when the market did.

Owned compute is risk management: it caps the shock at a fixed contingency

The 2020 hardware partnership meant the rent line was never on the books.

CUMULATIVE COST OVER 12 MONTHS · THE WEDGE IS THE RISK REMOVED

WHAT EACH PATH CARRIES PER MONTH

RENTED DGX / HPC HOSTING

20.0K USD / month, market-tracking
12-15K pre-war climbing past 20K+ post-war

OWNED HARDWARE + CONTINGENCY

4.2K USD / month, fixed ceiling
USD 51K capped across 12 months

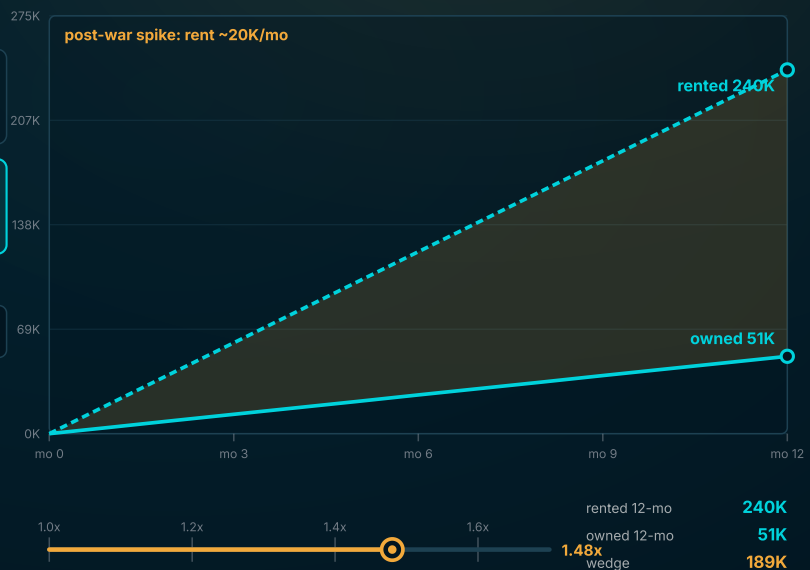
WHAT DROVE THE RENT LINE

+400%
energy prices**+394%**
electricity bills

Owned hardware bought pre-shock does not re-price when the rental market and the grid do. That is the moat, stated plainly.

MARKET-SHOCK LEVERdrag the rental market from pre-war (1.0x)
into the post-war spike; owned stays flat

sourced: 12-15K to 20K+/mo rent, energy +400% / elec +394%, USD 51K cap · monthly ramp illustrative



ES-1920 · OWNED VS RENTED RISK WEDGE · A BOUNDED CONTINGENCY VS A RUNAWAY RENT

Two cumulative cost paths over the twelve-month contingency window, integrated from the sourced monthly rates. The rented DGX/HPC path tracks the hosting market as it climbs from USD 12-15K per month pre-war toward the USD 20K+ post-war floor, compounded by energy up about 400 percent and electricity up 394 percent. The owned path carries only the bounded USD 51K contingency spread across the twelve months, because a 2020 hardware partnership meant the cards were already bought and the rental market never touched the programme's cost base. The orange wedge between the two lines is the single arguing element: the open-ended market exposure the ownership decision converted into a capped line item. Drag the market-shock lever from 1.0x pre-war into the 1.7x post-war spike to watch the rented path run away while the owned path holds flat. The rental band, the energy and electricity rises, the USD 51K contingency, and the 2020 partnership are sourced from the engagement's war-time contingency and cost-comparison memos; the month-by-month ramp shape is an illustrative straight-line interpolation between the sourced endpoints, not a reported monthly ledger, and this is a risk-exposure comparison, not an audited invoice reconciliation.

