

SUPPLY CHAIN RESEARCH · EDITORIAL FEATURE

The Vendor Lock-In Playbook

How Switching Costs Are Engineered, and How Buyers Take Their Leverage Back

Lock-in is not an accident of enterprise software. Much of it is a deliberate product decision, and the buyer signs it away in the first contract without noticing. Regulators are now dismantling one part of it. This is what remains, and what to negotiate before you sign.

Independent, vendor-neutral research for supply chain and technology leaders

Supply Chain Research | supplychainresearch.com | 2026

Contents

A decision framework for the AI era

01	Executive summary	01
02	What lock-in is, and what it is not	02
03	What a locked-in position is worth	03
04	The six mechanisms of switching cost.....	04
05	Data gravity and the egress asymmetry.....	05
06	The contract mechanics nobody reads	06
07	The uplift clause that compounds.....	07
08	What it actually costs to leave	08
09	The regulators arrived.....	09
10	Why the regulation solves less than it appears to	10
11	Leverage is spent at signature.....	11
12	The terms to negotiate, before you sign.....	12
13	The one-page exit test.....	13
14	Negotiation, timing, and a scoring rubric.....	14
15	Conclusion: optionality is a purchased asset	15
16	Methodology, caveats, and sources.....	16

01 Executive summary

A buyer signing an enterprise software contract is negotiating two things at once, and is usually aware of only one of them. The first is the price, which is visible, comparable, and the subject of a great deal of attention. The second is the cost of leaving, which is invisible, is almost never quantified, and is overwhelmingly determined by terms in the contract that the buyer will not read carefully because they do not appear to be about money. The second negotiation matters more than the first, because the price of the first year is a number the buyer chose and the price of every subsequent year is a number the vendor will choose, and how much freedom the vendor has to choose it depends entirely on how expensive the buyer has made it to walk away.

This article is about how those switching costs come to exist. Some are the natural consequence of using a system for years: data accumulates, people learn it, processes form around it, and none of that is anybody's fault. But a substantial share is engineered, in the sense that a product manager made a decision that had the effect of making departure more expensive, and that decision was taken deliberately and is defended in board papers as improving retention. We say honestly that this is not a scandal and is not illegal; it is a rational commercial strategy that a buyer should expect from any competent vendor. What is a scandal is how little most buyers do about it, given that almost every protection worth having is available for free at the moment of signature and is unobtainable at any price thereafter.

800-1,500%

reported renewal increases faced by customers of a major vendor after an acquisition changed who held the leverage

Jan 2027

the date on which the European Union abolishes charges for switching cloud providers entirely

~6%

the share of the true cost of leaving that egress fees typically represent, which is why the regulation solves less than it appears to

The argument in brief

- **Lock-in is a product decision, not an accident.** Proprietary formats, egress pricing, and contract mechanics are chosen, and they are chosen because they work.
- **The asymmetry is the business model.** Getting your data in is free. Getting it out is not. That is not an oversight; it lowers the barrier to entry and raises the barrier to exit.
- **Regulation is dismantling the smallest part.** The European Union abolishes switching fees from January 2027, and the hyperscalers moved early. Egress is roughly six percent of the real cost of leaving.
- **The real cost of exit is the integrations and the data.** Rebuilding interfaces and reconciling records dominates the exit bill, and no regulation touches either.
- **Leverage exists only before signature.** Every term worth having costs nothing in the request for proposals and is close to unobtainable at renewal, when the vendor knows exactly what leaving would cost you.

02 What lock-in is, and what it is not

The term is used loosely and usually pejoratively, and the looseness costs buyers money, because it prevents them from distinguishing the switching costs they should accept from the ones they should refuse. A precise definition is therefore worth establishing before anything else: lock-in is the total cost, in money, time, disruption, and risk, of moving from the current vendor to an alternative. It is a number. It is measurable, at least approximately. And it is the number that determines how much pricing power the vendor has over the buyer for as long as the relationship lasts.

Understood that way, the first thing to say is that not all lock-in is illegitimate, and a buyer who treats it as uniformly sinister will make bad decisions. Consider an organization that has used a warehouse management system for eight years. Its people are expert in it. Its processes have been shaped around it. Its historical data lives in it, its integrations connect to it, and its exception handling has been tuned by years of accumulated learning. Moving to a different system would be expensive, and the expense is not the vendor's doing. It is the natural residue of having derived value from a system for a long time, and the same cost would exist if the vendor had done nothing whatsoever to create it. That is not lock-in in any blameworthy sense. It is what it feels like to have used something successfully.

But a substantial share of switching cost is not like that at all. Consider an export function that produces a file in a format no competing system can read. Consider a data egress charge levied per gigabyte on the buyer's own data. Consider a contract that renews automatically unless notice is given in a sixty-day window that closes before the buyer has begun evaluating alternatives. Consider a proprietary query language, or a licensing metric so complex that the buyer cannot determine what a competitor's equivalent would cost, or an application programming interface that is quietly degraded when a customer is discovered to be building connections to a rival. None of these is a by-product of value delivered. Each is a decision, taken by somebody, with the specific intent of raising the cost of departure, and each is negotiable at the moment before it is agreed to.

The distinction between these two categories is the most useful analytical move a buyer can make in this domain, and it is the organizing idea of this article. The switching costs that arise from use are a fact to be managed and planned for. The switching costs that were engineered are a term to be negotiated. Confusing the two produces either a buyer who accepts everything, on the reasoning that some lock-in is inevitable, or a buyer who fights everything, on the reasoning that all of it is predatory. Both are wrong, and both leave money and freedom on the table.

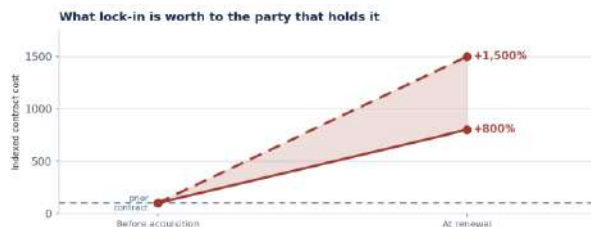
A useful way to make the concept operational is to observe that lock-in behaves exactly like a put option that the buyer has sold to the vendor without receiving a premium. The vendor holds the right, at each renewal, to reprice the relationship, and the buyer's ability to refuse is bounded by the cost of leaving. The higher that cost, the more valuable the option, and the option was granted for nothing, in a contract negotiation in which the buyer was concentrating on the year-one price. A finance function that would never permit such an instrument to be written on a commodity exposure permits it routinely on the software that runs the company, because the instrument is not called an option and does not appear on any schedule.

This framing also explains why lock-in is invisible on the balance sheet and therefore invisible to governance. There is no line item for it, no register that records it, and no executive whose objectives include reducing it. It accumulates silently, as a by-product of decisions taken for other reasons, and it becomes visible only at the moment when it is exercised against the organization, at which point the response is indignation rather than preparation. An organization that maintained a simple register of what it would cost to leave each of its major vendors would be doing something unremarkable in every other category of concentration risk and something almost unheard of in software.

It is also worth distinguishing lock-in from concentration, since the two are related and are not the same. Concentration is the amount of the organization's operation that depends on a single supplier. Lock-in is the difficulty of replacing that supplier. An organization can have high concentration and low lock-in, if the supplier is critical but readily substitutable, and it can have low concentration and high lock-in, if a small system is impossible to replace because it holds data nothing else can read. The risk that matters is the product of the two, and a register that captures both, for each significant vendor, is the foundation of any serious governance of this exposure. Most organizations have neither number for any of their suppliers.

03 What a locked-in position is worth

Abstract arguments about switching costs become concrete the moment a buyer sees what happens when a vendor decides to convert a locked-in position into cash. The most instructive recent demonstration, shown in Figure 1, followed the acquisition of a major virtualization vendor by a company with an explicit strategy of maximizing revenue from an installed base.



Reported largest percentage increase experienced by VMware customers following the acquisition of a major virtualization vendor, with customers not requiring the migration to bundled subscriptions and multi-year terms. Required by customer groups and/or analysts, the vendor requires disclosure. The point is a curve, please multiply by the number of years of a locked-in position, and the curve is not in dispute.

