COLDAI
PUBLICATION

AGENTIC HYPERSCALING

The fully autonomous Enterprise Transformation

Dr. Shayan Salehi H.C.

Preface

This is not a book about adopting artificial intelligence. The literature on adoption is already enormous and almost entirely useless. It is useless because it assumes the thing being adopted is a tool and the thing doing the adopting is unchanged. Both assumptions are false. The tool is not a tool. It is a new substrate for cognition. And the firm doing the adopting is, on inspection, mostly a queue of human labour wrapped around that older and slower substrate. Change the substrate and the wrapper has to be torn off and rebuilt.

Most organisations today use machine intelligence the way a Victorian household used electricity in 1895: to illuminate the rooms it had always lit with gas. A drafting assistant here, a support bot there, a coding copilot in engineering, a summariser in meetings. The lamps are brighter; the architecture is unchanged. Within a decade those same households would be redesigned around the wall socket, and the building trade itself would be re-tooled. The interesting story was never the bulbs. It was the rewiring, the load distribution, and the new professions the load distribution made possible. The interesting story now is the same.

The central claim of these pages is severe and simple. Every company is a system for converting information into decisions and decisions into coordinated action. Until very recently, most of that conversion had to flow through the brains of paid humans, because nothing else could perform open-ended cognition cheaply enough to leave a margin. That constraint produced the entire scaffolding of the modern firm: org charts, span of control, middle management, status meetings, escalation paths, the monthly close, the campaign approval round, the quarterly review. None of that scaffolding was sacred. It was load-bearing under a particular cost structure. When the cost structure of cognition collapses, the scaffolding stops being load-bearing and starts being, mostly, drag.

The deeper opportunity, then, is not to help the old organisation type faster. It is to redesign the organisation around machine-speed perception, reasoning, and action. The aim is not the worship of automation. The worship

of automation is a ghost story for journalists. The aim is what I will call disciplined autonomy: the deliberate transfer of repeatable, policy-governed, information-heavy work from human routing to machine execution, while preserving — and enlarging — human responsibility where values, ambiguity, and accountability truly matter.

This book is written against three audiences at once. Against the boards who still believe AI is a productivity line item. Against the consultants who have rebranded change management as transformation and sold the same deck twice. And, most of all, against the operators and founders who have intuited that something foundational is shifting but who lack a language for it that is neither hype nor denial. I have tried to give them a language. It is, at times, an unkind one. I do not apologise for that. The next decade will not be unkind to firms that flinched.

A note on form. Each chapter ends with a single image: not decoration, but argument compressed into a curve. Read them. The book without the charts is half a book.

— Dr. Shayan Salehi H.C.

Contents

I. The End of Headcount as Strategy

Scale used to mean more people. Soon it will mean fewer delays.

For roughly a hundred and fifty years, the firm treated labour as its dominant variable. Hire more people, add more managers, multiply meetings, call the result scale. That formula built the industrial empires we still live inside, but it also installed something more lasting than the empires themselves: a hidden religion. The religion teaches that capability and headcount are the same thing. They are not. Capability is the speed and quality with which an organisation converts information into action. Headcount is one of many possible substrates for performing that conversion, and it has always been the slowest and most expensive one. It only seemed natural because, for most of recorded commerce, it was the only one we had.

The old company was a hierarchy of permissions. Information climbed upward, decisions moved downward, and time leaked out of the system at every rung. Frederick Taylor broke work into motions. Alfred Sloan broke General Motors into divisions. Taiichi Ohno turned the elimination of waste into a religion. Every managerial revolution since the 1880s has been, in essence, a new answer to the same question: how should intelligence move through the firm? And every answer has had to take for granted that intelligence moved through the firm by being routed through human heads. AI does not refine that question. It revokes it. It is no longer merely about how intelligence moves. It is about whether the firm must remain so dependent on human routing at all.

That is why the most important strategic question of the next ten years will not be, *how many employees do we need?* It will be, *which decisions still require a human, and why?* Once executives ask this honestly, the balance sheet starts to look different. Entire departments reveal themselves as relay stations rather than sources of judgment. The company discovers that much of what it has been calling "experience" is really accumulated delay, polished by

repetition until it looks like wisdom. The queue, the status meeting, the escalation path, the overflowing inbox, the monthly close, the campaign approval round: these were never sacred processes. They were artifacts of a world in which open-ended cognition was scarce and expensive. They are coping mechanisms for a constraint that is being removed from underneath them.

The false arithmetic of scale

THE INDUSTRIAL CORPORATION LEARNED to speak in the language of bodies because bodies were once the only scalable container for capability. If a retailer wanted more shelves replenished, it hired more clerks. If a bank wanted more loans underwritten, it hired more analysts. If a manufacturer wanted more coordination between three plants and a railhead, it hired supervisors and the supervisors hired clerks of their own. The budget became a census. The org chart became a theory of reality. Even when companies claimed to value innovation, most still planned in units of salary band and span of control. Strategy ended up disguised as staffing mathematics. Growth meant more requisitions. Control meant more approvals. Quality meant more reviewers. Risk reduction meant another committee.

This arithmetic was perfectly rational in the age of clerks, telephones, and paper queues. It is no longer rational in an age in which open-ended cognition can be purchased by the token. A firm does not become more formidable simply because more humans can touch the same problem; in fact the reverse is usually true. Every additional handoff introduces three things that nobody on the org chart is paid to notice: latency, diffusion of responsibility, and the politics of explanation. The modern enterprise is filled with work that exists only because other work is slow. Status decks exist because operating reality does not travel cleanly. Forecast meetings proliferate because information arrives too late and so must be socially negotiated rather than mechanically reconciled. Managers become traffic police because the underlying system cannot coordinate itself. Whole job titles — director of programme management, head of business operations, chief of staff to the chief of staff —

are concessions to a substrate that cannot keep up with its own ambitions. When executives talk about preserving jobs, they are very often, without realising it, talking about preserving these repair loops.

Ronald Coase explained the firm as an answer to transaction costs. Companies internalised activity because the alternative — coordinating it through markets — was even more expensive than feeding a payroll. Oliver Williamson refined this into a theory of asset specificity and opportunism. Generations of microeconomists treated the boundary of the firm as an equilibrium between two inefficiencies. What none of them anticipated, because they could not have anticipated it, was a third option: a substrate that performs the routing, drafting, monitoring, reconciliation, and execution work of the firm without being part of the firm in the older sense at all. Agentic systems are that third option. They do not merely lower transaction costs at the boundary; they lower coordination costs *inside* it, which means the equilibrium that Coase described is being dragged toward a new resting point that nobody has yet drawn on a graph in a textbook. The enterprise of the future will not look like a larger bureaucracy. It will look like a thinner command layer wrapped around dense computational capability — a small priesthood of irreducibly human judgment standing on top of a great deal of cheap, attentive cognition.

In practical terms, this means founders and boards should stop asking where AI can save labour hours and start asking where it can eliminate organisational latency. The two are not the same. Saving an hour of work inside an inbox is a productivity story. Removing the inbox itself is an architectural one. The first is what most companies are doing in 2026. The second is what the survivors of 2032 will already have done.

The lamp under the gas

Most "AI strategies" you will read in board packs this year describe the lamp, not the rewiring. They treat the model as a feature. They bolt a chatbot onto customer service. They buy a copiloting tool for developers. They run a procurement RFP for "an enterprise GPT". They appoint a head of AI whose job, in practice, is to negotiate licences. These are not transformations. They

are decoration of an unchanged structure. They preserve the org chart, the queues, the approvals, and the meetings — and then add a clever new tool that lets the people inside that machine type slightly faster.

There is nothing wrong with typing slightly faster. It is simply not what is on offer. What is on offer, for the first time since the joint-stock company, is a wholesale renegotiation of the firm as an information processor. The companies that take that on board early will not look dramatic at first. They will look "well run". They will ship faster. They will close the books sooner. They will hire less and grow more. Their rivals will continue to debate headcount plans while the leaders quietly redesign their metabolism. By the time the laggards realise the gap is structural rather than tactical, the gap will already be uncatchable. This is exactly how every previous infrastructure shift unfolded. Railroads, telephones, electrification, ERP, the cloud — none of them announced themselves as revolutions to the firms that lost. They presented themselves, until the very end, as merely operational improvements at the firms that won.

Why the resistance is moral, not technical

THE DEEPEST RESISTANCE TO redesigning the firm around machine cognition is not engineering. It is sociological and, increasingly, theological. Large organisations are status systems disguised as operating systems. Titles, reporting lines, approvals and meetings confer meaning. They tell people who they are. To remove the routing function from a senior director is not, in their lived experience, to free them from drudgery; it is to threaten the architecture by which they understand themselves as important. This is why so many "transformation" programmes stall at the level of pilot projects that are forever pilots. The pilot is psychologically tolerable because it stays on the periphery. The moment the new substrate begins to reach into the centre of the firm — the place where the real meetings are held and the real titles defended — the antibodies activate.

That is why enterprise transformation cannot be sold purely as cost reduction. Cost reduction will be the consequence; it cannot be the slogan.

The slogan must be about a *migration of human labour from administrative work to strategic work*: from process guardianship to system stewardship, from being busy to being decisive, from doing the work to defining the work the machine does. This sounds soft. It is not. It is the only frame in which the people losing their old jobs can bear to participate in building the new ones. A leadership team that cannot articulate this migration will find that it has correctly diagnosed the technology and completely failed at the politics, which is the only place where transformations actually live or die.

There is also a darker version of the resistance, which deserves to be named honestly. Some of the loudest internal opposition to agentic systems will come from people whose job, on inspection, is to introduce delay on purpose. Compliance officers who use process to launder their personal risk preference into corporate policy. Middle managers whose authority depends on being the only person who knows where the spreadsheet is. Long-tenured operators whose competitive advantage is a Rolodex of internal phone numbers. None of these people are villains. They are responding rationally to incentives a previous era set up. But their resistance is not, despite what they will tell you, about safety or culture. It is about the redistribution of legitimacy. Treat it as such. Negotiate accordingly.

What replaces the headcount plan

The headcount plan is the central artefact of the industrial firm. It is the document where strategy is translated into the only currency the firm actually understands: paid bodies. In the agentic firm, the headcount plan is replaced by something stranger and more revealing — the *capability plan*. The capability plan asks, for each function, three questions in sequence. *What outcome are we trying to produce, with what fidelity, at what frequency? Which portion of the work that produces that outcome is now mechanisable, and which portion is irreducibly human? For the irreducibly human portion, what kind of human, with what authority, with what tools at hand, with what right to override the machine?* You will notice that none of these questions begin with "how many". Numbers fall out at the end, often surprisingly small ones, and often distributed in surprising ways: more senior judgment, fewer

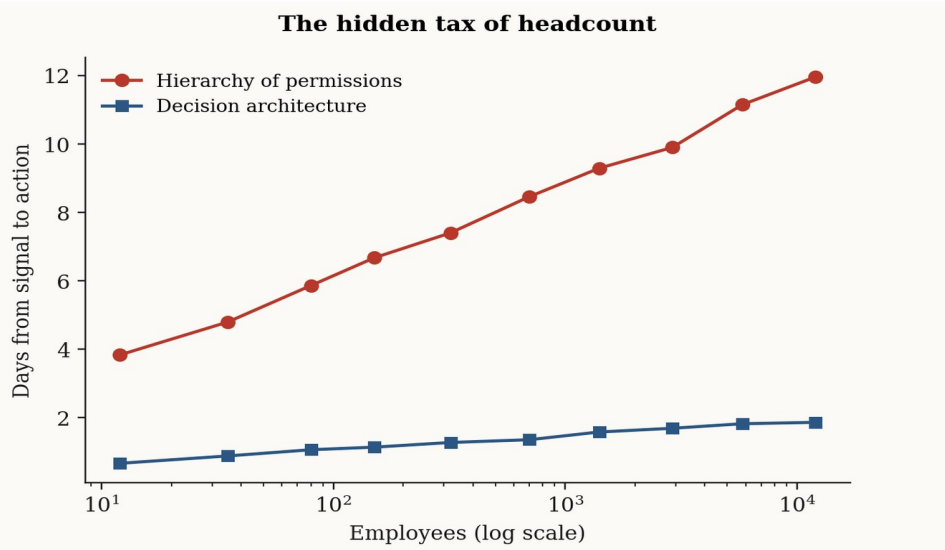
junior relays; more designers and ethicists and operators of last resort, fewer co-ordinators and approvers and compilers of slides.