The wedge between the two lines is the entire argument. It is not a modelled scenario or a projection; it is the difference between a cost that tracked a market through a shock and a cost that was fixed before the shock began. Drag the market-shock lever from its pre-war baseline into the post-war spike and the rented path runs away while the owned path holds flat, because the only thing moving is the rental market and the owned path is not connected to it. The programme converted an open-ended, market-tracking exposure into a capped line item, and the size of the wedge is the exposure that conversion removed.

Two features of this comparison deserve honesty. First, the month-by-month ramp is an illustrative straight-line integration between the sourced pre-war and post-war rates; we did not keep a monthly rental ledger, because we were not renting. What is sourced is the band the rental market moved across and the bounded contingency the owned path carried. Second, the comparison is a risk-exposure comparison, not an audited invoice reconciliation. The point is not that we saved a precise dollar figure; it is that one path had a ceiling and the other did not, and the shock is exactly the condition under which that distinction stops being academic.

The cost model behind the choice

The ownership decision rested on a three-way total-cost-of-ownership comparison built at proposal time, before any shock was in view. The model priced three ways to run the programme's infrastructure on a monthly basis: public cloud at about EUR 17,200, private cloud at about EUR 4,000, and on-premise at about EUR 20,000. Read as sticker prices, the ranking tells a simple story: private cloud is cheapest by a wide margin, and public cloud and on-premise are close to each other at the top. That reading is where most infrastructure decisions go wrong, because the sticker price is not what the decision turns on.

THREE-WAY MONTHLY TCO · PUBLIC VS PRIVATE VS ON-PREMISE

2.8K

EUR/mo between the two look-alike bills

The sticker price is not the decision; what the bar is made of is

On-premise and public cloud bill alike, yet one buys an asset and one buys nothing.

MONTHLY BILL, STACKED BY COMPONENT (EUR)

HIGHLIGHT A COMPONENT ACROSS ALL THREE

Compute / hardware slab

DevOps FTE (1.5)

Platform + repos

WHAT THE ON-PREMISE SLAB ACTUALLY BUYS

50-120K EUR per mid-size AI server

100-200K EUR for two, then amortised

a kept asset; public-cloud hosting at the same monthly figure keeps nothing

Public full-capacity hosting runs 60-80K EUR per server per year.

FOCUS: COMPUTE / HARDWARE SLAB

public cloud

6.3K

private cloud

3.0K

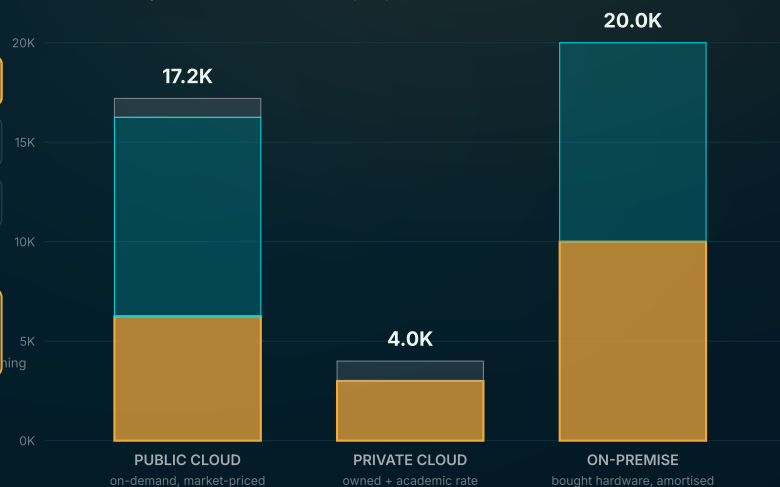
on-premise

10.0K

Same slab size, opposite ownership: on-prem keeps the asset, public cloud rents it away.

sourced: 17,200 / 4,000 / 20,000 EUR/mo totals, 10K DevOps FTE, 50-120K server capex, 950 platform line items · internal slab splits partly illustrative