Figure 1. What a locked-in position is worth to whoever comes to own it. Reported renewal increases following an acquisition; the vendor disputes characterizations, and the direction is not in dispute.

Customers across Europe reported renewal increases in a range that press accounts and customer groups have put at several hundred to well over a thousand percent, alongside the retirement of perpetual licenses, forced migration into bundled subscriptions, and the imposition of multi-year minimum terms. The vendor has disputed some characterizations, and a fair account should note that the reported figures come from affected customers rather than from audited disclosure. But the essential fact is not seriously contested, and it is the fact that matters: an installed base that could not readily leave was repriced, and the repricing was possible precisely because leaving was expensive.

It is important to be clear about what this case does and does not demonstrate, because the wrong lesson is easy to draw. It does not demonstrate that the acquiring company behaved improperly; a company that acquires an asset is entitled to price it, and its shareholders would have grounds for complaint if it did not. What it demonstrates is that the buyer's protection was never the vendor's goodwill. The customers affected had, in many cases, been well treated for years, had good relationships with their account teams, and had no reason to expect what happened. Then the ownership changed, the strategy changed, and the only thing standing between those customers and an arbitrary price was the cost of leaving, which was high, which is why the price could be arbitrary.

The generalizable lesson is therefore uncomfortable and worth stating plainly. A buyer's leverage over a vendor is not a function of the relationship, the account team, the years of partnership, or the assurances given during the sales process. It is a function of one number: what it would cost to leave. That number can be measured, it can be reduced, and the reduction is almost entirely accomplished by decisions taken before the first contract is signed. Everything that follows in this article is an elaboration of that single point.

Two further points about the case are worth drawing out, because both generalize far beyond one vendor and one acquisition. The first is that the vulnerability was created years before it was exploited. The customers who found themselves facing extraordinary renewal increases had been building on that platform for a decade or more, deepening their dependence with every workload they migrated onto it,

and at no point in that decade did anybody calculate what it would cost to reverse. The exposure was accumulated gradually, by people acting sensibly on each individual decision, and it was realized suddenly, by somebody else's strategic choice. That is the characteristic shape of this risk and it is why it cannot be managed reactively.

The second is that the mechanism of exploitation was not a price increase in isolation. It was a bundle: perpetual licenses were retired, products were repackaged into suites the customer had not asked for, minimum terms were extended, and the increase arrived inside a restructuring that made like-for-like comparison difficult. This is a common pattern and it deserves anticipation in a contract. A buyer who has negotiated a cap on the uplift of the product they bought, and has said nothing about what happens if the product is discontinued and replaced by a differently named bundle, has protected themselves against the least imaginative version of the risk.

The final observation from this case is the one most worth carrying into a boardroom. Every affected customer had, at some point, chosen that platform in a competitive evaluation, and had chosen well: it was a capable product, it was the market leader, and the decision was defensible on every criterion the evaluation contained. The evaluation simply did not contain the criterion that turned out to matter, which was what happens to us if the owner of this product decides to extract value from our dependence. That criterion is absent from almost every enterprise software evaluation conducted anywhere, and it is the one that determines a substantial share of what the organization will actually pay over the following decade.

04 The six mechanisms of switching cost

Switching cost is not one thing. It is an accumulation of six distinct mechanisms, shown in Figure 2, and they differ in an important respect: some are engineered by the vendor, some are structural by-products of using any system, and some are the honest consequence of the vendor having delivered value. A buyer who cannot tell them apart will negotiate the wrong things.

The six mechanisms of switching cost

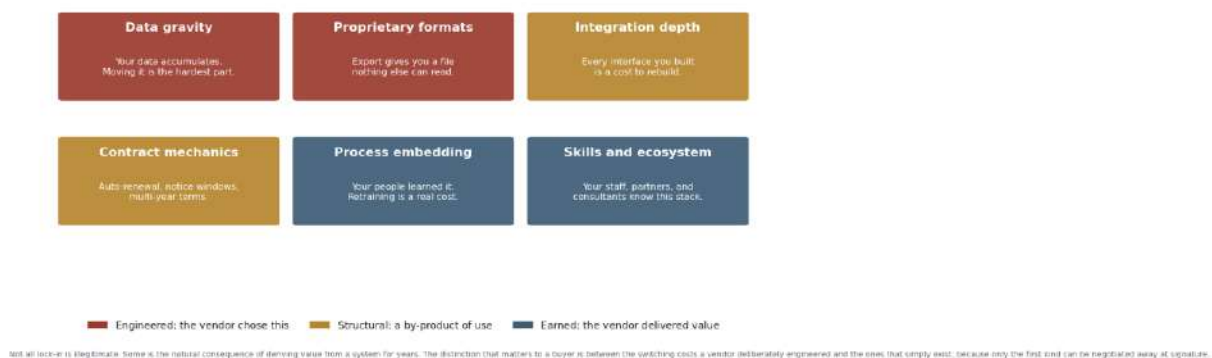


Figure 2. The six mechanisms. Only the engineered ones can be negotiated away, and they are the ones buyers most often ignore.

- **Data gravity.** Data accumulates in the system, and the more there is, the harder and more expensive it becomes to move. This is partly structural and heavily engineered, because the vendor chooses the export formats, the completeness of the export, and whether it charges for the extraction.
- **Proprietary formats.** An export that produces a file no competing system can read is technically an export and practically a hostage note. The right of egress is worthless without a usable format, and the format is entirely the vendor's choice.
- **Integration depth.** Every interface the buyer builds to the system is an asset that becomes a liability at exit, because it must be rebuilt against a new endpoint. This is largely structural, and it is the single largest component of exit cost.
- **Contract mechanics.** Auto-renewal, notice windows, multi-year minimum terms, and termination provisions. Purely engineered, entirely negotiable, and almost never negotiated, because they do not look like they are about money.
- **Process embedding.** The organization's ways of working have formed around the system. Retraining is a real cost and it is not the vendor's fault; it is the price of having used something for a long time.
- **Skills and ecosystem.** The buyer's staff, its consultants, and its partners all know this stack. Moving means retraining or replacing them. This is earned lock-in, and a vendor that has built a deep talent ecosystem has done something valuable.

The taxonomy has an immediate practical use. The last two categories are not worth fighting; they are the cost of having chosen a system and used it well, and a buyer who resents them is resenting the passage of time. The middle two are structural and should be managed architecturally, by decoupling integrations and avoiding unnecessary depth of coupling, rather than negotiated. But the first and fourth, the data mechanisms and the contract mechanics, are pure product and legal decisions, they

were made deliberately, they can be undone at the negotiating table, and they cost the buyer nothing to ask for at the moment when the vendor most wants the deal.

It is worth pausing on the word engineered, because a vendor reading this would object to it and the objection deserves an answer. Few software companies convene a meeting to decide how to trap customers. What happens is subtler and more forgivable, and it produces the same result. A product manager, asked to improve retention, notices that customers who have integrated deeply churn less, and prioritizes features that deepen integration. A pricing team, asked to improve margins, notices that egress revenue is high-margin and does not reduce it. A legal team, asked to reduce revenue volatility, standardizes on auto-renewal. Nobody in this chain has behaved improperly, each decision is defensible in isolation, and the aggregate is a product that is materially harder to leave than it needs to be. Intent is not the point. Effect is the point, and the buyer experiences the effect regardless of the intent behind it.

Supply chain buyers face a version of this problem with a distinctive shape, and it is worth naming because it changes what they should negotiate. A supply chain system does not merely hold the buyer's data; it holds the connective tissue to the buyer's trading partners. Carriers have been onboarded to it. Suppliers submit through its portal. Three-party logistics providers have built to its interfaces. Leaving therefore means not only migrating the organization's own systems but re-onboarding a network of external parties who have no interest in the migration, will not prioritize it, and cannot be compelled. That is a switching cost of a different order from anything on a software licensing schedule, it is almost never modeled, and it means that supply chain systems are among the stickiest software an enterprise buys.