The capability plan also changes how leaders think about hiring. Hiring stops being about filling boxes on a chart and starts being about whether a particular human's judgment, taste, or accountability is worth wrapping a permanent contract around. A great many roles fail this test on inspection. They were never about judgment at all. They were about coverage — making sure someone, anyone, was awake and responsible at the moment a particular kind of routine work needed doing. Coverage is a job for a machine. Judgment is a job for a person. Most twentieth-century roles are an unstable blend of both, and the agentic firm will, slowly and then suddenly, separate them.

The honest part

Let me say what almost no consultant or vendor will. The transition described in this chapter will not be evenly distributed, it will not be painless, and it will not be optional for very long. The firms that move first will move quietly, because they are not trying to win a press cycle; they are trying to win a decade. They will not announce their headcount cuts because there will be nothing to announce — they will simply have stopped backfilling, stopped opening, stopped escalating. The firms that move late will move loudly. They will hire chief AI officers, run all-hands meetings about the future of work, publish principles, sponsor conferences, and continue to confuse activity with capability. By the time their charts catch up to their press releases, the leaders will have already absorbed the customers and the talent. There is no way to be late to this and elegant about it. Choose now.

The future does not belong to the company with the biggest payroll. It belongs to the company with the smallest gap between signal and action.



Latency rises with size in any firm organised as a relay system. The agentic firm flattens the curve
by removing the relay, not by hiring fewer relays.

Headcount was never the strategy. It was the bill the strategy left behind.

II. From Software to Agency

The decisive software does not wait to be clicked.

Software used to wait. A person clicked the button, entered the field, approved the form, sent the email. The whole twentieth-century paradigm of computing — every spreadsheet, every CRM, every workflow tool, every dashboard — was built on the assumption that the human was the verb and the program was the noun. The program existed in the passive voice. It was used. It was opened. It was queried. It sat there, eternally and patiently, until somebody told it what to do.

Agency changes the posture of code. The program is no longer the noun. It is the verb. It perceives, reasons, acts, and learns within boundaries. This sounds, at first, like a subtle shift — a new feature wrapped around the same old programs. It is not. It is as consequential as the move from paper ledgers to relational databases, and like that earlier shift it will look obvious only in retrospect. Most enterprise software was built for *recordkeeping* and *workflow visibility*. It captured what happened after people did the work, then displayed it back to them in tables and charts. Agentic systems move upstream, into the work itself. They decide what to draft, who to contact, when to escalate, how to investigate, which data to verify, what next action should be triggered, and — crucially — whether the action they were just told to take is still the right one in light of something they noticed five seconds ago. The interface stops being the product. The outcome becomes the product. And the people who used to operate the interface have to find a new role in a play whose script they no longer hold.

A scout, not a map

Think about the difference between a map and a scout. A map is useful, but a map waits. It waits for someone to point at it, decide a route, and walk. A scout travels. A scout returns. A scout reports terrain you did not know to ask about. A scout reorders its own priorities when the wind changes. The map's value is structural; the scout's value is *temporal*. Every previous era of

enterprise software shipped maps. Even the most sophisticated CRM is a map of where the deals are; even the most elegant BI dashboard is a map of what just happened. Agentic systems ship scouts. The strategic implications of replacing your maps with your scouts are not minor. They include, among other things, the obsolescence of the dashboard, the transformation of the manager, and a complete renegotiation of what "weekly reporting" is supposed to mean.

A scout can be wrong. A scout can disappear into the wrong valley. A scout can return with a story you did not authorise it to investigate. The discipline of running a firm full of scouts is not the same discipline as the one we evolved for running a firm full of maps. It is closer, in fact, to the discipline of running a fleet of capable subordinates. There is a literature for that; it is older than computing; it is called *management*, and most of it was written off when software started to do the easy parts of management for free. Now, abruptly, the hard parts of management are about to apply to software itself.

The five postures

It helps to compress the history. There have been five postures in which software has stood in relation to the firm, and each was so totalising in its day that the people inside it had difficulty imagining the next.

The ledger era, roughly 1955 to 1975, treated the computer as a faster account book. Mainframes ran payroll, billing, and inventory. Software was an accounting clerk that did not get tired. The form factor was the punched card and the printout, and the verb was *to record*.

The database era, roughly 1970 to 1995, treated the computer as a place to *find* things, not just to write them down. The relational model was an act of severe intellectual hygiene by Ted Codd; it is hard to overstate how much of modern business depends on the idea that data and the queries against it are separable. The verb in this era was *to query*. A whole industry — Oracle, Sybase, Informix — grew up to sell the right to ask questions of corporate memory.

The workflow era, roughly 1990 to 2015, treated the computer as a *router*. Process automation, business process management, ticketing systems, ERP modules: all were built on the premise that the company was a finite-state machine and that software's job was to move work from one state to the next under human supervision. The verb was *to route*. The dominant emotion was *visibility*. Anyone who has lived inside an enterprise ticketing system knows what this paradigm felt like from the inside: the work itself was unchanged, but everyone was now looking at it.

The SaaS era, roughly 2005 to 2025, treated the computer as a *platform of platforms*. Software became a recurring rental and a network. The vendor sat on your data, the vendor pushed updates, the vendor mediated your relationship with adjacent vendors. The verb was *to integrate*. The unspoken assumption — visible only now that it is being violated — was that the human user remained the active party in every workflow. SaaS was, at heart, the most beautiful and most decadent expression of the click-driven enterprise.

The agentic era, beginning roughly 2023, is not the next layer on this stack. It is the first paradigm in which the program is the actor. This is why every previous mental model breaks down when applied to it. People keep asking *which agent should I buy?* the way they used to ask which database vendor to commit to, and the question quietly mis-states the situation. You do not buy a verb. You build a discipline around the verb. The right question is *what work am I willing to delegate, under what supervision, with what right to revoke, and what is my plan for the moment the agent is wrong in a way I did not anticipate?*

Three failures of analogy

When a paradigm shift is underway, the most expensive mistakes are made by people who reach for the closest familiar analogy and squeeze the new thing into the shape of the old one. Three analogies are doing particular damage in this transition, and they are worth naming.

The first is *the better intern*. Many leaders frame agents as enthusiastic, slightly underqualified, infinitely patient junior employees. The frame is

comforting and almost entirely misleading. An intern learns. An intern asks. An intern fails forward. An agent does none of those things by default; it does them only if the system around it has been built deliberately to let it. Treating an agent as an intern leads to the worst of both worlds: the company gives the agent tasks that require the intern's instinct to escalate, and the agent silently executes the wrong thing at machine speed. The intern frame also produces a peculiar emotional confusion in managers, who find themselves being polite to a system that has no use for politeness and impatient with a system whose impatience would be a feature.

The second is *the smarter macro*. Engineers in particular are drawn to this frame: the agent is just a slightly more flexible automation script. It will fail in the same ways scripts fail, just more often. This is also wrong, but in the opposite direction. A script has no theory of what it is doing. An agent has, at minimum, a working theory of what it is doing — sometimes a wrong one, sometimes a hallucinated one, but a theory nonetheless. That difference is the difference between a tool that breaks loudly when you misuse it and a tool that quietly invents a justification for misusing itself.

The third, and perhaps the most dangerous, is *the new SaaS category*. CIOs trying to govern this transition have a procurement habit. They treat any new capability as a category to be vendor-mapped, evaluated, RFP'd, and contracted. The trouble is that *agency is not a category*. It cuts across every category. It dissolves the line between software and operations. A vendor map of "agentic AI tools" in 2026 will look as quaint, by 2030, as a vendor map of "websites" did by 2010.

The hierarchy of agency

Inside the new paradigm, not all agency is equal. It is useful to distinguish four levels, because almost every confusing conversation about "AI strategy" turns out, on inspection, to be a conversation in which the speakers are at different levels and do not know it.

Level zero is the assistant. It drafts, summarises, retrieves. The human reads, edits, accepts, rejects. The system never acts on the world. This is the level at which most "AI productivity" lives in 2026.

Level one is the executor. The human says *do this*, and the system does it — sends the email, files the form, books the meeting, posts the update. The system's authority is delegated case by case. There is an explicit click before each action.

Level two is the operator. The human authorises a domain — *handle inbound qualification for tier-three accounts*, say — and the system runs inside it autonomously, surfacing exceptions. Most of the value of agentic hyperscaling lives at this level. So does most of the difficulty.

Level three is the deputy. The system not only executes inside a domain but also reshapes the domain over time. It notices that a particular escalation pattern has stopped occurring and proposes retiring the rule that produced it. It notices that a class of contract is now approved manually with no edits 98 percent of the time and proposes raising its own auto-approval threshold. It is, in effect, asking for a promotion. The deputy level is where the discipline of management becomes technical and where governance becomes load-bearing. Almost no enterprise is operating here yet. By 2030 the leaders will be.

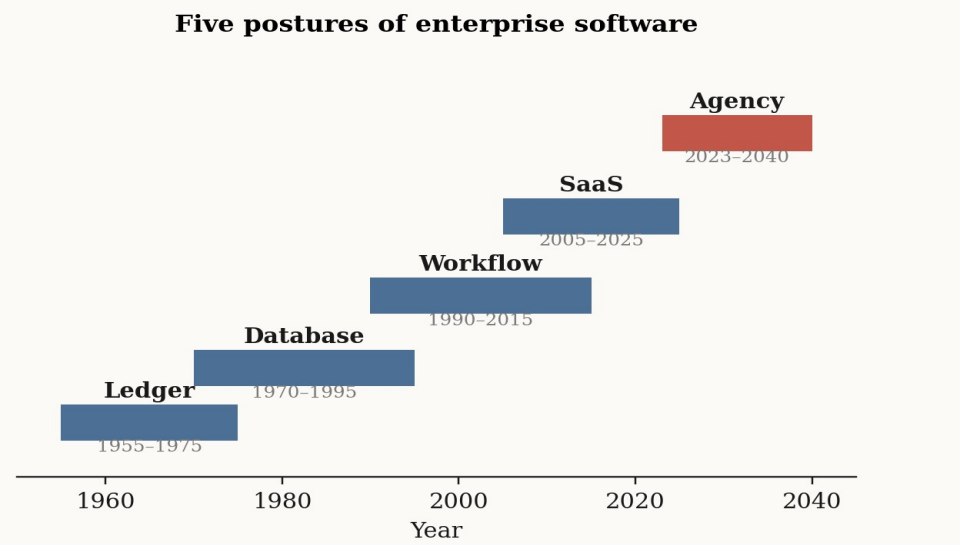
The mistake to avoid is treating these levels as a maturity curve up which every team must climb. They are not. Most workflows belong permanently at level one or two. A few belong at level three. Some belong at level zero forever, because the cost of an error there is greater than any plausible saving. The art is choosing the right level for the right work, and being honest about the difference between them.

The end of the click

THE ULTIMATE MARK OF the agentic era will be the disappearance of the click as the unit of value. For sixty years, the entire economic logic of enterprise software has been priced, designed, and reasoned about in clicks. Conversion funnels were measured in clicks. Workflow improvements were measured in *clicks saved*. The value of training was measured in how many clicks an

employee could perform per hour. The click became an accounting unit, a pedagogical unit, and a moral unit all at once. *Working* meant *clicking*.

The agentic era retires the click as the unit of value, and replaces it with the *outcome*. This is not a vendor's slogan. It is an accounting consequence. When the system can be told *get me a finalised, countersigned MSA with this counterparty by Friday*, the entire chain of intermediate clicks — open the CRM, find the account, draft the email, attach the template, route through legal, chase, follow up, file — collapses into something that is not measurable in clicks at all. It is measurable only in whether the outcome happened, and how good the outcome was, and what it cost the firm in trust and capital to produce. The dashboards are going to look different. The compensation plans are going to look different. The job descriptions are going to look different. And the ones that don't change in time are going to look like fossils embedded in amber: perfectly preserved, perfectly useless.



Each era did not merely add features. It changed what software was for. Agency is the first posture in which the program is the actor, not the instrument.

Every previous era of software made the human faster. This one makes the human optional, and therefore valuable for entirely new reasons.

III. The Firm as a Machine for Decisions

What a company truly is can be found in the pattern of choices it repeats.

A FIRM IS NOT a building. It is not a brand. It is not, as the modern HR function would have you believe, a culture. It is a machine for making decisions under uncertainty. Everything else is packaging — sometimes useful packaging, sometimes beautiful packaging, sometimes packaging worth dying for, but packaging nonetheless. The thing in the box is the decision rate, the decision quality, and the consistency with which the firm reacts to information it did not previously have.