ES-8559 · THREE WAY TCO STACK · SAME BILL, OPPOSITE OWNERSHIP



The infrastructure cost model read as three stacked monthly bills instead of three sticker prices. Public cloud (about EUR 17,200 per month) and on-premise (about EUR 20,000 per month) land close together, while private cloud on already-owned, academically discounted hardware sits far below at about EUR 4,000. The ranking is not the argument. The orange compute-and-hardware slab is: on-premise carries a mid-size AI server at 50-120K each, two servers at 100-200K amortised to roughly 10K per month, which is a bought asset the operator keeps; public cloud carries an equally large on-demand hosting slab that buys nothing and re-prices with the market. Same monthly figure, opposite risk profile. Toggle a component to highlight it across all three bars. The three monthly totals, the DevOps FTE, the server capex range, the platform and repository line items, and the 60-80K per server per year full-capacity figure are sourced from the engagement's cost-comparison memo; where a bar's total is sourced but its exact internal split is not fully itemised, the residual is grouped into the compute slab and flagged, so this is a composition comparison of the model, not an audited invoice.

What the decision turns on is what each monthly bill is made of. The public-cloud and on-premise figures land within a few thousand euros of each other, but the compute slab inside each is a completely different kind of money. On-premise carries a mid-size AI server at 50-120K each, two servers at 100-200K, amortised to roughly EUR 10,000 a month; that slab is a bought asset the operator keeps, and it does not re-price when the rental market moves. Public cloud carries an on-demand hosting slab of similar monthly size that buys nothing, keeps nothing, and re-prices with the market at every renewal. Same monthly figure, opposite ownership, opposite risk. The private-cloud option is cheap for the same

structural reason the on-premise option is safe: it runs on already-owned, academically discounted hardware rather than on-demand capacity, which is why its hosting line is a fraction of the public-cloud one.

The DevOps staffing is the one line item where public cloud and on-premise are nearly identical, roughly EUR 10,000 a month for one-and-a-half cloud-and-DevOps engineers off an 80K salary base. That symmetry is worth naming because it is where the cloud-versus-owned debate often gets muddled: the human cost of running the infrastructure is broadly the same either way, so it is not the axis the decision should turn on. The platform and repository line items (source control, storage, operating-system and machine-learning platform tooling, around EUR 950 a month combined) are a rounding error against the compute slab. Strip the model down and the only component that actually distinguishes the options by risk is the compute slab, and the only question that matters about that slab is whether you own it or rent it. For the deeper case on why an operator's models and their infrastructure tend to stay inside the perimeter, our companion piece [Why Energy Companies Keep Their Models On-Premises](#) walks the security and control mechanics; here the argument is narrower and financial.

There is one more number in the model worth surfacing for a board: public full-capacity hosting was quoted at 60,000-80,000 euros per server per year. That is the calm-market figure. It is already close to the amortised cost of owning the equivalent server outright, and it carries none of the ownership. The moment the market tightens, that figure moves and the owned figure does not. A board looking at the calm-market numbers alone would see a near-tie; a board that asks what happens to each number under stress sees the whole point.

What the utilisation ledger says about the crossover

Owning only beats renting if the owned fleet is actually used, and the strongest evidence for the decision comes not from the shock but from the programme's own compute-hour ledger. The ledger tracked budgeted against actual GPU and MLOps runtime hours phase by phase, and the actuals tell a story about why fixed-cost owned capacity fit this workload. The first phase ran on budget at roughly 1,200 hours. The second phase, where a deliberate decision to run supervised and unsupervised model tracks in parallel doubled the

compute paths, came in at about 2,600 hours against a 1,200-hour budget, a 2.2x overrun. The third phase ran to roughly 7,500 hours against a 5,800-hour plan. Across the programme the runtime landed near 11,300 hours against 8,200 budgeted.