The practical consequence is that the trading-partner dimension belongs in the exit analysis and in the contract. Ask, before signing, what the partner onboarding artifacts actually are, whether the mappings and configurations built for each partner are exportable, and whether the vendor will provide them in a form a successor could use. Most buyers never consider this, discover at exit that the partner configurations are locked inside the platform, and find that the largest single element of their migration is re-doing work that they already paid for once. A clause requiring the export of partner mappings and configurations costs nothing at signature and can be worth a substantial amount of money at exit.

05 Data gravity and the egress asymmetry

Of all the mechanisms, the data one deserves the closest attention, because it is the most consequential and because it contains the clearest evidence of design intent. Figure 3 states the asymmetry in the plainest possible terms.



Figure 3. The asymmetry that defines the cloud business model. Ingress is free. Egress is not. This is not an oversight.

Uploading data to a major cloud platform costs nothing. Retrieving it costs, at published list rates, somewhere in the region of five to nine cents per gigabyte, which puts the retrieval of a hundred terabytes somewhere between five and nine thousand dollars. As an absolute number that is not large, and a buyer who focuses on it has missed the point. The point is the direction. Every element of the pricing model is arranged to make arrival cheap and departure expensive, and no serious observer believes this is an artifact of the underlying cost of bandwidth, since the bandwidth is the same in both directions. It is a commercial decision that has been made consistently, by every major provider, for many years, and its function is to raise the cost of leaving.

The deeper problem, and the one that the fee obscures, is that the right to extract data is worthless without the ability to use what has been extracted. A vendor can comply with every portability obligation ever written and still leave a customer stranded, simply by exporting the data in a structure that only its own system understands. The customer receives a complete, timely, free export consisting of several hundred tables with opaque column names, encoded reference values, no documentation of the relationships between them, and no history of the configuration that gave them meaning. That is an export. It is not portability. And the distinction is precisely where a buyer's contractual attention should be concentrated, because it is the difference between a right that can be exercised and a right that merely exists.

There is a further category of data that buyers almost universally forget to ask about, and its absence is what turns a difficult migration into an impossible one. The transactional records are the easy part. What about the configuration: the workflows, the business rules, the approval hierarchies, the exception logic, tuned over eight years by people who have left? What about the historical audit trail, which may be a regulatory requirement? What about the custom reports, the dashboards, the saved queries, the trained models if the system contains any? Almost no export includes these, almost no contract requires them to be included, and they represent an enormous share of what the organization has actually built inside the system. A buyer who negotiates the right to extract its transactional data and nothing else has secured the right to leave with the bricks and without the building.

A specific and underappreciated form of data lock-in deserves separate mention, because it is growing quickly and almost no contract addresses it. Where a system contains machine learning models that have been trained or tuned on the buyer's own data over a period of years, that tuning is an asset the buyer paid to create, in the form of the data it supplied and the corrections its people made. Whether the buyer can take it with them is, in most contracts, unaddressed, which in practice means the answer is no. The organization can extract the records that trained the model and cannot extract the model, which means that the accumulated learning of eight years of operation stays with the vendor. Buyers should ask explicitly what happens to tuned models, learned exception logic, and accumulated recommendations at termination, and should not be surprised to find that nobody at the vendor has thought about it either.

The proof-of-concept stage offers a cheap and revealing test that almost nobody performs, and it should become standard practice. Ask the vendor, during the evaluation, for a full export of the demonstration environment: not a sample, not a report, but the complete extract the buyer would receive on termination. Then take it and try to read it. What comes back, and how quickly, tells the buyer more about the vendor's real posture on portability than any clause in the contract. A vendor whose export arrives promptly, in a documented structure that a competent analyst can interpret, has demonstrated something. A vendor that hesitates, that offers a report instead, or that produces an opaque dump of undocumented tables has also demonstrated something, and the buyer has learned it at the only moment when the knowledge is still worth having.

The organizational obstacle to acting on any of this deserves a candid word, because it is the reason the analysis so rarely turns into action. The person negotiating the contract will not be the person facing the renewal in year seven. Procurement leaders rotate, chief information officers move on, and the executive who traded an uncapped uplift for a fifteen percent first-year discount will be gone, promoted on the strength of the discount, long before the compounding arrives. The incentive structure rewards the visible saving now and imposes no cost for the invisible exposure later, which is a description not of bad people but of a badly designed scorecard. An organization that wants different behavior has to measure different things, and the thing to measure is the ten-year committed cost and the exit cost, not the discount.

There is a further and slightly awkward point about who is in the room. Software is usually bought by a coalition of a business sponsor who wants the capability, a technology leader who wants the architecture to work, and a procurement function that wants the price down. Nobody in that coalition owns the question of what happens in year seven, and so the question is not asked. The remedy is to make somebody own it, explicitly, and the natural owner is whoever will still be accountable for the total cost of the estate in a decade, which in most organizations means the chief information officer or the chief financial officer. Until somebody owns the exit, nobody will negotiate for it.

06 The contract mechanics nobody reads

The most effective lock-in mechanisms in enterprise software are not technical at all. They are clauses, and they work because they do not look as though they are about money, which means they are reviewed by lawyers optimizing for risk rather than by procurement teams optimizing for leverage, and they survive into the executed agreement unremarked.

Automatic renewal and the notice window

The standard enterprise agreement renews itself unless the customer gives notice, typically thirty to sixty days before the anniversary and sometimes ninety. Consider what that window actually requires of the buyer. To decline renewal responsibly, the buyer must have evaluated alternatives, run a procurement, selected a replacement, and planned a migration, all of which takes six to twelve months for a system of any consequence. A sixty-day notice window is therefore not a window at all; it is a formality that expires long before the buyer could possibly have exercised it, which means the renewal is effectively automatic and the vendor knows it. The remedy is trivially simple and almost never requested: negotiate a longer notice window, or a right to terminate for convenience with reasonable notice, and the buyer has converted an automatic renewal into an actual decision.

Multi-year terms and the discount that buys them

A vendor will offer a meaningful discount for a three-year commitment, and the discount is real. What the buyer is selling in exchange is optionality, and optionality is worth more than the discount in any market that is changing quickly, which enterprise software currently is. This is a legitimate trade and a buyer may reasonably take it, but it should be taken consciously, with the price of the optionality understood, rather than accepted because the discount appeared in a spreadsheet and the loss of flexibility did not.

The termination provisions that are not there

Most enterprise agreements permit the buyer to terminate for material breach, which is a remedy that requires the vendor to have done something egregious and the buyer to prove it. What they typically do not permit is termination for convenience, termination on a change of control at the vendor, or termination if the vendor materially degrades the product, discontinues a module the buyer depends on, or raises prices beyond a defined threshold. Each of those is a real risk, each has materialized repeatedly in this industry, and each can be addressed with a clause that costs nothing to insert before signature. A change-of-control termination right, in particular, is the clause that the customers in Figure 1 would have given a great deal to possess and could have had for the asking.

There is a further contractual mechanism that is rarely recognized as lock-in and functions as one, which is the license audit. Enterprise vendors reserve the right to audit a customer's usage, and in a complex licensing metric, with named users, processor cores, environments, and indirect access all defined in ways that are difficult to measure, almost any large customer can be found to be out of compliance somewhere. The audit is a legitimate contractual right and it is also, on occasion, a commercial

instrument, deployed with notable frequency at customers who have begun to look elsewhere. A buyer cannot negotiate the audit away and can negotiate its terms: reasonable notice, a defined scope, a cure period rather than immediate penalty pricing, and a cap on retroactive charges. These are ordinary requests and they materially reduce the leverage the audit right confers.