This proposition sounds severe. It is meant to. It is also analytically clarifying, which is more than can be said for most of the language executives use about their own companies. Strategy is a decision about where to play. Pricing is a decision about perceived value and margin. Hiring is a decision about future capability. Customer service is a decision about which promise to honour first. Procurement is a decision about risk, reliability, and cost. Even *culture*, once stripped of its slogans, reduces to a pattern of decisions a group consistently rewards. The firm, then, is a *distributed decision structure*, and to transform it one must begin by finding where the decisions originate, what evidence feeds them, how often they recur, and how often they are merely ceremonial.

This last word matters. A great many things that look like decisions inside a large organisation are not decisions at all. They are *rituals of the decision form*: a meeting that meets, a deck that decks, a vote that votes, but in which the outcome was already foreclosed by structural facts the participants pretend not to know about. I will use the word *ceremonial* throughout this book without contempt — ceremonies have their place — but I will not let ceremony be confused with cognition. The first job of any honest transformation is to look at the firm's decision portfolio and tell the truth about which decisions are real.

Three kinds of decisions

Once you start taking the decision-portfolio view seriously, three kinds emerge. They behave so differently that the same word — *decision* — does them all violence.

The first kind is the *routine determination*: a decision that can be characterised by stable inputs, stable rules, and a stable definition of a correct answer. Whether to approve a low-value expense report. Whether to extend a returning customer's order limit by one tier. Whether to assign an incoming support ticket to billing or to engineering. These decisions are decisions only by courtesy. They are computations that have been put inside human bodies for storage. The bodies were used because, until very recently, no other substrate could read the form, fetch the relevant context, and produce the correct yes or no. That excuse is now gone.

The second kind is the *novel under uncertainty*: a decision in which the inputs are partial, the rules are contested, and the right answer cannot be reached by computation alone but only by judgment, taste, and the willingness to be wrong in public. Whether to enter a new market. Whether to fire a senior executive. Whether to settle a lawsuit or fight it. Whether to rewrite a product around a thesis that has not yet been validated. These are decisions in the proper sense. They are also vanishingly rare in the daily life of even the most exalted corporation. Most CEOs make perhaps a dozen of them a year. Most middle managers, none.

The third kind is the *governance choice*: a decision about how to make other decisions. How fast a refund should be approved, by whom, with what cap. What level of confidence an automated system must reach before acting on its own. What information must be retained, for whom, for how long. Who is allowed to override the model. Governance choices are the decisions that scale, because they apply to all the routine determinations and many of the novel ones beneath them. They are, accordingly, the decisions in which leadership *should* be spending most of its scarce attention. In practice, leadership spends most of its attention on the routine determinations of last resort — the ones that have escalated upward not because they require judgment but because the system below was too anxious to commit.

Once you sort the portfolio into these three buckets, you see something uncomfortable. In the average mid-cap firm, perhaps 90 percent of decision-shaped activity is routine determinations dressed up as judgment, perhaps 9 percent is governance choices ducked because they are politically unpleasant, and perhaps 1 percent is the novel-under-uncertainty work that the entire corporate edifice claims to exist in order to perform. The transformation problem is mostly a problem of moving the 90 percent to a different substrate so that the 1 percent can finally get the oxygen it has been starved of for decades.

What a decision actually contains

To redesign the machine, you have to know what a decision contains. Strip a single corporate decision down to its skeleton and you find six parts, in this order:

A *trigger* — something happened in the world or inside the firm that demands a response.

A *context* — the relevant facts and history at the moment of the trigger, which somebody has to find, summarise, and put in front of the decider.

A *frame* — the implicit theory the decider brings, which determines which facts even count as relevant and which alternatives are even visible.

A *choice* — the actual selection between alternatives.

An *action* — the execution of the choice, which is almost always a sequence of further smaller decisions made by people and systems downstream.

A *learning* — the post-hoc reconciliation of what was decided with what actually happened, which feeds back into the next trigger.

In the legacy firm, all six parts were performed by humans, often by the same human, often badly, often without separating them into distinct steps. The status meeting compressed trigger, context, frame, choice, action, and learning into a single ninety-minute event in a conference room — and then, frequently, did all of them again the following Tuesday because the action step had been quietly skipped. The transformation question is which of the six parts deserves to remain human and which can move to the substrate.

Trigger, context, action, and learning can almost always move. Frame and choice often cannot, and should not. Confusing them is one of the most common implementation errors in agentic projects: a team automates the choice without redesigning the frame, and the system makes faster decisions inside an obsolete theory of the business.

The long tail of corporate cognition

If you plotted every decision made inside a typical enterprise in a quarter, sorted from most routine on the left to most novel on the right, you would see a brutal long tail. Tens of thousands of decisions per day on the left, all very similar. A few thousand per day in the middle. A handful per week on the right. The legacy firm built a single career ladder — analyst, senior analyst, manager, director, VP — and asked it to handle all three regions of this distribution. Junior staff handled the high-frequency left-hand mass and were promoted on the basis of their accuracy and stamina; senior staff drifted toward the right-hand tail and were promoted on the basis of their willingness to sign things. The whole structure assumed that the only way to develop good judgement at the right of the distribution was to spend ten years reliably executing the left of it.

Agentic systems break the assumption that produced the ladder. The left-hand mass can be served, faster and more consistently, by a substrate that does not get promoted. The right-hand tail is where humans should be congregating — but the path to the right-hand tail can no longer go through the left-hand mass, because the left-hand mass is no longer staffed. This is not a small point. It is a generational restructuring of what early-career experience inside a firm looks like, and it deserves its own honest reckoning rather than a slogan about reskilling. Most "reskilling" programmes I have seen are euphemisms for *we don't know what to do with the people whose entire career was the relay we just retired*. The honest answer involves smaller cohorts, longer apprenticeships, earlier exposure to genuinely novel decisions, and an acceptance that fewer humans will be needed at all — but

the ones who are needed will be expected to be unusually capable, unusually early.

The ceremonial overhang

Every firm carries an overhang of ceremonial decisions: meetings, votes, reviews, approvals, sign-offs, and committees that do not change outcomes but consume time, attention, and political capital. The overhang is rarely visible to the people inside it, because the ceremonies are functionally identical to the real thing. They use the same vocabulary. They occupy the same calendar slots. They produce the same minutes. They are, in fact, the thing the meeting culture was *designed* to be in the first place — a substitute for cognitive work that cannot be performed at the speed the business now requires.

A useful diagnostic is the *counterfactual test*: for each recurring meeting, decision forum, or approval chain, ask what would change about the outcome if the meeting did not happen at all and the default answer simply went through. If the answer is *almost nothing*, the meeting is ceremonial, and the people inside it are not making decisions; they are performing them. The agentic firm cannot afford to keep paying for performances. Some of them are worth keeping for political or symbolic reasons; the test is whether you keep them deliberately, with eyes open, rather than by inertia.

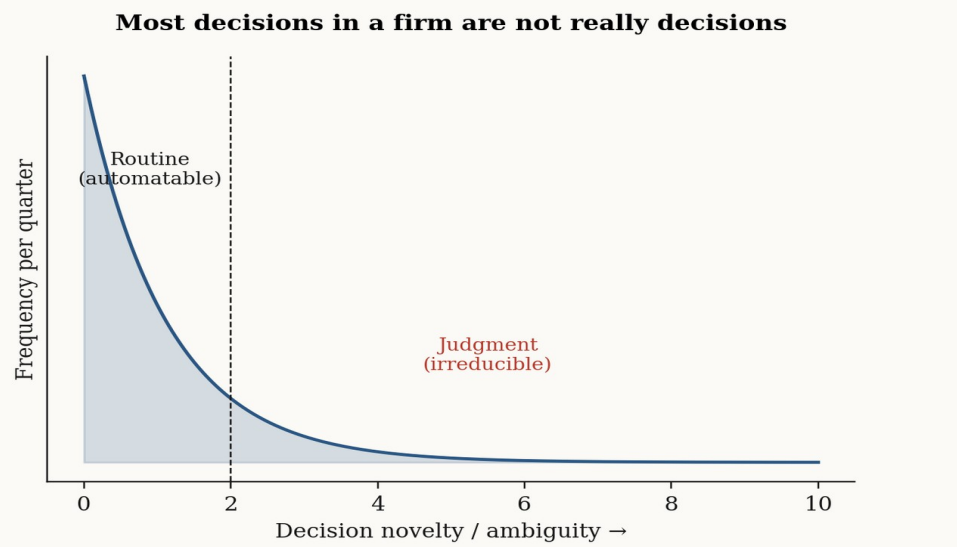
Carlota Perez observed that the productive deployment phase of any technological revolution requires the institutional matrix to be rewritten — not just the tools but the rules of the game, the categories of professional life, the rituals of legitimacy. This book sits inside that observation. The single most expensive hangover from the previous era is the ceremonial decision, and most large firms in 2026 are still deciding everything ceremonially even when the substrate could decide it for real.

The diagnosis before the architecture

Before you build any agent, before you choose any platform, before you draw a single architecture diagram, do this. Sit with a pen and one trusted

operator from each function, and write down every decision the function made last month. Not every task. Every decision. For each one, mark whether it was routine, novel, or governance. Mark who made it and who *thought* they made it. Mark how long it took from trigger to action, and how much of that time was waiting on someone or something. Mark whether the decision turned out to have been correct, wrong, or impossible to evaluate. Then count.

The output of this exercise will be the most uncomfortable document your leadership team has produced in years. It will reveal which functions are mostly cognition and which are mostly ceremony. It will reveal which managers are mostly routing and which are mostly judging. It will reveal where the firm has been congratulating itself on a culture of empowerment when in fact it has built a culture of unaccountable consensus. And it will give you, finally, a map of what to redesign — not by department, not by tool, not by vendor, but by *decision*. Everything that follows in this book is built on the assumption that you have done this exercise. If you have not done it, you are not ready to build agents. You are ready to buy demos.



The decision portfolio of any large enterprise is wildly skewed. The mass on the left is what agents should absorb. The thin tail on the right is what humans should be paid for.

Show me your decision portfolio, and I will tell you what your firm is. Your org chart is fiction.

IV. The Enterprise Nervous System

A firm without nerves is a firm that learns by autopsy.

Most enterprise architectures are skeletons. They were designed to hold the body up. They were not designed to feel. The IT estate of a mature corporation — its ERP, its CRM, its data warehouse, its lake, its lakehouse, its planning system, its dozen specialised tools per function — is a brilliant exoskeleton for recording what already happened, and a profoundly clumsy organism for noticing what is happening now. It can tell you, with great accuracy and a six-week delay, that something went wrong in the European supply chain in the second week of last quarter. It cannot tell you that something is going wrong this morning, in the same way that a corpse cannot tell you it is in pain.

The agentic firm needs an extra layer the legacy firm did not — a *nervous system*. The metaphor is exact, not poetic. A nervous system is what allows a body to sense the world continuously, route signals to the right organ at the right speed, react before damage propagates, and remember the reaction so that next time the body responds slightly differently. A nervous system does not replace the skeleton; it runs through it. It does not replace the organs; it co-ordinates them. And it makes the difference between an animal that can run from a predator and an animal that can be prepared as dinner. The difference between most enterprises in 2026 and most enterprises in 2032 will be whether they have built one.

Skeletons and nerves

It is worth being precise about what the legacy stack actually does well. ERP is excellent at recording transactions and producing auditable reports of them. CRM is excellent at storing the agreed-upon facts about a customer relationship as those facts existed at the moment somebody last clicked the save button. Data warehouses are excellent at letting analysts ask questions

about the past in a structured way, provided the questions were anticipated by whoever built the schema. None of these systems were built to *react*. They were built to *record*. The implicit assumption was always that the reaction would happen elsewhere, in a meeting, in a person's head, in a Tuesday morning ritual where someone read a report and decided what to do next. The system's job ended when the report was rendered. The human's job began.

This division of labour was sane in 1995. It is increasingly insane in 2026. The volume, velocity, and variety of operationally relevant signals reaching a modern enterprise are now too high for the human-mediated reaction loop to keep up. Not because the humans are stupid — many of them are extraordinarily good — but because there are not enough hours in the week for senior people to read the reports the systems are producing, let alone decide what to do about them, let alone ensure that what they decided was actually executed downstream. The reaction layer is the bottleneck. The skeleton is fine. The nerves are missing.

A nervous system has four properties that matter here. *Continuous sensing*: it does not wait to be queried, it streams. *Selective routing*: it does not show every signal to every organ, it sends each signal to the part of the body that can act on it. *Prioritised reaction*: it allows reflexes to fire on important signals without waiting for higher cognition to deliberate. *Memory and adaptation*: it changes its routing and its reflexes over time in response to experience. Build these four properties into the gap between your existing systems and your agents, and you have begun to build a nervous system. Skip one of them and you have built a marketing campaign about a nervous system.