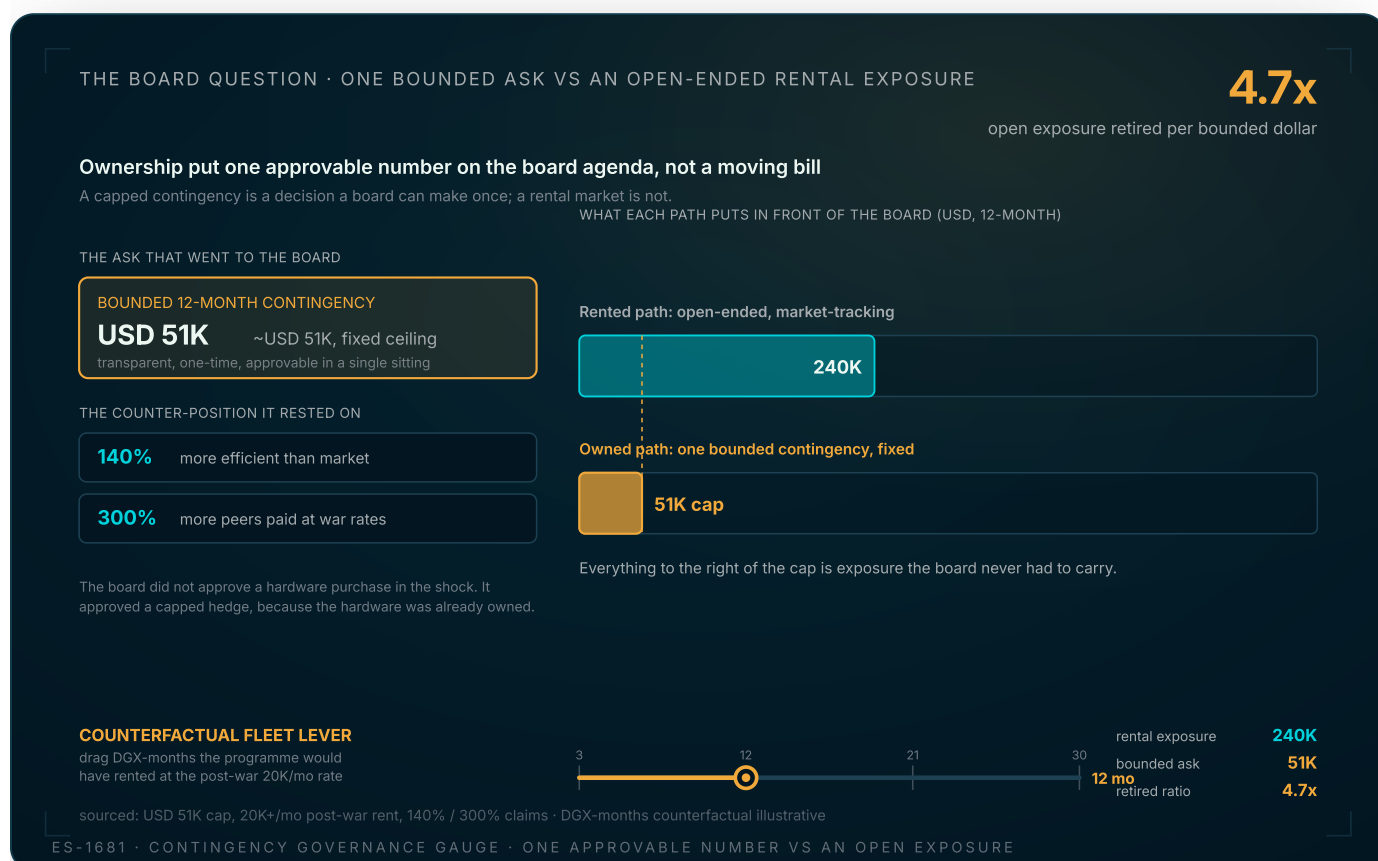
Those overruns are the exact condition under which owned compute wins and rented compute hurts. A rented programme that runs 38 percent more hours than budgeted pays 38 percent more, linearly, on top of whatever the market is doing to the per-hour rate; an owned programme with the capacity already bought absorbs the extra hours at close to zero marginal cost, because the cards are sitting there either way. The parallel-track decision that drove the second-phase overrun was a research choice made for model-quality reasons, and on rented compute it would have carried a direct and compounding cost penalty. On owned compute it was free to make. That is a subtler form of the same moat: ownership does not just cap the price of the hours you planned, it caps the price of the hours you did not, and a research-grade programme almost always runs more hours than it planned.