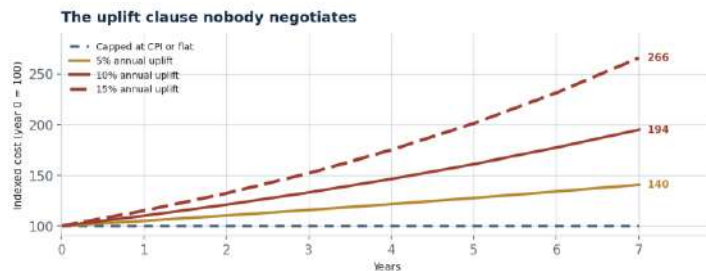
The demonstration in Figure 3 has an important limit that a fair account should state. Egress pricing is not purely extractive; moving data across a network does have a cost, the providers do incur it, and a charge that recovers it is not in itself illegitimate. What is difficult to defend is the asymmetry, since the cost of moving a byte does not depend on which direction it is going, and no provider has offered a persuasive technical account of why inbound transfer should be free and outbound should not. The most parsimonious explanation is the commercial one, and the fact that every major provider arrived independently at the same asymmetric structure is consistent with it. Buyers should price the fee as a switching cost rather than as a service charge, because that is how it functions regardless of how it is described.

Indirect access deserves a specific warning because it has surprised a great many buyers and has produced litigation. Many enterprise licensing models charge not only for the people who log into the system but for the systems and users that access its data indirectly, through an interface, a portal, or another application. An organization that builds a customer portal reading from its enterprise system, or connects a warehouse system that queries it, may discover at audit that it owes licensing for every customer and every device that touched the data, retroactively, at list price. The exposure can be enormous, it is rarely understood at the time of signature, and it functions as a powerful deterrent to integration and therefore to change. The remedy is to establish, in writing and before signing, exactly what indirect access is chargeable and at what rate, and buyers who have not done this should assume the exposure exists.

The complexity of licensing metrics is itself a lock-in mechanism, and it is worth recognizing as one. When a buyer cannot readily determine what it is currently paying per unit of value, it cannot compare that figure to a competitor's proposal, which means it cannot run a meaningful competitive process, which means the incumbent faces no competitive pressure. Opacity is a moat. A buyer should insist on a licensing metric it can measure with its own instruments, should refuse metrics that depend on the vendor's telemetry to compute, and should be suspicious of any structure so intricate that the vendor's own account team needs a specialist to explain it. That intricacy is not an accident of engineering. It is a barrier to comparison, and comparison is the only thing that disciplines price.

07 The uplift clause that compounds

There is one contractual term that costs buyers more than all the others combined, that is present in almost every enterprise agreement, and that receives a small fraction of the attention devoted to the year-one price. It is the annual uplift, and Figure 4 shows why it deserves more.



Annual uplifts in the range of 5 to 15 percent are common in enterprise software renewals. At 10 percent, the subscription roughly doubles in seven years, and the buyer negotiating that renewal has by then accumulated the data, the integrations, and the trained staff that make leaving expensive. The cost is eight times that of the year one discount.

Figure 4. The compounding effect of annual uplift. At ten percent, the subscription roughly doubles in seven years, and the buyer negotiating that renewal has by then lost the ability to leave.

Annual increases in the range of five to fifteen percent are common at enterprise software renewal, and they are frequently not capped in the original agreement at all, which means the vendor is free to propose whatever the market, and the customer's evident inability to leave, will bear. The arithmetic is unforgiving. A subscription escalating at ten percent a year has roughly doubled by year seven. At fifteen percent it has nearly tripled. And the buyer confronting that increase in year seven is in a dramatically weaker position than the buyer who signed in year one, because seven years of data have accumulated, the integrations have proliferated, the staff have been trained, and the processes have hardened. The vendor is raising the price at precisely the moment the customer is least able to refuse, which is not a coincidence but a design.

The correct response is to trade the year-one discount for the cap, and buyers reliably do the opposite. A procurement team is measured on the saving it negotiated this year, which is visible, comparable, and reportable, and it is not measured on the increase it prevented in year five, which is invisible and will be somebody else's problem. So the team pushes hard on the initial price, wins a fifteen percent discount, and accepts an uncapped renewal, which is a trade the vendor is delighted to make because it gives away a known amount now to acquire an unlimited amount later. A cap at the consumer price index, or at a low fixed percentage, is worth far more than the discount, and it is available for the asking in a first-year negotiation.

Two related provisions belong alongside it. The first is a price hold on additional volume: what will it cost to add a hundred more users in year three, and is that price fixed now or determined then, when the buyer has no alternative. A vendor that will not commit to the price of expansion has retained the right to charge whatever it likes for the growth the buyer is planning. The second is a most-favored-customer provision, which commits the vendor not to offer materially better terms to comparable customers. These are standard requests in mature procurement functions and are treated as exotic in most others, which is a reasonable summary of why the same software costs different organizations markedly different amounts.

The notice-window problem has a simple and often-overlooked administrative dimension that costs organizations real money every year. Somebody has to actually know when the window opens. In a large enterprise running hundreds of software agreements, the renewal dates live in a contract repository that nobody reads, and the first anyone hears of a renewal is the invoice, at which point the window has closed and the term has extended for another year. A contract register with renewal dates, notice deadlines, and an owner for each agreement, reviewed quarterly, is administrative housekeeping of the dullest kind and is the difference between having a choice and discovering that a choice was available and has expired. It costs one person a few hours a quarter, and organizations that lack it are paying for software they decided two years ago to stop using.

Perpetual licenses deserve a note of their own, since a generation of buyers grew up with them and a generation of vendors has been steadily retiring them. The perpetual license, whatever its other faults, gave the buyer an asset that could not be taken away and a maintenance decision that could be declined. A buyer paying maintenance on a perpetual license who is unhappy with the vendor can stop paying maintenance and keep running the software, which is an ugly position and is a position. A subscription customer who stops paying has no software. The migration of the industry from perpetual to subscription has therefore transferred a substantial amount of leverage from buyers to vendors, and it has done so under a banner of flexibility, which is a description that would surprise anyone who has looked closely at what happens at renewal.

The multi-year commitment deserves one further caution that is specific to the current moment. Enterprise software is undergoing an unusually rapid capability shift, and a three-year lock on a category that may be reshaped within that period is a materially worse trade than the same lock would have been a decade ago. A buyer signing a long minimum term in a fast-moving category is paying a discount for the privilege of being unable to adopt whatever emerges next, and should at least price that explicitly rather than accepting the term because the discount was attractive. Where a long term is unavoidable, the buyer should seek the right to substitute products within the vendor's portfolio, so that the commitment is to a spend level rather than to a specific product that may be obsolete before the term expires.

08 What it actually costs to leave

Before a buyer can negotiate against switching costs, it has to know what they consist of, and here most organizations are working from a mistaken picture. The public conversation about lock-in has been dominated by egress fees, because they are a visible, quotable, per-gigabyte number that makes for good copy and good regulation. Figure 5 shows what the exit bill actually looks like.

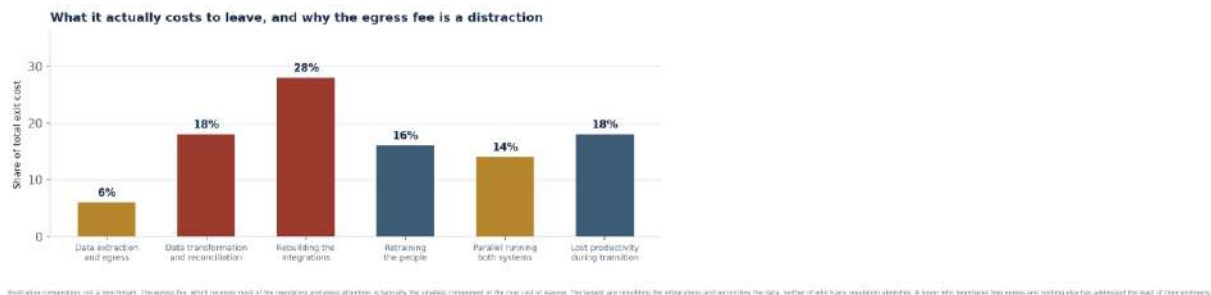


Figure 5. The composition of a real exit cost. The egress fee, which receives most of the attention, is the smallest component.