Sensing without surveillance

CONTINUOUS SENSING IS EASY to misunderstand. It is not surveillance. It is not "log everything in case we need it later". It is the disciplined construction of a small number of well-chosen signals that meaningfully reflect what is happening in the parts of the business that matter, sampled at a frequency that lets the organisation react in time. The discipline is in *what to ignore*. A

firm that streams everything ends up with a data lake that no one swims in. A firm that streams the right twenty things ends up with reflexes.

In sales, the right signals might be intent shifts in target accounts, changes in contract usage patterns, and silences from accounts that should be talking. In operations, they might be deviations between planned and actual cycle times, supplier reliability drift, and quality regressions detected at machine vision stations. In finance, they might be margin deviations at the SKU and channel level, anomalies in collections behaviour, and early warnings of covenant pressure. In product, they might be activation drop-offs, retention regressions in newly released features, and the slow accumulation of small defect signals that have not yet become incidents. Each of these has the property that *something can be done about it now if it is noticed now*. That is the test of a sense-worthy signal. Streaming a metric that nobody can act on within its decay window is not sensing. It is data hoarding with extra steps.

Routing as a first-class problem

Routing is the part of the nervous system that legacy enterprise architecture treats as an afterthought. The implicit routing assumption in the old stack is *every signal goes to a dashboard, and somebody is presumably looking at the dashboard*. Nobody is looking at the dashboard. They were never going to look at the dashboard. Even if they did look at the dashboard, they would not look at the right cell at the right second. Dashboards are an honest acknowledgement that the system has given up on routing and outsourced the routing problem to whichever human happens to feel guilty about ignoring it.

The agentic firm replaces dashboards with *routes*. A route is a deliberate, codified pathway from a particular kind of signal to a particular kind of decision-capable destination. The destination might be a human (when judgment is required), an agent (when policy can absorb the case), or a workflow (when the action is mechanical). Routes have priorities, escalation paths, quiet hours, and deliberate redundancies. They are designed objects, not spontaneous artifacts. They are also auditable, which matters more than

most teams realise: when something goes wrong inside an automated firm, the question is not *who failed* but *which route fired, and was the route correct*. A firm whose routes are not designed, named, owned, and version-controlled is a firm that has automated its accidents.

The most underrated property of well-designed routes is that they make *silence* legible. In a legacy architecture, the absence of a signal is invisible — the dashboard simply shows whatever it shows, and nobody notices the thing that did not happen. In a route-based architecture, you can build an explicit reflex around *expected signals that did not arrive*: the customer who did not log in this week, the supplier who did not confirm the order, the close package that did not include the European subsidiary. Most operational disasters announce themselves as silences before they announce themselves as crises. A nervous system that cannot hear silence is half a nervous system.

Reflexes versus deliberation

THE BIOLOGICAL NERVOUS SYSTEM is built around a hierarchy of response speeds. The reflex arc fires in milliseconds without consulting the brain. The midbrain handles rapid, coarse responses. The cortex handles slow, deliberative ones. Different signals get different latencies and different organs make the call. Pain from a hot stove does not wait for an executive decision. The decision to leave a job does.

The enterprise nervous system has the same structure or it is not a nervous system. Some signals — payment failures, security alerts, regulatory triggers, customer-impacting outages — must trigger instantaneous, codified reflexes. The reflex is not a *decision* in any meaningful sense; it is a pre-authorised response that the firm has already agreed to take in this class of situation. Other signals — strategic shifts in a market, the third quarter in a row of margin compression, the slow erosion of activation among newly acquired customers — must trigger deliberation, in which humans and agents collaborate to interpret and respond. Confusing the two is a category error. A firm that deliberates over reflexes is paralysed; a firm that reflexively

responds to deliberative signals is reckless. The art is knowing which is which, in advance, and writing it down.

Memory and adaptation

A NERVOUS SYSTEM WITHOUT memory is just a switchboard. The fourth property — memory and adaptation — is what turns continuous sensing and disciplined routing into something that *learns*. Three kinds of memory matter. *Episodic memory*: what happened, in what order, with what consequence, indexed so that later cases can be compared. *Semantic memory*: the firm's accumulated understanding of how its parts interact, codified in something more durable than a slide deck. *Procedural memory*: the codified policies, routes, and reflexes themselves, versioned over time.

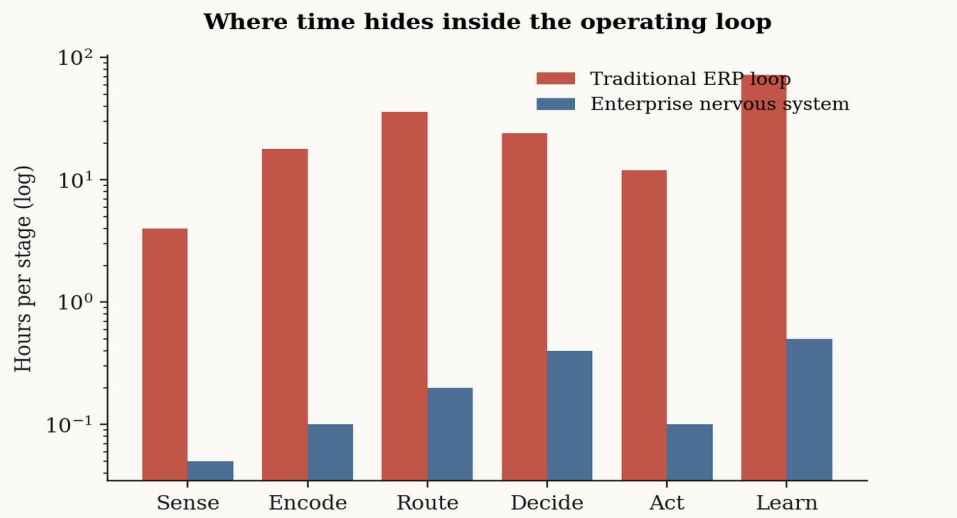
Most enterprises today have, at best, the first kind, scattered across a dozen systems and unjoinable. They lack the second almost entirely; their semantic memory lives in the heads of long-tenured employees and walks out the door at retirement. Their procedural memory is the company's policies, which are out of date, contradicted by practice, and read by no one except in litigation. The agentic firm has to build all three deliberately. The semantic layer in particular — the structured representation of what the business is and how its pieces relate — is where most ambitious AI projects either succeed or quietly fail. If your agents do not know what your firm believes about itself, they cannot act on its behalf without inventing their own beliefs, and you will not like the beliefs they invent.

Where to begin

A PRACTICAL SEQUENCE: PICK one operational loop that matters and that is currently slow. Not the most strategic loop. Not the most visible loop. The loop where the gap between *signal exists* and *action is taken* is most embarrassing if you are honest about it. For most companies this is somewhere in the order-to-cash cycle, or in lead routing, or in incident response, or in the supply chain reconciliation between two adjacent tiers. Build the four properties around that single loop. Continuous sensing of the relevant signals. Designed routes

from each signal class to a decision-capable destination. Codified reflexes for the high-frequency cases. Memory that lets the system get better at this loop over time. Then watch what happens to the loop's median latency, its tail latency, and its error rate over the next two quarters.

What you will find — and this is the part that nobody quite warns you about — is that the rest of the firm starts to look obviously broken in comparison. The loop you fixed becomes a bright object against a background of slowness, and the people working in the slow background will either demand the same treatment for their own loops or will resist the comparison. Both reactions are useful. The first gives you a pull rather than a push. The second tells you where the political work is. Either way, the nervous system has begun to grow.



The largest savings rarely come from the act itself. They come from collapsing the routing, encoding,
and learning stages, which is exactly what the legacy stack treats as inviolable.

The skeleton holds the firm up. The nerves let it survive.

V. The Agentic Stack

Every miracle becomes a stack once it has to work on Tuesday.

In the first software era, companies asked which application to buy. In the agentic era, they must ask which layer of cognition they are actually building. The two questions have nothing in common, and the executives who answer the second by reaching for the procurement habits formed in answer to the first are heading for a particular and very expensive disappointment.

The stack of the autonomous enterprise is not, despite what the slide decks of vendors imply, a model on top of some data with a chat interface in front. It is a layered architecture of perception, memory, reasoning, action, supervision, and measurement, in which each layer has its own discipline, its own vendors, its own internal politics, and its own unique failure modes. Leaders who fail to distinguish these layers buy point solutions that demo beautifully and collapse under operational load. One tool has a better model. Another has a prettier interface. A third promises orchestration. Meanwhile nobody can explain where the company's memory lives, who governs the tool permissions, how confidence thresholds get enforced, or how exceptions are routed when the agent encounters a case it has not seen before. The result is not a system. It is a piece of theatre that looks like a system on a Wednesday afternoon and stops looking like one by Friday.

This chapter is a tour of the six layers. It is also, by implication, a defence against the idea that any single vendor can provide all of them honestly. Companies that try to buy a complete stack from a single vendor end up either with a stack that does not fit their reality or with a vendor whose lock-in becomes a strategic dependency. Companies that try to assemble all six layers in-house from scratch end up burning two years on plumbing that will be commoditised by the end of the third. The right answer, almost always, is a deliberate hybrid in which a small number of layers are owned, a small number are bought, and the seams between them are designed and defended like state borders.

Layer one: substrate

The substrate is the firm's data, telemetry, documents, and APIs — everything the agents will perceive and act on. This is the unsexy layer. It is also, with great consistency, the layer at which most ambitious agentic projects fail. The model is not the bottleneck; the substrate is. If your contracts are scattered across seventeen drives, your customer records are inconsistent across CRM and billing, your product telemetry is sampled and lossy, and your APIs were designed in 2014 by someone who has since left the company, then no model — however brilliant — can give you reliable agency on top of them. It will simply give you fast hallucination, which is worse than slow human error because the firm will trust it for longer.

The substrate layer is where the quiet work happens. Cleaning up master data. Defining the canonical record for a customer, a contract, a supplier, an asset. Building the API surfaces by which agents will read and write that record. Instrumenting the operational systems that have, until now, been silent. None of this is glamorous. All of it is load-bearing. The firms that win the agentic decade will be the firms that did this work in 2024 and 2025 while their competitors were buying chat interfaces. They will look, in retrospect, as if they got lucky with their AI strategy. They did not get lucky. They paid the substrate tax early.

Layer two: memory

Memory is the layer most people skip and almost everyone underestimates. In a single-shot model interaction, memory does not matter; the model takes its inputs, produces its outputs, and forgets. In an agentic system that runs against the same business problem for weeks or months, memory is the difference between competence and amnesia. There are at least three kinds, and they need to be distinguished and architected separately.

Vector memory — the ability to retrieve relevant content based on semantic similarity — is what most teams build first, because it is the easiest. It is necessary but radically insufficient. *Episodic memory* — a record of what

happened in earlier interactions, who did what, what the outcome was, what the agent learned — is what allows an agent to build on its own previous work rather than restarting from zero each session. Almost no enterprise builds this seriously. *Structured memory* — explicit, queryable representations of the firm's entities, relationships, and policies — is what allows the agent to reason about the business as a thing with a shape, rather than as a soup of documents. The graph databases of the early 2010s, the enterprise ontologies of the early 2000s, the knowledge graphs of the mid-2010s — all of them were early, doomed attempts at this layer. They failed because the cognition layer above them was not yet ready to use them. It is now.

A good memory layer is the difference between an agent that can be given a quarter-long objective and an agent that has to be re-briefed every Monday morning. The cost of building memory deliberately is high. The cost of building agents on top of no memory is higher and is paid in a different currency: the currency of trust eroded the first time the agent forgets something the firm thought it knew.

Layer three: reasoning

The reasoning layer is the one everyone talks about, and the one that gets less and less of the marginal value over time. Models matter. They will continue to matter. But the differences between frontier models are narrowing for almost all enterprise use cases, while the differences in how well a firm has built its substrate, memory, action, and supervision layers are widening. Treating the reasoning layer as the place where strategic differentiation lives is, in 2026 and beyond, a mistake of perspective. It is like a nineteenth-century factory owner believing that the strategic advantage of his firm lies in which specific brand of steam engine he installed.