The utilisation profile is also what made this specific programme a candidate for owning in the first place. A single training run took six to eighteen hours, and the programme projected running dozens of these concurrently as it scaled into the later phases, with sustained GPU utilisation in the 80-90 percent band. That is a fleet that stays warm. An owned server at 85 percent utilisation is a good asset; the same server at 15 percent utilisation would have been a bad one, and a bursty workload with long idle stretches is precisely the case where renting and releasing capacity beats owning it. The decision was not owning-is-always-right; it was owning-is-right-for-a-long-heavy-high-utilisation-programme, which this was.

The board only ever saw one number

From the operator's chair, the ownership decision was invisible in exactly the way a good hedge should be. When the shock arrived, a programme renting its compute would have brought an open-ended, market-tracking exposure to the board every quarter, asking for more as the rental market climbed and unable to say where it would stop. That is an uncomfortable thing to put on an approval agenda, because it is not a decision the board can make once; it is a standing liability the board has to keep re-approving as the market moves.

The programme we ran brought a single approvable number to the board once: the bounded OMR 20K, about USD 51K, twelve-month contingency. It was transparent, it had a ceiling, and it could be approved in one sitting because the hardware it protected was already owned. The board was not asked to approve a hardware purchase in the middle of a crisis; it was asked to approve a capped hedge, which is a far easier thing to reason about and a far easier thing to govern.



The ownership decision read from the board's chair. A rented programme through the 2022 shock puts an open-ended, market-tracking exposure on the approval agenda every quarter; the owned programme put a single approvable number in front of the board once, a bounded USD 51K contingency for twelve months, because the hardware was already bought under a 2020 partnership. The top bar is the counterfactual rental exposure the programme would have carried had it rented instead; the bottom bar and its dashed marker are the fixed contingency cap. The orange element is the only one that argues: the bounded ask itself, held fixed while the rental exposure runs. Drag the fleet lever to sweep how many DGX-months of rental the bounded line item stood in for; the headline ratio is how much open-ended exposure one capped, transparent number retired. The contingency, the post-war rental rate, and the 140 percent efficiency and 300 percent peer over-billing claims are sourced from the engagement's war-time contingency memo; the DGX-months fleet sweep is an illustrative counterfactual, so this is a governance framing of the exposure, not a booked rental invoice.

The gauge reads that bounded ask against the open-ended rental exposure it stood in for. Drag the fleet lever and the counterfactual rental exposure grows without limit while the bounded contingency holds fixed; the ratio between them is how much open-ended, unbounded exposure one capped and transparent line item retired. The contingency memo made the counter-position explicit: through the shock the owned programme claimed to remain about 140 percent more cost-efficient than the market and estimated that clients paying war-time rental rates were paying roughly 300 percent more. Those are the programme's own claims from the period, not independently audited figures, and we present them as what they are: the argument the memo made to the board at the time. The DGX-months fleet sweep in the gauge is likewise an illustrative counterfactual over the sourced post-war rate, because the programme did not rent and so has no rental invoice to reconcile against. What is sourced is the bounded contingency, the rental band, and the efficiency claims.

The governance point stands regardless of the exact ratio. A bounded contingency is a decision a board can make; a rental market is not. Ownership did not just save money in expectation, it changed the shape of the decision the board had to make, from an open-ended standing liability into a single capped line item. That change of shape is worth as much as the dollars, and on a multi-year programme it may be worth more.

Where owning stops being the right answer

An honest paper about a hedge has to say when the hedge is a bad trade, because owned compute is not the right answer for every programme, and treating it as a universal recommendation would be exactly the kind of overreach a CIO should distrust. The ownership decision is favourable under a specific and identifiable set of conditions, and unfavourable outside them.