The largest single item is rebuilding the integrations. Every interface the buyer built to the outgoing system must be built again against the incoming one, and if the estate contains dozens of interfaces, as most supply chain estates do, this is a substantial engineering program in its own right, with its own testing, its own defects, and its own risk. The second largest is data transformation and reconciliation: extracting the records is the easy part, and making them mean the same thing in a system with a different data model, then proving that nothing was lost, is where the months go. Together these two items typically account for around half of the exit cost, and neither is addressed by any regulation, any egress waiver, or any portability obligation.

Beneath them sit the costs that organizations forget to model entirely. Retraining the people who had become expert. Running both systems in parallel for a period, because no responsible organization cuts over a supply chain system in a single weekend, which means paying for two systems at once. And the productivity loss during the transition, which is real, is measurable, and is routinely omitted from migration business cases because it is uncomfortable to write down. Add these to the integration and data work and the picture is clear: the egress fee, which is what everyone talks about, is a rounding error in the true cost of departure.

This has a direct and slightly counterintuitive implication for how a buyer should spend its negotiating capital. A buyer who wins a free-egress commitment and considers the lock-in question addressed has secured perhaps six percent of the value available. The terms that actually reduce exit cost are the ones concerning format and completeness of the export, the ones guaranteeing continued interface access on non-discriminatory terms, and the ones requiring documentation of the data model, because those are the terms that determine whether the extracted data can be used and whether the integrations can be rebuilt against something. Egress pricing is the headline. Format, completeness, and interface continuity are the substance.

The mechanisms compared

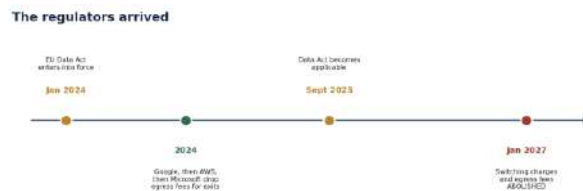
The table below sets the six mechanisms against their origin and the appropriate response. The single most useful column is the last one, because it tells a buyer where to spend the finite negotiating capital available before signature.

Mechanism	Origin	Negotiable?	The right response
Data gravity	Engineered	Yes	Free, complete, documented export in a standard format, at any time
Proprietary formats	Engineered	Yes	Require a documented data model and test the export before signing
Contract mechanics	Engineered	Yes	Uplift cap, real termination rights, a workable notice window
Integration depth	Structural	Partly	Decouple architecturally; secure non-discriminatory interface access
Process embedding	Structural	No	Accept it, and price it into any migration business case
Skills and ecosystem	Earned	No	Accept it. A deep talent pool is a reason to buy, not a grievance

Note that the three engineered mechanisms are all fully negotiable and are the three that buyers most consistently ignore, while the two that cannot be negotiated at all are the ones buyers most often complain about. That misallocation of attention is, in a sentence, the reason most organizations end up with less freedom than they could have had for the same money.

09 The regulators arrived

For the first time in the history of enterprise software, one of the mechanisms described in this article is being dismantled by law rather than by negotiation, and a buyer should understand precisely what is changing and, more importantly, what is not. Figure 6 sets out the timeline.



Under the EU Data Act Regulation (EU) 2023/1654, charges for switching cloud providers are progressively reduced and are abolished entirely from 31 January 2027. Companies' obligations to data portability to existing cloudable formats and a standard transition period. The regulation also allows cloud providers to charge for egress data. Google, AWS, IBM, Microsoft start egress fees for EU.

Figure 6. The regulatory timeline. Switching charges are progressively reduced and then abolished, and the hyperscalers moved ahead of the deadline.

The European Union's Data Act entered into force in January 2024 and became applicable in September 2025. Among its provisions is a set of obligations on providers of data processing services covering switching between providers: an obligation to enable switching, an obligation to provide data in structured, commonly used, machine-readable formats, a standard transition period, and, most consequentially, the progressive reduction of switching charges culminating in their complete abolition from January 2027. From that date a provider may not charge a customer for the act of leaving. The regulation is not a recommendation and does not depend on the goodwill of the industry.

The industry did not wait for the deadline. In early 2024 one major hyperscaler announced that customers leaving its platform entirely could do so without paying egress charges; a second followed within weeks; the third followed within a further week. One provider has since gone further than the regulation requires, waiving egress for certain multi-cloud traffic in the relevant markets rather than only for full exits. Competition authorities in the United Kingdom have separately investigated the cloud market with egress fees and switching costs among their concerns. The direction of regulatory travel is unambiguous and it is worth noting that it has already changed behavior in advance of any enforcement.

Buyers outside the European Union benefit from this indirectly, and the indirect benefit is worth understanding because it can be used in a negotiation. Global vendors rarely maintain two entirely separate commercial regimes, one for regulated markets and one for the rest, because the operational cost and the reputational cost of doing so exceed the revenue protected. A buyer in a jurisdiction with no such regulation can therefore reasonably ask, in a negotiation, for the same terms the vendor is already obliged to offer elsewhere, and the vendor will find it awkward to explain why a right that is mandatory in one market is unavailable in another. That is a serviceable negotiating position, and it costs nothing to adopt.

It is worth being precise about what a machine-readable format does and does not guarantee, since the phrase appears in the regulation and will appear in vendor marketing. A comma-separated file is machine-readable. A comma-separated file containing three hundred tables with names like T_OBJ_4471, columns identified only by integers, reference values encoded against a lookup the vendor

did not export, and no documentation of the relationships between any of it, is also machine-readable, and is close to useless. The obligation that would actually protect a buyer is one requiring a documented data model, and the regulation does not clearly compel that. This is not a criticism of the drafters, who could not reasonably specify the schema of every enterprise application, and it is a warning to buyers who assume that a legal right to portability is the same thing as an operational ability to port.

There is a supply chain dimension to the regulatory shift that is worth flagging, because it is not obvious. Supply chain systems are unusually likely to fall outside the clearest scope of these obligations, because a great deal of the relevant software is not a data processing service in the sense the regulation contemplates but an application whose data model is proprietary by nature. The obligations bite most cleanly on infrastructure and platform services, and least cleanly on the warehouse management system, the transportation system, and the planning engine, which are precisely the systems in which a supply chain organization's most valuable and least portable data actually lives. A supply chain leader reading the coverage of the regulation should therefore be careful not to assume that it applies to the systems that matter most to them.

10 Why the regulation solves less than it appears to

It would be a serious misreading of the previous section to conclude that the lock-in problem is being solved for buyers by legislators. It is not. What is being solved is the smallest and most visible component of it, and the parts that actually determine whether a buyer can leave are largely untouched.

Begin with the scope of the waivers that the industry has already offered, because the detail is instructive. In general they apply only to a full exit: the customer must be terminating the relationship entirely, not moving one workload while retaining others. Some require an application and an approval process rather than applying automatically. Some exclude particular services from the waiver. And none of them affects the ordinary operational egress that a customer pays every day while remaining a customer, which is where the great majority of egress revenue actually originates. The concession that has been offered is real, and it is narrower than the headlines suggested, and a buyer relying on it should read the specific terms rather than the press release.

More fundamentally, recall Figure 5. The egress fee is a small fraction of the true exit cost. Abolishing it removes that fraction and leaves the rest exactly as it was. The regulation does not rebuild the buyer's integrations. It does not reconcile the data into a new data model. It does not retrain the staff, does not pay for parallel running, and does not compensate for the productivity loss during transition. It does not require the vendor to export the configuration, the business rules, the custom reports, or the tuned exception logic, which is where much of the buyer's own investment actually resides. A machine-readable format is a meaningful obligation and it is not the same thing as a usable one, and the gap between those two is where a determined vendor retains most of its position.

The honest summary is therefore that the regulation is a genuine improvement, is worth having, and changes the negotiating landscape at the margin, and that a buyer who treats it as having solved the problem has been lulled. The mechanisms that actually hold customers in place, the depth of the integration, the accumulated data in a proprietary model, the configuration that cannot be exported, the contract that renews itself, and the uplift that compounds, are all exactly where they were, and every one of them remains a matter for the buyer to negotiate rather than for a legislator to fix.