That said, the reasoning layer has its own internal architecture and its own choices. *Which models for which tasks* — frontier models for novel reasoning, smaller models for high-volume routine work, specialised models for vision, code, or domain-specific extraction. *Which prompting and planning strategies* — single-shot, chain-of-thought, multi-step planners, tool-using agents,

hierarchical decomposition. *Which guardrails* — constraint satisfaction, output validation, refusal policies, confidence thresholds. The discipline of the reasoning layer is matching the cognitive cost to the cognitive requirement. Using a frontier model for a task a small model could do is not a sin, but it is a tax. Using a small model for a task that needs frontier reasoning is an unforced error. Most teams, when they actually measure, discover that they are doing both at once on different parts of the same workflow.

Layer four: action

THE ACTION LAYER IS where agents stop being interesting toys and become operational systems with consequences. It is the layer that turns *the agent recommends sending an email* into *the email has been sent*. Everything that can happen in the world as a result of agentic decisions — sending messages, writing to records, executing transactions, creating files, modifying configurations, calling external services — happens here.

Action is the layer with the largest gap between its conceptual difficulty and its operational difficulty. Conceptually, it is easy: agents need permissioned, audited connectors to the systems they need to act on. Operationally, it is the place where every legacy security architecture, every compliance regime, every change-management process, and every internal political fight about who controls what comes together in a single explosive bundle. Most of the failures I see in agentic projects in 2026 are not failures of model quality. They are failures of the action layer: connectors that do not exist, permissions that cannot be granted at the right granularity, audit trails that cannot answer the question *who actually did this*, and rollback mechanisms that were not built because nobody believed the system would ever need them.

The single most important architectural principle in the action layer is *idempotency and reversibility*. Idempotent actions can be safely retried. Reversible actions can be safely undone. A system in which every action is both is a system that can fail loudly and recover; a system in which neither is true is a system that will eventually do something irrevocable on a Tuesday

that nobody wanted. Build the action layer assuming you will be wrong about everything else.

Layer five: supervision

Supervision is the layer that makes the difference between a system that is trusted and a system that is theatrical. It includes evaluation (how do we know the agent is doing what it should), policy enforcement (what is the agent allowed to do, under what conditions), and intervention (how can a human stop, redirect, or correct the agent in real time). Supervision is also where most enterprise AI projects discover, late and expensively, that they have built a system whose behaviour they cannot explain.

The hardest part of supervision is not the technology. It is the discipline of writing down, in advance and in detail, what *correct* means. A great many teams skip this step because they assume correct behaviour is self-evident. It is not. Correct behaviour for a sales-development agent is not the same as correct behaviour for a contract-redlining agent or for an inventory-replenishment agent, and the difference is not captured in any out-of-the-box metric. The teams that build supervision well are the ones that treat *eval design* as a first-class engineering activity and update their evals continuously as they discover edge cases. The teams that build supervision badly are the ones that ship the agent and then look at the results once a month in a slide.

Layer six: measurement

Measurement is what closes the loop. It is the layer that converts the agent's behaviour into evidence that can be acted on by the rest of the stack. It includes outcome tracking (did the work produce the result the firm wanted), audit trails (can we reconstruct what happened, in what order, with what authority), and learning loops (does the system get better over time as a function of what it observed in production).

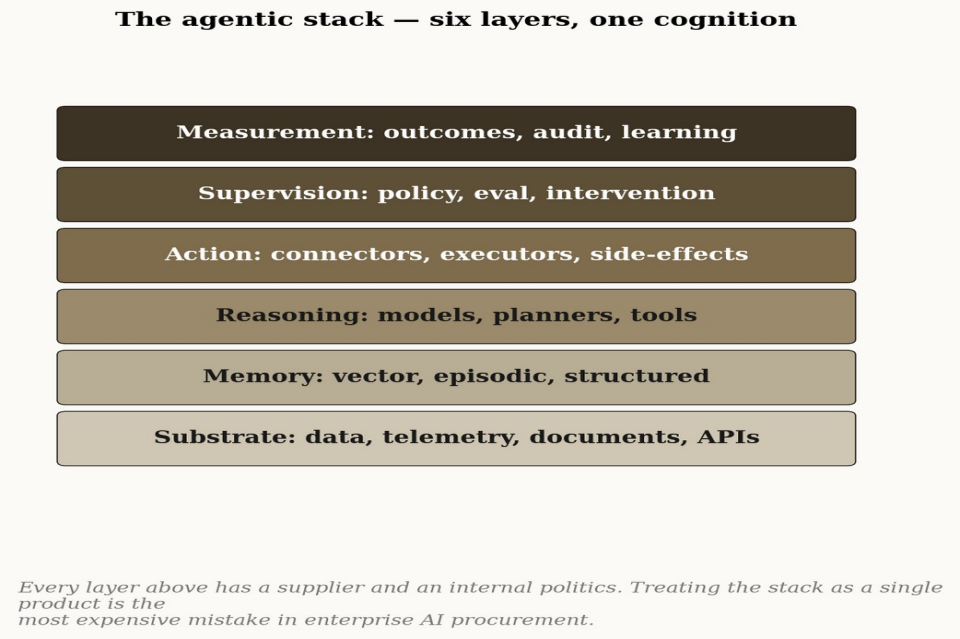
The principle to internalise is that measurement is not optional and is not a feature you add later. Agents that are not measured will silently drift, and the

drift will be invisible until it is expensive. The supervision layer above tells you whether the agent is doing the thing you wanted *in the abstract*. The measurement layer tells you whether the thing you wanted is *the right thing to want*. Without both, the firm has bought a system that may be optimising the wrong objective at machine speed.

How to think about the stack as a whole

THE TEMPTATION, WHEN YOU see all six layers laid out, is to try to build them in parallel and ship in eighteen months. Resist this. Build them in vertical slices. Pick one workflow, build all six layers for that one workflow, ship it, learn from it, then expand. Each vertical slice teaches you something the slide-deck version of the architecture would have hidden — usually something embarrassing about the substrate or the action layer. Each slice also produces something the firm can actually use, which is the only currency in which transformation accumulates.

The other temptation is to outsource the stack to a vendor who claims to provide all six layers. Some vendors will eventually provide a credible version of this. Most current claims do not survive contact with a real enterprise. The right test is not the demo. The right test is to ask the vendor to walk you through a single failure case end to end — what happens when the model is wrong, who notices, how is it corrected, what is the audit trail, who is liable, what is the rollback path. If the vendor cannot answer this test in detail, the vendor is selling you a layer-three story and implying the rest. You will be the one paying for the implication.



Every layer above has a supplier and an internal politics. Treating the stack as a single product is the most expensive mistake in enterprise AI procurement.

The model is the vocabulary. The stack is the grammar. Without the grammar, the vocabulary just argues with itself.

VI. The Data War

The moat is not the data. It is the loop the data feeds.

Every five years, the technology press declares a new substance to be "the new oil". In the 2010s it was data. The metaphor was always lazy and is now actively misleading. Oil is fungible. Oil is consumed once. Oil's value is captured by whoever owns the well. Data, by contrast, is non-rivalrous, infinitely copyable, rapidly perishable, and almost worthless without the surrounding apparatus that turns it into action. Treating data as the strategic asset of the agentic era is therefore exactly as wise as treating sand as the strategic asset of the semiconductor era. There is sand involved. The sand is not where the value lives.

What lives there instead is the loop. The strategic asset of the next decade is not the data your firm has accumulated. It is the *closed loop* between an action your firm takes, the consequences of that action, the reconciliation of those consequences with what you expected, and the updated policy that fires the next time the same situation occurs. A firm with mediocre data and a tight loop will outrun a firm with excellent data and no loop, every single time, by a margin that grows monthly. This is the central insight of the chapter and the rest of it is consequences.

Why the data-as-moat thesis was always weak

The data-as-moat thesis came from a particular historical moment — roughly 2012 to 2020 — in which a small number of consumer internet firms had access to user behaviour data at a scale nobody else could match, and used that data to train models that nobody else could train. The thesis hardened into orthodoxy by analogy and survived long after the conditions that made it true had begun to dissolve. The conditions dissolved for three reasons.

First, the marginal value of additional general-purpose training data has fallen sharply. Frontier model performance is now bounded much more tightly by training compute, post-training technique, and architecture than by raw pre-training data volume. The advantage of having an extra terabyte of general text is, today, close to zero for the firm doing the training and close to zero for the vendor consuming it.

Second, the *specific* data that matters for an enterprise — its contracts, its customer interactions, its operational telemetry, its policies — was always the data nobody else had, and was always available to the firm that owned the operations producing it. The data-as-moat thesis described a world in which raw scraped data was scarce. The world of enterprise data has been the opposite for decades: every firm has had its own private corpus, and almost none of them have done anything serious with it.

Third, and most importantly, the value of data in the agentic era is conditional on something the data-as-moat thesis ignored: the existence of an action layer that can do something with what the data implies. A perfect dataset that nobody acts on is a museum exhibit. A modest dataset that feeds a tight act-observe-update loop is a learning system, and learning systems compound. The compounding is the moat. The data is the substrate the compounding runs on.

Anatomy of a closed loop

A CLOSED LOOP HAS four parts and they have to all exist or the loop is broken. *An action* — the firm does something in the world or to itself. *An observation* — the firm captures what happened as a consequence of the action, in time, in detail, and at the right level of granularity to learn from it. *A reconciliation* — the firm compares what actually happened to what it had predicted or intended. *An update* — the firm changes the policy, the model, or the threshold that produced the original action so that the next action of the same kind is better.

The legacy enterprise has all four of these in pieces, but rarely connected. Sales takes actions; sales operations observes results; the analytics team

reconciles them quarterly; the policy update happens, if at all, through a rewritten playbook the next year. The latency between action and update is measured in months. The fidelity of the reconciliation is degraded by a dozen handoffs. The update is almost never traced back to the action that motivated it. It is, in any technical sense, an open loop with a long delay and a great deal of social ceremony in the middle.

The agentic firm closes the loop by collapsing the four steps into a single mechanised pipeline. The action is taken by an agent (or by a human assisted by one). The observation is captured as part of the same instrumented system that took the action. The reconciliation happens automatically and continuously — not at the end of a quarter, but at the end of each decision cycle. The update happens as a versioned change to the policy or to the model that fires the next action. Crucially, the loop closes *fast*. A loop that closes in a day learns about a hundred times faster than a loop that closes in a quarter, and the leading firms in any agentic vertical will be running loops that close in hours. The compound advantage of closing your loop ten times faster than the field for two years is an insurmountable lead in the same way that the compound interest of a 30 percent return for thirty years is an insurmountable fortune. The arithmetic is unfair. It is also accurate.

The three levels of data work

It is useful to distinguish three levels at which "data work" happens inside an enterprise, because they require different disciplines and most teams confuse them.

The *infrastructure level* is the boring, expensive, necessary work of getting the firm's data into a state where it can be queried at all. Pipelines, schemas, masters, lineage, governance, retention, access controls. This is the level at which the data engineering profession spent the 2010s, and it is the level at which most enterprises are still spending money in 2026 with limited apparent return. The work is not wrong. It is necessary but insufficient.

The *analytic level* is what most people mean by "the data team": dashboards, reports, ad-hoc queries, statistical investigations of hypotheses

raised by humans. This level has been the centre of gravity of data work since the early 2000s and has been quietly declining in marginal value for the last five years. Not because the work is bad but because the bottleneck has moved. Producing one more dashboard is no longer the binding constraint on a firm's intelligence. Acting on the dashboards already produced is.

The *agentic level* is new. It is the work of feeding data not into human eyes but into automated decision systems that take action. This level has different requirements from the analytic level. It needs lower latency, higher reliability, tighter coupling to the action layer, and a much more rigorous treatment of correctness — because the consumer of the data is not a human who can squint at a chart and notice the outlier, but an agent that will act on the outlier before anyone has had a chance to squint.

Most data teams in 2026 are still organised, staffed, and rewarded around the analytic level while their employer is starting to need the agentic level. This is one of the central organisational problems of the next three years and it deserves its own chapter, which I do not have room for. The short version: if your data leadership thinks of itself as the steward of dashboards, you are about to discover that the dashboards have stopped mattering.

Data ownership and the contracts that nobody read

There is a quieter front in the data war, and it is being fought in the procurement contracts of the SaaS vendors that hold most of a typical enterprise's operational data hostage. These contracts were signed in an era when nobody seriously believed they would want to do AI work on the data. They contain clauses that look innocuous and turn out to be expensive: restrictions on bulk export, restrictions on derivative use, restrictions on training models on the content, mandatory use of the vendor's own AI features, and — most maddening of all — clauses that give the vendor the right to use your data to train models that they then sell back to you and to your competitors.