Owning wins when the programme is long. Amortising 100-200K of hardware over two servers only beats renting if the programme runs long enough to consume that capacity; a six-month proof of concept should rent, full stop, because it will never reach the crossover point where ownership's fixed cost undercuts rental's recurring one. Owning wins when utilisation is high and sustained. An owned server sitting idle is pure loss, whereas rented capacity can be released; a programme with steady, heavy training demand is the one that fills an owned fleet, and a bursty, occasional workload is the one that should rent and let someone else carry the idle time. Owning wins when the cost variance actually matters to

you. If your programme's compute cost is a small fraction of its budget, the variance is noise and the flexibility of renting is worth more than the certainty of owning; the calculus flips only when compute is a large enough line item that a market move can threaten the programme's viability, which is precisely the situation a compute-heavy subsurface AI programme is in.

And owning carries its own risks that renting does not. Owned hardware ages, and a fleet bought in 2020 is a generation behind by 2023; the hedge against price volatility is bought at the cost of a hedge against obsolescence, which renting provides for free. This is a genuine cost, not a rhetorical concession. The programme's owned fleet was a specific vintage of accelerators, and by the later phases newer cards with more memory and faster interconnect were available on the rental market that the owned fleet could not match. A renting programme could have moved onto that newer capacity at the next renewal; the owning programme was committed to the cards it had bought. The mitigation the programme used was to make the owned hardware go further through engineering rather than replacement, containerising workloads across partitioned instances so a fixed fleet could serve many more concurrent runs, but that is a workaround for the obsolescence exposure, not an elimination of it. A board weighing this trade has to accept that owning locks in the hardware generation as firmly as it locks in the price, and decide whether the programme's horizon is short enough that the generation still matters or long enough that the price certainty dominates. Owned hardware has to be run, and the DevOps cost of doing so is real even though, as the model showed, it is broadly the same as running rented infrastructure. Owned hardware ties up capital that could have gone elsewhere. None of these cancels the case for owning on a long, compute-heavy, multi-year programme, but all of them are reasons the decision has to be made deliberately rather than by default, and reasons a CIO should demand the specifics before signing.

The questions to ask before a multi-year commitment

The value of this reconstruction is not the specific numbers; it is the decision framework the numbers illustrate. A CIO facing a multi-year AI infrastructure commitment should be able to answer, before signing, a short set of questions that turn the compute-sourcing decision from a finance default into a risk decision made on purpose.

- How long will the programme actually run, and does it run past the point where owned hardware amortises below the rental rate? If the honest answer is under a year, rent.

- What fraction of the programme's total budget is compute, and how much would a doubling of that cost hurt? If compute is a large line item, its variance is a risk worth hedging; if it is small, it is not.
- What is the utilisation profile? Steady heavy demand fills an owned fleet and justifies it; bursty demand wastes it and argues for renting.
- If you rent, what is your exposure if the rental market moves against you mid-programme, and can you bound it? A programme that cannot answer this is carrying an unpriced risk on its largest line item.
- If you own, how are you hedging obsolescence, and what is the DevOps and capital cost of running the fleet yourself? Owning trades one risk for another, and both have to be named.
- Can the infrastructure decision be brought to the board as a single bounded number, or does it arrive as an open-ended standing liability? The shape of the decision matters as much as its size.

Answer those and the compute-sourcing decision stops being a reflexive lean toward the cheapest first-meeting option and becomes what it always was: a decision about how much variance the programme can carry on its largest cost for years, and whether a known premium today is worth a bounded outcome tomorrow. We made that trade in 2020 for reasons that had nothing to do with the war that validated it, and the validation came in the form of a single capped contingency where a rented programme would have carried a runaway bill. The moat was never the hardware. It was the decision to take the programme's biggest cost off the table before anyone knew the table would catch fire.