A brief word on the interaction between this regulation and the rest of the world, because it is a live question in every procurement outside Europe. The most common outcome, when a regulator in one large market imposes a requirement on a global platform, is that the platform implements the requirement globally, because maintaining two commercial and technical regimes is expensive and because the reputational cost of explicitly offering worse terms to unregulated customers is significant. This has been the pattern in privacy, in accessibility, and in payments. It is a reasonable expectation here, and it means that a buyer anywhere can ask for the European terms and can point out that the vendor is already building them. Whether the vendor agrees is a negotiation. Whether it is a reasonable thing to ask is not in question.

It should be acknowledged that the industry deserves some credit here, and that the credit is instructive. The hyperscalers moved before the deadline, went in one case beyond what the regulation

required, and did so in a competitive scramble that suggests each of them concluded that being seen as the one that traps customers was worse for business than the egress revenue was worth. That is the market working, and it happened because a regulator changed the payoff. The lesson for a buyer is not that vendors are benevolent; it is that vendor behavior on lock-in is responsive to pressure, and that pressure, whether it comes from a regulator or from a procurement team that has read its contract, produces movement. The absence of pressure produces the opposite.

11 Leverage is spent at signature

Everything in this article converges on a single point about timing, shown in Figure 7, and it is the point that most determines whether a buyer will spend the next decade as a partner or as a captive.



Before signature the buyer has alternatives and the vendor wants the deal. Afterward the data begins to accumulate, the integrations are built, the people are trained, and the alternatives become expensive. Every protection described in this article costs nothing at the request for proposal stage and is close to unobtainable at renewal.

Figure 7. Buyer leverage across the life of a relationship. It is at its maximum immediately before signature and collapses immediately after.

Before signature, the buyer has alternatives, the vendor wants the deal, a competitor is available, and the sales team has a quota. Every term described in this article can be requested in that moment, and most of them will be granted, because they cost the vendor nothing today and the vendor is optimizing for closing. After signature, the position inverts, and it inverts progressively and irreversibly. The data begins to accumulate. The integrations are built. The staff are trained. The processes form. And at each renewal the vendor has a better estimate than the buyer does of what leaving would actually cost, because the vendor can see the usage data, the integration count, and the volume of records, and the buyer has never done the calculation at all.

This is the reason that lock-in is properly understood as a procurement discipline rather than a technology one. The technical facts, the formats, the interfaces, the data model, are the substance of the problem, but the moment at which anything can be done about them is a commercial moment, and it occurs once. An architecture review conducted in year four can identify every coupling and every proprietary dependency with perfect clarity and can do almost nothing about any of them, because the leverage that would have been required to change them was spent in a meeting three years earlier by somebody focused on the year-one discount.

It follows that the buyer's exit strategy must be written before the entry. This sounds adversarial and is not: a vendor that expects to keep a customer by delivering value has no reason to object to terms that make leaving possible, and the better vendors say so explicitly. The vendors who resist these terms hardest are, by revealed preference, the ones whose retention strategy depends on the difficulty of departure rather than on the quality of the product, which is information the buyer has obtained for free, at the only moment when it is still useful.

The strangler pattern deserves a mention here because it is the architectural response to the problem this article describes, and it is available to organizations that cannot contemplate a full migration. Rather than replacing a locked-in system in one movement, the organization progressively routes new capability around it, moving one function at a time to a new platform behind a stable interface, until the incumbent's footprint has shrunk to something that can be retired without drama. This is slower than a

replacement and is enormously less risky, it produces value continuously rather than at the end, and, most relevantly here, it steadily reduces the switching cost while the organization is still a paying customer, which changes the balance of the next renewal negotiation without requiring any confrontation at all.

The architectural corollary is that coupling is a commercial decision as much as a technical one, and it should be made with the commercial consequences in view. Every time a team builds directly against a vendor's proprietary interface rather than against an internal abstraction, it has increased the cost of leaving by a small amount, and the sum of many such decisions is a position from which the organization cannot move. An abstraction layer between the business logic and the vendor's interface costs a modest amount to build and preserves the option to substitute. Most engineering teams will regard this as unnecessary indirection, and in a narrow technical sense they are right; in a commercial sense they are helping to write an extremely expensive option and giving it away.

It follows that a buyer's real protection is a combination of three things, none of which is a regulation. The first is the contract, negotiated at the moment of maximum leverage, which is section twelve of this article. The second is the architecture, which determines how deeply the organization couples itself and therefore how much of the exit cost it is creating with its own hands. And the third is information, in the form of a current, costed understanding of what leaving would actually involve, which is what makes the alternative credible and is therefore the entire substance of leverage at renewal. Regulation helps at the margin, and it will not do any of these three things for anybody.

It is worth setting out plainly what a buyer should conclude about the regulation, since the coverage of it has been enthusiastic and the practical effect is narrower than the enthusiasm implies. Treat it as a floor rather than a ceiling. It establishes that a buyer is entitled to leave without paying a toll, which is worth having and which nobody was going to concede voluntarily. It does not establish that a buyer can leave, which depends on the format, the completeness, the integrations, and the data model, none of which any legislator can specify. And it does not apply cleanly to the application software in which most supply chain organizations hold their most valuable data. A buyer who reads the headlines and concludes that the problem has been handled by Brussels has been handed a floor and has mistaken it for a roof.

The broader lesson from the regulatory episode is about where change actually comes from in this market. Egress fees survived unchallenged for the better part of two decades, through countless procurement negotiations, because no individual buyer had enough leverage to move a hyperscaler on a point of pricing architecture, and because the fee was small enough in any individual contract to be waved through. It took a regulator to change it, and the regulator changed it in months. That is a useful piece of information about the limits of individual negotiation against a concentrated supplier, and it should temper the confidence with which anybody, including this article, asserts that a well-prepared buyer can always negotiate their way out of a structural problem. Sometimes they cannot, and the candid thing is to say so.

12 The terms to negotiate, before you sign

What follows is the operative content of this article: the specific provisions worth requesting, in approximate order of value. None of them is exotic, all of them are obtainable by a buyer of any consequence, and each is close to unobtainable once the contract is executed.

1. **Data export: complete, usable, and free.** Not merely the transactional records but the configuration, the business rules, the historical audit trail, the custom reports, and any trained models. In a documented, non-proprietary, machine-readable format, with the data model documented. Available at any time during the term, not only on termination, and at no charge. Test it during the proof of concept: request an actual export and try to read it.
2. **Interface access, maintained and non-discriminatory.** A commitment that the application programming interface will remain available, documented, and functionally complete for the duration of the term and for a defined transition period afterward, and that the vendor will not degrade a customer's ability to integrate with competing systems. Interfaces are a competitive instrument as well as a technical one.
3. **A cap on annual uplift.** At the consumer price index, or at a low fixed percentage, for a defined period. Worth substantially more over a decade than any first-year discount, and reliably traded away by procurement teams measured on the discount.
4. **Termination rights that reflect real risks.** Termination for convenience with reasonable notice. Termination on a change of control at the vendor. Termination if a module the buyer depends on is discontinued or materially degraded. The customers in Figure 1 would have paid a great deal for the second of these and could have had it for the asking.
5. **A notice window that is long enough to be exercised.** Sixty days is not a decision period; it is a formality. Ninety to one hundred eighty days, or a right to terminate for convenience, converts an automatic renewal into an actual choice.
6. **Price certainty on expansion.** What will additional users, volume, or modules cost in year three, agreed now. A vendor that will not commit has retained the right to price the buyer's own growth at whatever the buyer's inability to leave will support.
7. **Transition assistance, specified.** A defined period, at a defined rate, during which the vendor will support a migration away, including data extraction support and cooperation with the successor. Vendors resist this and grant it, and the resistance itself is informative.
8. **Escrow or continuity provisions where the risk warrants.** For a system on which the business truly depends, provided by a vendor whose independent survival is not assured, source code escrow or a continuity commitment is proportionate. Do not ask for it reflexively; do ask where the concentration risk is real.