The agentic firm cannot afford to be a tenant in its own data. Re-reading these contracts, renegotiating them, and where necessary moving off them is

one of the least glamorous and most consequential strategic projects of the next two years for any enterprise that is serious about agency. The CIOs who lead this effort will not get press releases for it. Their successors will be quietly grateful.

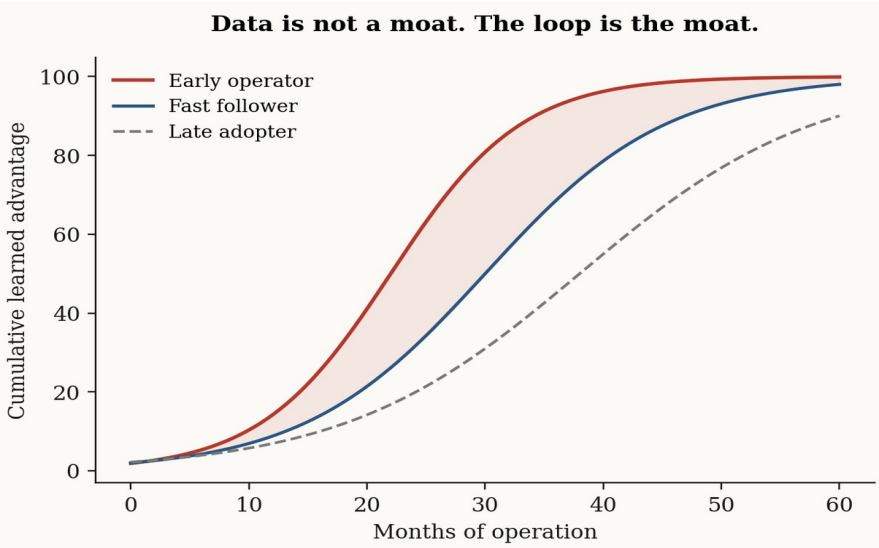
What the war is actually about

IF DATA IS NOT the moat and the loop is, then what is the war really about? It is about three things that the data-as-moat language obscured.

The first is *the right to learn from your own operations*. This is increasingly contested by SaaS vendors and increasingly threatened by data partnerships that look attractive in the short term and corrosive in the long term. Defending it is a corporate hygiene project, not a technology project.

The second is *the speed at which your loops close*. This is the technical and organisational core of the agentic firm and the place where the next decade's competitive differentiation will live. Closing loops faster requires the substrate work of chapter five, the routing discipline of chapter four, and the willingness to act on imperfect information that most legacy organisations have spent thirty years training out of themselves.

The third is *the discipline to forget*. This is the part of the data conversation that nobody likes. Data accumulates. Old policies, old models, old assumptions about the customer base, old constraints that no longer apply. A firm whose loops are closed but whose memory is full of obsolete patterns will learn very fast in the wrong direction. The discipline of expiring training data, retiring stale rules, and deliberately forgetting last year's customer profile is part of running a healthy agentic system. It has no analogue in the data warehouse era. It is one of the genuinely new managerial competencies of the agentic firm, and it requires people who can argue, calmly, that less memory will make the firm smarter.



Three firms with the same models and the same access. The only difference is when they began closing
the loop between action and observation. The shaded region is uncatchable.

The leader is not the firm with the most data. It is the firm whose Tuesday is shorter than the field's quarter.

VII. Orchestration, Memory, and Control

Intelligence without coordination is merely expensive improvisation.

A SINGLE AGENT IS useful. A society of agents is dangerous unless somebody writes the constitution. This is the part of the agentic decade that the demo culture of 2025 was least prepared for, and the part on which the most expensive failures of the next three years will be built. The single-agent demo is seductive because it puts a clever model in front of a clever interface and lets a clever person drive. Nothing in that experience prepares the viewer for what happens when ten agents must coordinate over a multi-step business process, with conflicting constraints, partial visibility into each other's state, asynchronous timing, and the standing possibility that any one of them is wrong about something the others are about to act on.

The mistake to avoid, in this transition, is the mistake of analogy. Most people, when they first encounter the orchestration problem, reach for the analogy of a manager and a team of subordinates. They start designing systems that look like miniature org charts: a "supervisor agent" giving instructions to "worker agents", with "review agents" auditing the output. This works for about three weeks of demo, and then it stops working for the same reason that command-and-control management stopped working in human organisations decades ago: it does not scale, it produces brittle hierarchies, and it concentrates failure at the apex of the tree.

The right model is not the org chart. The right model is closer to the *protocol*. A protocol is a small, precise set of rules that allow independent actors to interact reliably without any one of them having to understand the whole system. The internet works because of protocols. Markets work because of protocols. Ant colonies, immune systems, and well-designed engineering teams all work because of protocols. The agentic firm needs the same. Designing the protocols — what we will call orchestration — is one of the few genuinely new disciplines of the next decade, and it deserves to be

understood as a discipline rather than as a feature of someone's vendor contract.

What orchestration must decide

Orchestration is the discipline of deciding, in advance and in detail, six things. *How work is decomposed* — which parts of a complex objective become discrete sub-tasks for individual agents. *How context is shared* — what each agent knows, what it does not know, what it is allowed to assume about the others. *What memory persists* — what each agent remembers from previous sessions, what is shared across the system, what is intentionally forgotten. *Which tools are available to whom* — the action layer must be permissioned at the granularity of the agent and the case, not at the granularity of "the AI system". *When an action requires approval* — the threshold above which a human is in the loop, and the path by which the human is found, briefed, and waited on. *How failures are retried, escalated, or rolled back* — the discipline that turns a system that fails into a system that recovers.

These six are not technical details. They are the constitution. They determine whether the agentic system, in operation, behaves like a state with the rule of law or like a warlord economy in which whichever agent shouts loudest gets to write to the database. Most early enterprise agent deployments are warlord economies, and they have not yet failed in public only because the stakes have been deliberately kept small.

The decomposition problem

The hardest of the six is decomposition. Given a business objective — *retain this customer*, *close the books*, *resolve this incident*, *ship this feature* — how should it be broken into discrete tasks that individual agents can execute? Get the decomposition wrong and the rest of the system has no chance.

There are three failure modes to avoid. *Over-decomposition*: chopping the work into so many micro-tasks that the orchestration overhead exceeds the

cognitive content of the tasks themselves. The system spends most of its time passing messages between agents and almost none of its time doing the work. This is the sin of engineering teams that fall in love with multi-agent diagrams. *Under-decomposition*: leaving the work in such large chunks that no individual agent can hold the context required to do its part well, leading to hallucinated assumptions and brittle outputs. This is the sin of teams that try to solve everything with a single super-agent and a long prompt. *Mis-decomposition*: cutting the work along boundaries that do not match its natural seams, so that critical context has to be reconstructed across agent boundaries that should not have existed in the first place. This is the sin of teams that copy the org chart and call it a system.

The right decomposition is almost always the one that mirrors the natural decision boundaries of the work — the places where one kind of judgment ends and another begins. For a contract review, the natural seams are between extraction (what does the contract say), comparison (how does it compare to our policy), risk assessment (what could go wrong), and negotiation strategy (what should we ask for instead). For an incident response, they are between detection, triage, root-cause analysis, mitigation, and post-mortem. The seams reveal themselves to anyone who has actually done the work and are usually invisible to anyone who has only read about it. Decomposition is, therefore, a job for operators, not for engineers. The engineers build the protocol; the operators draw the borders.

Memory as politics

Memory in a multi-agent system is a political problem disguised as a technical one. The political question is *who is allowed to remember what about whom*. The technical question — vector store, graph store, episodic log — is the part the engineers want to talk about. The political part is the part that determines whether the system survives its first contact with the legal department.

A few principles that have to be made explicit early. Every memory write must have a *purpose* — a stated reason that this thing is being remembered,

which can be audited later. Every memory read must have an *authority* — a permission to access this information that is checked at the moment of access. Memory must have a *retention policy* that is enforced mechanically and not by hope. And the system must support *deliberate forgetting*, which is not the same as deletion: forgetting means the memory ceases to influence future agent behaviour, even if a copy is retained for compliance reasons. None of these are exotic requirements; all of them are routinely skipped by teams in a hurry to ship.

The deeper issue is that memory in an agentic system is *cumulative power*. An agent that has been operating in your environment for six months and has built up context about your customers, your contracts, your operations, and your idiosyncrasies is much harder to replace than one that started yesterday. This is good for capability and worrying for vendor lock-in. The firms that take memory architecture seriously will be the firms that own their own memory layer rather than renting it from the model provider. The firms that do not will discover, two years in, that their vendor's pricing power is now indexed to how much of their operational reality the vendor's system has absorbed.

Control without bureaucracy

Control is the word legacy organisations use when they want to apologise for the bureaucracy they are about to install. The agentic firm needs control — it needs it more than the legacy firm did, because the consequences of an uncontrolled action are larger and faster — but it cannot afford the bureaucracy. The trick is to invest in control mechanisms that do not require human attention to operate, and to reserve human attention for the cases where the mechanisms have detected an exception.

Three control mechanisms do most of the work. The first is the *confidence threshold*: the agent only acts on its own when its internal estimate of correctness exceeds a level that the firm has explicitly set and that can be tuned per workflow. Below that level, the agent escalates. This sounds simple and is the difference between a system you can trust and a system you cannot.

The second is the *blast radius cap*: every action the agent can take has a maximum scale, and the agent cannot exceed that scale without human intervention, even if every other check passes. The cap might be financial (no transaction above $X), reputational (no message to more than Y customers), or structural (no change to a record in this set of tables). Caps are crude. That is the point. Crude caps catch the failures that subtle controls miss.

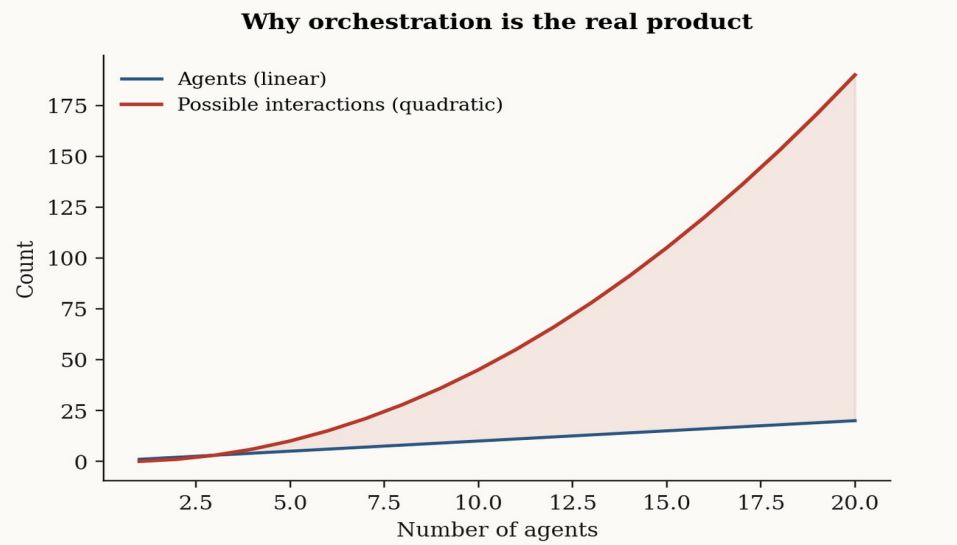
The third is the *reversibility window*: actions that can be undone are taken freely; actions that cannot be undone are taken slowly, with a pause that gives the firm a chance to notice. The pause might be a delay before the email is actually sent, a hold before the transaction is committed, a grace period before the configuration change propagates. Reversibility windows are the one thing that lets a firm move fast without occasionally catastrophically regretting it.

These three mechanisms are not glamorous. They will not appear in any vendor's marketing material. They are, in operational terms, the difference between an agentic firm that runs and an agentic firm that crashes. Build them first, before the cleverness.

The supervisor question

SOONER OR LATER EVERY multi-agent system raises the question: should there be a supervisor agent? An agent whose job is to coordinate the others, manage shared state, make routing decisions, and intervene when things go wrong. The answer is *yes, but not the supervisor you are imagining*. The supervisor in a well-designed agentic system is not a charismatic CEO-agent giving orders to a team of underlings. It is a thin, almost boring, deterministic process — closer to a Kubernetes scheduler than to a manager. Its job is to apply the protocol, enforce the constitution, route work to the right agent, and surface exceptions to humans. It does not do the work. It does not have opinions about the work. It does not generate novel content. It is, deliberately, the dullest part of the system, because every interesting thing it does will eventually be a thing the firm has to defend.