WHAT THIS WHITEPAPER ARGUES

- 1** A multi-year AI programme's compute-sourcing choice is a risk decision, not a finance default: renting pins your largest recurring cost to a market you do not control, while owning fixes that cost the day you sign, so the real question is how much cost variance the programme can carry for years.
- 2** Owned compute is best evaluated as a hedge with a known premium, not as capital spending: in calm 2020 conditions owning and renting amortised to similar monthly figures, but the owned path had almost no variance and the rented path had a wide one.
- 3** The 2022 shock made the premium pay off measurably: DGX and HPC hosting moved from USD 12-15K per month pre-war to past USD 20K post-war with energy up about 400 percent and electricity up 394 percent, while a 2020 hardware partnership and a local data-center deal held the programme to a single bounded OMR 20K (about USD 51K) twelve-month contingency.
- 4** The three-way cost model shows why sticker price misleads: public cloud (~EUR 17,200/mo) and on-premise (~EUR 20,000/mo) bill alike, but on-premise's compute slab (50-120K per server, 100-200K for two, ~10K/mo amortised) is a bought, owned asset while public cloud's equal-sized slab buys nothing and re-prices with the market.
- 5** Ownership changed the shape of the board decision, not just its size: it put one approvable, capped, transparent contingency on the agenda instead of an open-ended standing liability that would have had to be re-approved as the rental market climbed.
- 6** Owning is not universal: it wins on long, compute-heavy, high-utilisation programmes where compute is a large budget line, and loses on short, bursty, or compute-light ones; it also trades price-volatility risk for obsolescence and capital risk, so the decision must be made deliberately against a specific set of CIO questions.

Limitations

This is a board-level synthesis of a real decision, and it should be read with its boundaries in view. The month-by-month cost paths in the risk-wedge exhibit are a straight-line integration between the sourced pre-war and post-war rental rates, not a reconstructed monthly ledger; the programme owned its compute and therefore has no rental invoice trail to reconcile against. The DGX-months counterfactual in the governance exhibit is likewise an illustrative sweep over the sourced post-war rate, used to make the exposure legible, not a booked cost. The 140 percent market-efficiency and 300 percent peer over-billing figures are the programme's own claims from its 2022 contingency memo, presented as the argument made to the board at the time rather than as independently audited results. The three-way total-cost-of-ownership figures (EUR 17,200 / 4,000 / 20,000 per month) and the server capex ranges are sourced from the engagement's proposal-stage cost model, and where a bar's internal split is not fully itemised in the source, the residual is grouped and flagged in the exhibit. Finally, the specific crossover conditions under which owning beats renting depend on programme length, utilisation, and the compute share of budget, all of which vary by engagement; the framework generalises, the exact numbers do not.

References

Contingency-Energy-Electricity memo, 2022. The engagement's war-time cost-contingency record: the DGX and HPC rental band moving from USD 12-15K to past USD 20K per month, energy up about 400 percent, electricity bills up 394 percent, the bounded OMR 20K (~USD 51K) twelve-month contingency, the 2020 hardware partnership, the local data-center deal, and the 140 percent efficiency / 300 percent peer over-billing claims.

Infrastructure cost-comparison model, proposal stage. The three-way monthly total-cost-of-ownership comparison: public cloud ~EUR 17,200, private cloud ~EUR 4,000, on-premise ~EUR 20,000, with the mid-size AI server at 50-120K each, two servers at 100-200K amortised, the 1.5 DevOps FTE at ~EUR 10,000 a month, the platform and repository line items, and the 60-80K per server per year public full-capacity figure.

MLOps infrastructure cost ledger. The programme's phase-by-phase compute-hour and cost record that grounds the utilisation and amortisation reasoning behind the owning-versus-renting crossover conditions.

The Hardware Moat: Why Owning GPUs Beat Renting Them Through a Macro Shock

Published October 2025. Generated 2026-07-22.

<https://www.earthscan.io/whitepapers/hardware-moat-owning-gpus-beat-renting-macro-shock>

Copyright EarthScan. All rights reserved.