A buyer who obtains the first three of these has materially changed the economics of the next decade. A buyer who obtains all eight has purchased something more valuable than any discount, which is the ability to walk away, and therefore the ability to negotiate as an equal every year thereafter.

There is a version of this discipline that goes too far and it is worth naming, because a reader energized by this article may be tempted by it. An organization that refuses all coupling, insists on abstraction everywhere, and selects only products with perfect portability will pay for that freedom in capability, in integration cost, and in speed, and it will most likely end up with a worse operational outcome than a competitor who accepted some dependence in exchange for something valuable. Optionality is an asset and, like any asset, it can be over-purchased. The disciplined position is to know what the dependence costs, to have chosen it deliberately, and to have obtained something worth having in exchange, rather than to have acquired it by accident and discovered it at renewal.

There is one further category of protection that mature buyers negotiate and that deserves a mention, which is the ability to bring in a third party. Many enterprise agreements restrict who may access the system, what tools may be connected, and whether the buyer may engage a competing consultancy or a migration specialist. These restrictions are presented as security and support provisions, and some of them are, and the aggregate effect is to make it harder for the buyer to obtain independent help at exactly the moment when independent help is what they need. A clause preserving the buyer's right to engage third parties of its choosing, subject to reasonable security requirements, is unremarkable and is worth having before the day it is needed.

13 The one-page exit test

There is a simple diagnostic that separates organizations with leverage from organizations that merely believe they have it, and it can be applied to any incumbent vendor this afternoon. It is shown in Figure 8, and it consists of asking whether the organization can describe, on a single page, how it would leave.



Figure 8. The one-page exit test. If the page cannot be written, the organization does not have enough information to negotiate.

The page has to answer five questions. Can the data be extracted, in a format something else can read, without a fee and without the vendor's permission? What exactly would be lost in the move: which configurations, which history, which custom logic, which reports? What would the move cost, in the categories set out in Figure 5, at least to an order of magnitude? How long would it take, and could the business tolerate running both systems in parallel for that period? And, finally, the question that determines whether any of the previous four matter: would the vendor believe the organization might actually do it?

Most organizations cannot write this page for any of their major vendors, and the inability is itself the finding. An organization that does not know what leaving would cost cannot know whether the price it is being charged is reasonable, cannot know how much of its budget is being determined by the quality of the product and how much by the difficulty of departure, and cannot negotiate with any credibility, because a threat that the threatener has not costed is not a threat. The vendor, meanwhile, has done the calculation. Account teams are given retention data, usage depth, and integration counts precisely so that they know how firmly a customer is held. The asymmetry of information in a renewal negotiation is total, and it is entirely self-inflicted.

The page is not a hostile document and does not need to be shared with anyone, least of all the vendor. It is an internal artifact, it takes a competent analyst perhaps two weeks to produce for a major system, and its value is that it converts a vague sense of dependence into a number. Once the number exists, several things follow automatically. The organization knows which vendors hold it most tightly, and can prioritize accordingly. It knows which architectural decisions are increasing the number, and can stop making them. And it can enter a renewal knowing precisely what its alternative is worth, which is the entire substance of negotiating leverage. Everything else is posture.

A practical note on who should write the page, because it will not write itself. The exercise sits awkwardly between functions: procurement does not know the technical dependencies, the architecture team does not know the contract terms, and the business owner knows neither. In practice

the page requires an hour each from three or four people and a fortnight from one analyst who is willing to read the contract and interview the engineers. The reason it is so rarely done is not difficulty; it is that nobody's objectives include it. An executive who wants this to exist should assign it explicitly, to a named person, with a date, in the same way they would assign any other risk assessment, and it will be done.

The list above will strike some readers as a great deal to ask for, and it is worth addressing that reaction directly, because it is the reaction that keeps buyers weak. None of these provisions is unusual in other categories of long-term commercial contracting. A company entering a ten-year facilities lease negotiates termination rights, assignment rights, and rent-review mechanics as a matter of course, and nobody regards this as aggressive; it is simply what a competent party does when committing to a long relationship with an information asymmetry. Enterprise software is treated differently only because the buying process was designed by the sellers, the evaluation is dominated by features, and the contract arrives at the end as an administrative formality. The provisions are not exotic. The category has merely been trained not to ask.

Two further tactics are worth naming because they are cheap, legitimate, and underused. The first is to run a real evaluation of at least one alternative before every significant renewal, even where the organization has no serious intention of moving. The evaluation costs a few weeks of an analyst's time, it produces a costed alternative that makes the negotiating position credible, and it has the useful secondary effect of telling the organization whether the incumbent has quietly fallen behind, which is a question nobody asks about a system that is working. The second is to decouple the timing of the renewal from the timing of the budget, so that the organization is not negotiating under an internal deadline that the vendor can see and wait out. Both of these are ordinary commercial hygiene and both are absent from most software renewals.

It is worth adding a word about the account relationship, since a buyer reading this may worry that a rigorous negotiation will damage it. In practice the opposite is more common. Vendors deal with prepared buyers constantly and respect them; account teams find it far easier to justify concessions internally when the buyer has articulated a specific, evidenced position than when the buyer is simply asking for a better price. The relationship that suffers is the one in which the buyer is passive for six years and then, at a renewal, becomes suddenly and emotionally aggressive because a price increase has arrived. Consistent, informed, unsentimental engagement produces both better terms and better relationships, and the vendors who dislike it are precisely the vendors whose business model this article is about.

14 Negotiation, timing, and a scoring rubric

Two practical points about the negotiation itself are worth making before the rubric, because they are where a prepared buyer converts analysis into money.

Timing is a lever, and it is free

Enterprise software is sold by people carrying quota, and the authority to discount escalates as a reporting period closes. Figure 9 shows the pattern. A buyer prepared to sign in the final days of a vendor's quarter, and equally prepared not to, is negotiating with a counterparty whose incentives have shifted markedly in the buyer's favor. This is not a secret and it is not a trick; it is a well-known asymmetry that most buyers decline to use, usually because their own internal timeline was set without reference to the vendor's. Aligning the two costs nothing.

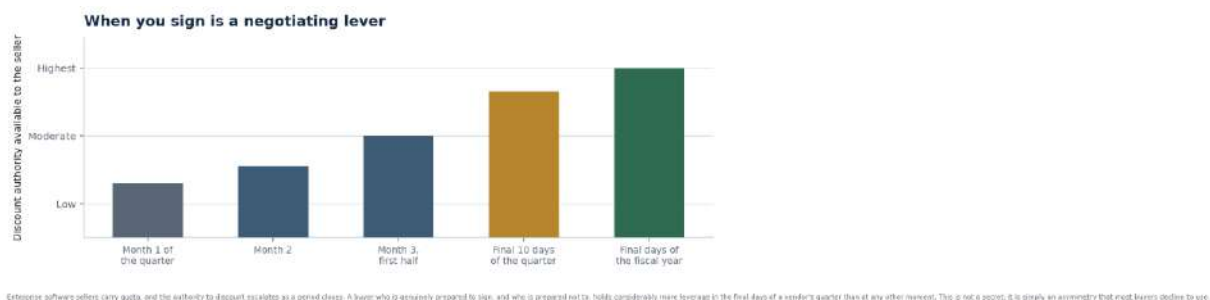


Figure 9. Discount authority across a vendor's quarter. The asymmetry is well known and is used by a minority of buyers.

Credibility, not intention

The most useful sentence in this entire subject is that a buyer does not need to be willing to switch; the buyer needs the vendor to believe they might. Those are different states, and only the second one is required. A buyer who has run a genuine evaluation of an alternative, who has costed the migration, and who can discuss the alternative's capabilities with specificity is credible, whether or not they intend to move. A buyer who has done none of these things and asserts that they are considering their options is not credible, and the account team, who has seen the pattern many times, will price accordingly. Credibility is manufactured by preparation, and preparation is the one input the buyer entirely controls.

A scoring rubric

The dimensions below allow a buyer to assess how much optionality a proposed contract actually preserves, and to compare vendors on something other than price.