The seductive alternative — a powerful, model-based supervisor that "understands" the whole workflow and "decides" how to orchestrate — sounds elegant and works in demos. In production it is a single point of failure with a personality. When it is wrong it is wrong about everything at once, and it is wrong with the confidence of a senior model and the latency of a global rollout. Avoid it. Your supervisor should be the kind of code you can read on a Friday afternoon and understand fully. The intelligence should live in the workers.



Adding agents adds interactions faster than it adds capability. Every multi-agent system either
becomes a constitution or becomes a riot.

Orchestration is the constitution. Without one, you have not built a system. You have built a small, fast civil war.

VIII. Sales Without Sellers?

The buyer was always trying to talk to someone who knew the answer. The seller, very often, was in the way.

Sales is the function that the consulting class loves to romanticise and the operating class secretly resents. It is romanticised because of its mythology — the rainmaker, the closer, the relationship that nobody else can replicate. It is resented because the gap between the mythology and the daily reality is so large that almost no one inside a sales organisation believes in the mythology anymore. The daily reality of most sellers in 2026 is a calendar full of internal meetings, a CRM full of stale contacts, a pipeline they do not really believe, and a quota they will hit by closing four deals out of three hundred opportunities, almost none of which they personally generated. The discipline has become, against the will of the people inside it, mostly logistics with a sales jacket on top.

This chapter is not about replacing salespeople with chatbots. That framing is what vendors sell to CFOs and what salespeople fear in the abstract; it is not what is actually happening. What is actually happening is that the *logistical layer* of selling — the parts that consume 70 percent of a seller's week and produce almost none of a seller's value — is being absorbed by agentic systems, which is forcing a long-overdue clarification of what the remaining 30 percent of the work is actually for.

What the seller actually does

If you sit with a typical enterprise sales rep for a week and write down what they do hour by hour, you will produce a list that looks roughly like this. Research a list of accounts. Find people inside those accounts who match a buyer profile. Look up their backgrounds, their employers, their public statements. Draft outreach messages tailored to each. Send the messages. Follow up on the messages. Update the CRM after each interaction. Schedule meetings. Prepare for meetings by re-reading whatever the previous meeting

touched on. Take notes during meetings. Write up the notes after meetings. Send follow-up emails referencing the notes. Coordinate internally with sales engineering, legal, finance, and customer success. Build proposal documents from templates. Coordinate redlines. Chase signatures. Update the forecast.

Now go through the list and mark the items that genuinely require a human's judgment, taste, or relational presence. A handful — perhaps the meetings themselves, perhaps the negotiation, perhaps the relationship-defining moments where the customer trusts the rep with something they would not type into a chat. Almost everything else is logistics. The seller's week is mostly preparation for the small number of interactions that actually require a seller. Most of the preparation is a tax paid for the privilege of having those interactions.

The agentic firm absorbs the tax. It does not absorb the interactions. The result, when done well, is that the seller's week gets shorter and the seller's deals get larger, because the seller is now spending the part of the day that used to be CRM updates on the part of the day that used to be neglected: thinking about which customers are most likely to convert into something durable, which conversations actually deserve the seller's presence, and which long-shot accounts are worth a personal call this week. The agentic firm does not have fewer sellers because it has eliminated selling. It has fewer sellers because most of what was called selling was actually administration, and administration is now performed by the system.

The collapse of the funnel as a unit of analysis

For thirty years, the canonical mental model of a sales organisation has been the funnel. Prospects in at the top, deals out at the bottom, conversion rates at each stage, math at the end. The funnel has been a useful fiction. Its usefulness is now ending. Two things are killing it.

The first is that the funnel was always a way of talking about *activity* rather than *outcomes*. It assumed that the seller's job was to maximise the throughput of leads through stages and that the leads themselves were a roughly homogeneous input. Neither assumption is true any more. With

agentic systems doing most of the qualification and most of the outreach, the bottleneck is no longer the seller's ability to process volume. It is the firm's ability to identify, very precisely, which accounts have a real reason to buy, and to construct a credible reason to talk to them at the moment that reason exists. The funnel was a metaphor for an era of indiscriminate volume. The era is ending.

The second is that the funnel hides the fact that most of the value of a sales organisation comes from a tiny number of relationships. The mathematics of enterprise sales are brutally Pareto: in most B2B businesses, the top 5 percent of accounts generate something like half of the long-term revenue, and the top 1 percent of accounts often justify the existence of the entire sales organisation by themselves. The funnel treats these as just larger items in the same flow. In reality they are a different kind of object that requires a different kind of attention, a different kind of person, and a different kind of measurement. The agentic firm makes this distinction explicit: there is one machine for the volume tier and a small number of unusually senior humans for the relationship tier, and the two operate on different time horizons, different metrics, and different definitions of success.

What the agent actually does

In a well-built agentic sales system, the agent does roughly what a tireless and unusually well-briefed sales development rep would do, except faster and without the political need to look busy. It monitors a defined set of accounts for intent signals. It maintains a continuously updated picture of who at each account is in a position to buy and who is in a position to influence the buyer. It drafts personalised outreach grounded in actual context — recent product launches, organisational changes, hiring patterns, public statements — rather than the generic blast that gives the entire profession a bad name. It follows up on its own outreach with the persistence and specificity that human reps almost never sustain. It escalates to a human the moment a conversation begins to require judgment, taste, or commitment.

The hard part is the escalation. The agentic system has to know — and this is mostly a matter of careful design rather than model intelligence — when the conversation has stopped being qualification and has started being something a seller needs to be in. The wrong threshold burns the relationship by either escalating too late (the customer feels like they are being run through a script) or escalating too early (the human seller is dragged into conversations that the agent could have closed). Tuning the threshold is one of the operational disciplines of running an agentic sales system, and it requires honest measurement of what the agent did, what the seller did, and what the customer thought about each.

Two failure modes deserve specific warning. The first is what I will call *the velocity trap*: the agentic system makes outreach so cheap that the firm runs vastly more campaigns than it used to, drowning its own market in increasingly indistinguishable messages. Cheaper outreach is not better outreach. The discipline of restraint is harder when restraint costs the firm nothing. The second is what I will call *the scripted intimacy problem*: the agent becomes so good at producing the surface markers of a thoughtful human interaction — the references to recent posts, the mention of the mutual connection, the precisely-pitched compliment — that customers stop trusting any of those markers. The firms that overplay the personalisation card will discover, two years in, that personalisation has become a negative signal. Treat scarcity as a strategic input.

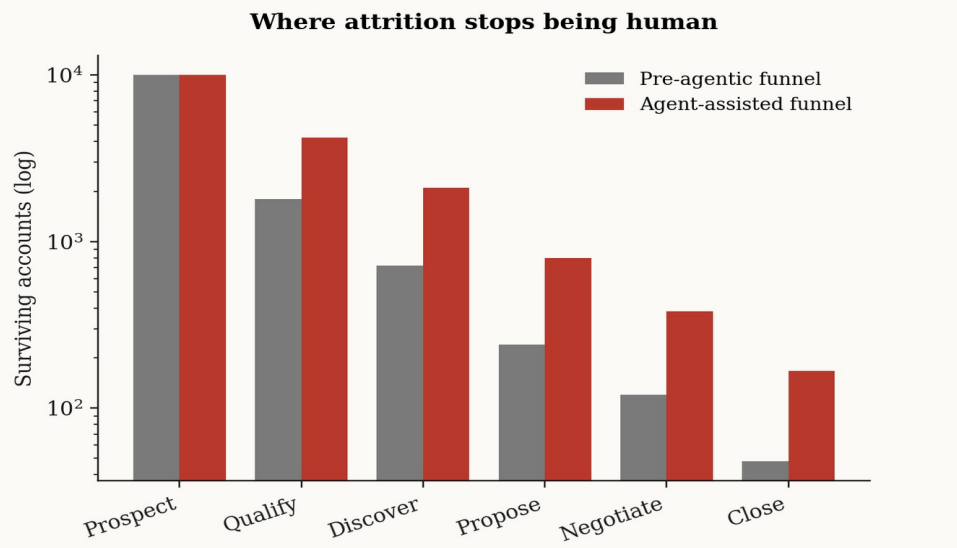
What the seller becomes

THE SELLER IN AN agentic firm is closer to what the best sellers always were: a senior commercial operator whose job is to build a small number of consequential relationships and to intervene in a small number of high-stakes moments where a human's word is the thing the deal turns on. The seller is not measured by the number of meetings booked, the number of emails sent, or the number of opportunities created. The seller is measured by the quality and durability of the business they are responsible for. This sounds obvious. It is not how 90 percent of sales organisations measured anything in 2024.

The seller's tools change accordingly. The CRM stops being a punishment and becomes a briefing system, populated by agents in real time with what the seller needs to know before walking into a meeting. The dashboard stops being a forecasting confessional and becomes a portfolio view of the relationships the seller is responsible for, with the agentic system's running assessment of each. The compensation structure stops rewarding volume and starts rewarding the slow, hard, judgment-dependent work of building accounts that compound. None of this is easy to roll out. All of it threatens the existing order of the sales organisation. Most of the resistance to agentic sales comes not from the customers, who notice almost nothing, but from the middle layers of the sales organisation, whose authority depended on running the volume the agents are now running.

The CRO question

A FINAL NOTE FOR chief revenue officers. The CRO who survives the next five years is the one who can answer, without flinching, a simple question: *which parts of your revenue organisation are buying logistics, and which parts are buying judgment?* If you cannot tell the difference for your own team, you will not be able to redesign it, because you will not know which roles to reduce, which to elevate, and which to reinvent. The CROs who confuse the two will end up either firing the wrong people (the patient relationship-builders whose value will become visible only after they are gone) or protecting the wrong people (the charming activity-generators whose contribution disappears the moment the activity is automated). The honest map of your organisation is the difference between leading the transition and being its casualty.



Most funnel attrition is logistical, not commercial. Buyers do not vanish because they hate the product.
They vanish because nobody followed up on Tuesday.

The end of the funnel is not the end of selling. It is the end of pretending that motion was the work.

IX. Marketing After the Content Factory

When making one more thing costs nothing, the scarce input is the willingness not to make it.

Marketing was the first function to discover, in 2023, that generative models could produce its outputs faster than its humans could brief them. It is therefore the function in which the most enthusiasm, the most disappointment, and the most quietly catastrophic mistakes have already accumulated. In the optimistic version of the story, marketing was about to become superhumanly productive. In the actual story, marketing has become superhumanly *prolific*, which is not the same thing. The production of marketing artefacts — emails, posts, landing pages, ads, scripts, decks, microsites — has gone up by an order of magnitude. The amount of attention available to receive them has not. The marketing function in 2026 is in roughly the position of a country that has just discovered how to print its own currency: the printing is impressive, the consequences for the value of the currency are not yet fully internalised.

The interesting question is therefore not how to produce more marketing. It is how to do less marketing better, in a world where everyone else is producing more. The answer requires unwinding two decades of accumulated assumptions about what marketing is for, how it is measured, and what counts as a productive marketer. None of this will be welcome to the people who built their careers inside the assumptions. It is, however, the only honest path through.

The collapse of the per-unit assumption

For most of the modern marketing era, marketing budgets were structured around an implicit per-unit cost. A piece of content cost something to produce; a campaign cost something to run; an ad placement cost something to occupy. The budget was, in effect, a portfolio of bets at known unit prices, and the discipline of marketing was the discipline of allocating capital across those

bets to maximise some objective. CMOs talked in CPM, CPL, CPA, CAC, LTV, and the math at the end was supposed to be honest. It often was, within the constraints of the era.

The per-unit assumption is now collapsing. Producing one more email, one more landing page, one more ad variant costs essentially nothing in agent-hours, and the cost is asymptoting toward the cost of the model call itself, which is asymptoting toward zero. This sounds like an obvious win for marketing budgets, and in the short term it is. In the longer term it is something stranger: the entire portfolio logic of the marketing budget has to be rewritten, because the things you are spending money on are no longer the things that determine the outcome. The bottleneck has moved from production to discrimination — from making the thing to deciding whether the thing should exist.

A useful test for any marketing leader in 2026: take your last quarter's outputs and ask, for each, *would I have produced this if it had cost me an hour of personal attention?* The honest answer for most of the volume is no. Most of it was produced because the production was cheap. Cheap production is not free production. Each artefact consumes a slice of the brand's credibility with the audience. Each indistinguishable email further trains the recipient to treat the channel as ignorable. Each generic post lowers the marginal value of any post on that channel for the entire industry. The cost was real. It was just paid by the brand's reputation rather than by the marketing budget.