Dimension	What good looks like	Red flag
Data export	Complete, documented, standard format, free, any time	On termination only, or priced per gigabyte
Configuration export	Rules, workflows, reports, history all included	Transactional tables only
Interface access	Maintained, documented, non-discriminatory, in writing	Available at the vendor's discretion

Dimension	What good looks like	Red flag
Uplift	Capped at an index or a low fixed percentage	Uncapped, or set at renewal
Termination	For convenience, on change of control, on degradation	Material breach only
Notice window	Long enough to actually run a procurement	Thirty to sixty days before auto-renewal
Transition assistance	Defined period, defined rate, cooperation committed	Not addressed, or refused

The rubric has a second use beyond scoring, and it may be the more valuable one. How a vendor responds to these requests is itself a signal about the vendor. A vendor confident that it will retain the customer by delivering value has little reason to resist terms that make leaving possible, and the good ones agree to them readily and say so. A vendor that fights hardest over export completeness, interface continuity, and termination rights is telling the buyer, without meaning to, exactly where its retention strategy actually lives.

One further preparation is worth making and costs almost nothing: maintain a live, credible view of the alternative. This does not mean running a full procurement every year, which would be absurd. It means knowing what the two or three plausible replacements are, roughly what they cost, what they would do better and worse, and who at the buying organization has actually seen them. A buyer who can say, specifically, that a competing product would cost a stated amount, would require a stated migration effort, and is preferred by a named peer company is negotiating from a different position than one who gestures vaguely at the existence of competition. The information is available, mostly for free, and the effort of maintaining it is trivial next to the value it confers at every renewal thereafter.

The rubric can also be run against incumbent contracts, and the exercise is bracing. Take the three most important software agreements the organization holds, score them against the seven dimensions, and observe how many red flags appear. In most organizations the result is a row of them, and the value of seeing it laid out is that it converts a diffuse unease into a specific list of things that were given away, mostly without anybody deciding to give them away, in negotiations conducted by people who were measured on the price. The list cannot be undone. It can inform the next negotiation, and it can inform the renewal, which is the moment at which a small number of these terms occasionally become available again, usually because the vendor wants something.

Finally, it is worth being explicit about the counterargument that a thoughtful vendor would make against this entire article, because it has force and a fair account should concede it. The argument runs as follows: deep integration is what makes software valuable, a customer who has embedded a system into their operations is a customer who is getting value from it, and a buyer who architects for easy departure is architecting for shallow adoption and will get shallow returns. There is something in this. The organizations that extract the most from their systems are frequently the ones that committed to them, tuned them, built around them, and did not hedge. Optionality has a price, and the price is sometimes paid in performance.

The response is that the two are separable, and that the vendor's argument conflates them deliberately. Nothing in this article suggests that a buyer should refuse to integrate, refuse to configure, or refuse to build expertise. It suggests that the buyer should be able to get its own data out in a usable form, should know what leaving would cost, and should not have signed away its termination rights and its price protection in a contract it did not read. None of those things reduces the depth of adoption by a single percentage point. They simply ensure that the value the system produces is shared according to a bargain that both parties struck knowingly, rather than according to how expensive it turns out to be to walk away. That is not hedging. It is the ordinary condition of a commercial relationship between equals, and it is remarkable that in this one category it has to be argued for.

15 Conclusion: optionality is a purchased asset

The central claim of this article is that the ability to leave a vendor is an asset, that it has a price, that the price is paid at the moment of signature and at no other moment, and that most buyers decline to pay it because they do not know it is on offer. The consequence is that they spend the following decade negotiating from a position whose weakness they did not choose and have never measured, against a counterparty who has measured it precisely.

Nothing in this argument requires a buyer to be adversarial, and a great deal of what passes for tough procurement in this industry is theater that achieves nothing. The terms set out in section twelve are not aggressive; they are the ordinary provisions that any well-advised party would want in a long-term relationship with an asymmetric information structure, and the better vendors grant them without much argument, because they intend to keep the customer by being good rather than by being difficult to leave. The buyer who asks for them is not being hostile. The buyer who does not ask for them is being naive, and the naivety will be priced.

It is worth ending on the observation that this is a domain in which a small amount of preparation produces a disproportionate return. The one-page exit test takes an analyst two weeks. The terms in section twelve take a paragraph each in a request for proposals. The uplift cap costs a procurement team the willingness to trade a visible first-year discount for an invisible seventh-year protection. None of this is difficult, none of it is expensive, and the alternative is to discover, at some renewal several years from now, after an acquisition or a strategy change or simply a bad quarter at the vendor, that the only thing that was ever protecting the organization was goodwill, and that goodwill is not a term of the contract. The customers in Figure 1 learned this. It was available to them, in advance, for the price of asking.

The question to carry from this article into the next contract review is short, and an organization that can answer it has already done most of the work. For each of our three most important software vendors: what would it cost us to leave, how long would it take, and does the vendor believe we could. If nobody in the organization can answer, the answer to the third question is no, and the pricing the organization receives at every renewal from now until it does the work will reflect that fact precisely.

16 Methodology, caveats, and sources

Methodology

- This article synthesizes regulatory texts, provider announcements, competition-authority proceedings, press and customer-group accounts, and practitioner literature on software procurement, current to mid-2026. Supply Chain Research is independent and accepts no payment from the vendors, consultancies, or platforms discussed.
- Contractual recommendations reflect commonly obtainable provisions in enterprise agreements. They are not legal advice and should be reviewed by counsel against the buyer's jurisdiction and circumstances.

Caveats

- The renewal increases shown in Figure 1 are drawn from press accounts and customer-group reporting rather than from audited disclosure, and the vendor has disputed characterizations of its conduct. The figures should be read as a reported range illustrating a dynamic rather than as a verified measurement of any individual contract.
- The exit-cost composition in Figure 5 is illustrative and is not a published benchmark. The relative ordering, in which integration rebuild and data reconciliation dominate and egress is small, is consistent with practitioner accounts of migrations, and the specific proportions will vary substantially by estate.
- Egress list rates change and vary by provider, region, and volume tier. The figures cited are approximate published list prices at the time of writing and should be verified against current pricing.
- The provisions of the EU Data Act summarized here are a simplification of a lengthy regulation, and the scope of the switching obligations, including which services are covered and how the transition period operates, contains detail beyond what is set out above. Buyers relying on these provisions should consult the text and take advice.
- The discount-authority pattern in Figure 9 is a well-attested characteristic of enterprise software sales and is presented as a directional description of seller incentives rather than as a measured finding.

Sources

- [1] European Union. [Regulation \(EU\) 2023/2854 \(the Data Act\), including obligations on switching between data processing services and the abolition of switching charges.](#)
- [2] European Commission. [Data Act explained: switching, portability, and cloud transition obligations.](#)
- [3] Amazon Web Services. [Announcement on free data transfer out when moving off AWS.](#)
- [4] Google Cloud. [Announcement eliminating data transfer fees for customers switching providers.](#)
- [5] UK Competition and Markets Authority. [Cloud services market investigation, including egress fees and switching barriers.](#)
- [6] Gartner. [Research on software asset management, license optimization, and unused entitlements.](#)
- [7] Flexera. [State of the Cloud Report \(cloud spend, waste, and multi-cloud posture\).](#)
- [8] The Register and associated trade press. [Reporting on virtualization licensing changes and customer renewal experiences following the Broadcom acquisition of VMware.](#)

Additional context drawn from customer-group statements and trade reporting on enterprise licensing changes, from practitioner literature on software procurement and contract negotiation, and from provider documentation on data export and egress pricing. Figures originating with interested parties are identified as such and are directional. This article is analysis, not legal, procurement, or investment advice, and contractual provisions should be reviewed by counsel.

Supply Chain Research is an independent, vendor-neutral research platform for supply chain and technology leaders. We accept no payment from the vendors, consultancies, or platforms discussed. This article is analysis, not legal, procurement, or investment advice, and its conclusions should be validated against your own requirements before any decision.