What the agent does well, and what it does badly

Agents are excellent at producing fluent, on-brand, reasonably tailored marketing artefacts at scale. They are excellent at generating variants for testing. They are excellent at adjusting tone, format, and length for different audiences and channels. They are good, with the right context, at producing something that feels close enough to a senior marketer's voice that the recipient will not notice the substitution.

Agents are bad at three things, and the three things matter. They are bad at *selecting which thing to make* — at the strategic question of what the

audience needs to hear right now, given everything the brand has already said and everything the world is currently doing. They are bad at *taste*, in the sense of knowing when a piece of work has crossed the line from clever to embarrassing, or from on-brand to brand-erasing. And they are bad at *restraint* — at the instinct to send less, post less, ship less, in the service of being heard more.

The good marketing organisations of the next five years will be the ones that have built their human team around exactly these three things: selection, taste, and restraint. The bad marketing organisations will have built their team around the things the agents already do better, and will be wondering, eighteen months in, why their numbers look fine and their brand feels thinner.

The death of the content factory

THE CONTENT FACTORY WAS a mid-2010s organisational invention: a team of writers, designers, video producers, and project managers responsible for the steady production of marketing assets at industrial scale. It was a sane response to the demand for omnichannel content in an attention economy. It is also, in 2026, an obsolete form of organisation. Almost every individual role inside a content factory is now performed faster and at lower cost by an agentic system supervised by a much smaller human team. The content factory does not need to be reformed. It needs to be retired.

What replaces it is a smaller, more senior team whose function is editorial and strategic rather than productive. The editor decides what gets made and what does not. The strategist decides what the audience needs to hear and what is not worth saying. The designer makes the few things that need to be unmistakably ours, in a voice and style that the agent cannot easily counterfeit. The producer co-ordinates the small number of irreducibly human productions — events, conversations with executives, original research — that the audience cannot get from any agent. And nobody, in this team, is rewarded for volume. Volume is now free. Rewarding volume is rewarding a commodity.

The transition from content factory to editorial team is, mechanically, a layoff. There is no honest way to describe it otherwise, and the firms that try to

describe it as a "reorganisation" or a "shift in focus" will lose the trust of the people remaining without saving the dignity of the people leaving. The honest version is: this work is now done by machines, the people who did it are owed a real transition with real money, and the team that emerges on the other side will be smaller, more senior, and paid more per head. That is the deal. Any story that pretends otherwise is dishonest in a way that the remaining team will eventually feel.

Measurement, attribution, and the lie of the dashboard

Marketing measurement has been a polite fiction for most of its existence. Multi-touch attribution, marketing mix modelling, last-click, view-through, lift studies — each has been an honest attempt to answer a fundamentally difficult question, and each has been used dishonestly by people who needed a number to put in front of a CFO. The dishonesty was usually venial. Marketers needed to justify their budgets, the CFO needed a number, the number was produced, the budget was approved, everyone went home.

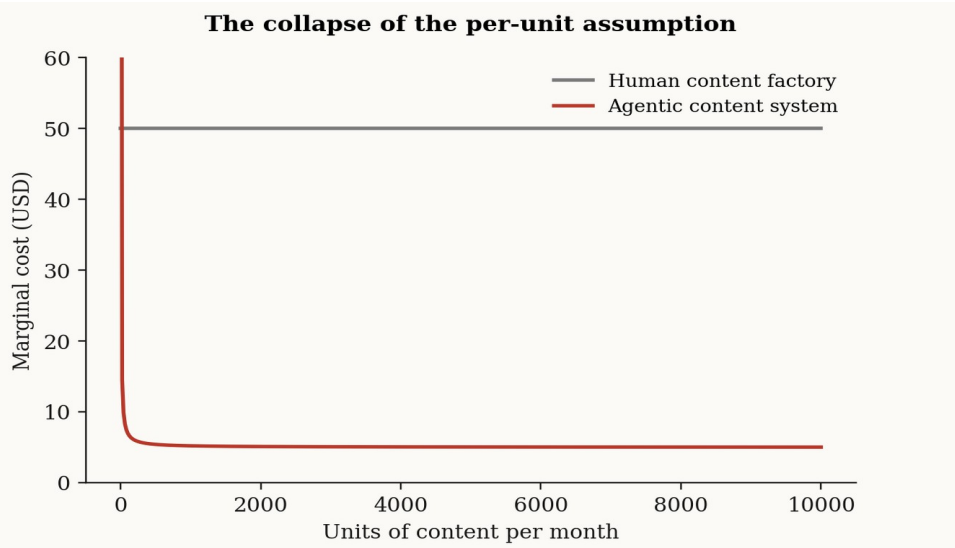
The agentic era makes this fiction harder to sustain, in two directions at once. On the one hand, the firm now has much more granular data about what happened after each marketing action, because the action and the response can both be instrumented at much finer resolution than before. On the other hand, the firm is producing so many marketing actions that the attribution problem becomes statistically intractable — every customer is now being touched by dozens of agentic outreaches, posts, retargeting variants, and personalisations, and assigning credit to any one of them is a fool's errand. The dashboards will get prettier and the underlying truth will get murkier.

The honest answer is to stop pretending that marketing is a discipline of attribution and start treating it as a discipline of *experimentation*. Run small, deliberate, well-controlled tests on things that matter. Be willing to ship and kill quickly. Reserve the strategic conversation for outcomes that survived a real test, not for the fluctuations of a dashboard nobody fully understands. This is how the best growth teams have always operated. The agentic era simply makes it the only sane mode for everyone else.

Brand as the only durable asset

If content is free, distribution is contested, attribution is murky, and personalisation has become a negative signal — what is left? The answer, slightly old-fashioned and increasingly obvious, is *brand*. Brand is the one marketing asset that the agentic substrate cannot manufacture for you, because brand is the residue of a long history of consistent behaviour and consistent voice that the audience has come to recognise and trust. An agent can produce content in your brand voice. It cannot produce the brand voice itself. The brand voice was earned, slowly, by the small number of decisions a real human team made about what to be and what to refuse to be.

In an environment where everyone can produce a thousand on-brand artefacts a day, the only thing that distinguishes one brand from another is the underlying *position* — the small set of things the brand stands for, the smaller set of things it refuses to do, the even smaller set of public commitments it has made and visibly kept. The CMOs who survive this transition will be the ones who treat brand strategy as the actual job and content production as the commoditised utility it has become. The CMOs who continue to treat content production as the centre of marketing's identity will be running smaller teams every year and explaining, in increasingly anxious terms, why the numbers are not what they used to be.



Marketing budgets are still set as if content were a unit good. They are not. The interesting cost in a world of zero-marginal output is judgement, not production.

Anyone can write a thousand emails today. The interesting marketer is the one who decided not to send nine hundred and ninety-eight of them.

X. Finance as an Autonomous Control Tower

The company that sees truth sooner buys itself time. The company that sees it later pays for it in cash.

Finance has always been the conscience of the firm, but conscience that arrives at month-end is conscience too late. By the time the books are closed, the variance is explained, and the deck is presented to the board, the operational moment that produced the variance is already forty days in the past. The team that caused it has moved on. The customer that triggered it has either been retained or not. The decision that should have been made in week one of the quarter is now being made in week one of the next. Finance, in most large firms, is a delayed photography studio: it captures what happened, cleans the image, and distributes it after the moment has passed. The photographs are technically excellent. They are also useless for changing what is in the frame.

Agentic systems give finance the chance to become something the function has wanted to be for forty years and never quite managed: a continuous control tower that watches commitments, cash movements, margin signals, anomaly patterns, contract obligations, collections risk, and forecasting drift in something approaching real time. This is not automation of the close. It is the obsolescence of the close as the central artefact of the function. The teams that internalise this will lead their finance departments through one of the most profound transformations in the history of corporate accounting. The teams that do not will spend the next five years buying expensive automation tools that make the photographs slightly faster while leaving the underlying photography paradigm intact.

Two waves of value

The transition will arrive in two waves, and confusing the two is the most common mistake.

The *first wave* is the obvious one. Document-heavy, repetitive, audit-sensitive work — invoice processing, expense review, reconciliations, close checklists, budget variance commentary, board pack drafting, vendor master maintenance, intercompany matching — gets compressed. A well-built finance agent can inspect evidence more patiently than an overworked analyst and can do so at three in the morning without complaint. The savings are real. The error rate often goes down. The audit trail gets better. CFOs who run this play in 2026 and 2027 will look heroic for a quarter or two and will be congratulated by their boards. This is the easy part. This is also the part the vendors will sell hardest, because it is the part that fits cleanly into the existing finance architecture and does not threaten the existing finance organisation.

The *second wave* is the harder and more consequential one. It is the wave in which finance stops being a periodic accountability function and becomes a continuous decision support layer for the rest of the business. It is the wave in which the FP&A team stops producing monthly variance reports and starts producing real-time guidance. It is the wave in which the controller stops chasing the close and starts running an autonomous reconciliation process that closes itself, every day, with exceptions surfaced as they occur. It is the wave in which the treasurer stops running quarterly cash forecasts and starts operating a continuous liquidity engine. The first wave saves money. The second wave changes what finance is for. Most CFOs are planning for the first and will be unprepared for the second.

What a continuous close actually looks like

THE MONTHLY CLOSE IS one of those rituals that everyone in finance complains about and almost no one questions in public. It exists because, historically, the alternative — closing the books continuously — was technically infeasible at any but the simplest entities. Transaction volumes were too high, intercompany matching was too messy, manual journal entries were too frequent, and the systems were too disconnected. So the firm batched the work into a monthly heroic effort, paid the team in pizza and overtime, and called the result a close.

A well-architected agentic finance system makes the monthly close progressively meaningless. Transactions are reconciled as they occur. Intercompany matches happen continuously, with exceptions routed to humans only when the agent's confidence falls below a threshold. Accruals are calculated and updated daily on the basis of operational signals from upstream systems, rather than estimated at month-end on the basis of someone's memory. The close ceases to be an event and becomes a state: the books are *always* close to closed, with an explicitly tracked confidence interval, and the formal close at month-end is a small ceremonial act that confirms what is already substantially true.

This is not a technology project. It is an organisational project that uses technology. The technical pieces have existed in some form for years; what has been missing is the willingness to redesign the close ritual itself. The agentic substrate makes the redesign possible by absorbing the routine reconciliation work that used to require an army. The hard part is the political work of telling the team that has been celebrated for years for closing in eight days that the eight-day close is now embarrassing and the goal is closing in eight hours.

The control tower

THE CONTINUOUS CONTROL TOWER is the mental model finance leaders should adopt instead of the monthly close. Think of an air traffic controller. The job is not to write a monthly report on what aircraft did. The job is to watch a continuously updating picture, with anomalies surfaced in real time and authority to intervene before a small problem becomes a large one. The instrumentation is dense; the action surface is narrow but immediate; the human attention is reserved for the cases the system cannot handle.

A finance control tower, properly built, watches several things continuously. *Cash position* across all accounts and entities, with rolling forecasts updated as commitments and receipts move. *Margin signals* at the SKU, customer, channel, and contract level, with anomalies flagged as they emerge rather than after the close. *Collections risk*, with predictive models on

each receivable that update as customer behaviour changes and that escalate to a human when the model's confidence drops. *Covenant pressure*, with continuous monitoring of leverage, coverage, and other ratios against the covenants in the firm's debt agreements. *Contract obligations*, with a live picture of what the firm has committed to deliver and whether it is on track to deliver it. *Forecasting drift*, with continuous comparison between the rolling forecast and reality, and an explicit signal when the forecast has stopped being credible.

None of these are exotic. All of them already exist in pieces inside most finance functions, scattered across spreadsheets and systems and people's heads. The control tower is simply the discipline of bringing them into a single live view, instrumented continuously, with explicit thresholds for when each requires human attention. The technology is not the hard part. The hard part is choosing which signals matter, which thresholds are right, and which exceptions deserve to wake somebody up at two in the morning.

The treasury problem

Treasury is the part of finance most ready for agentic transformation and least often discussed in transformation programmes, because treasury is invisible to most of the firm until something goes wrong. A modern treasury function manages cash positioning across accounts and currencies, hedges currency and interest rate exposures, runs the firm's banking relationships, monitors counterparty risk, and forecasts liquidity. Most of this work is high-frequency, rule-governed, and consequence-laden — exactly the kind of work that benefits from continuous automated execution with human oversight at the policy level rather than the transaction level.

A well-built treasury control tower sweeps cash automatically, executes hedges within policy bands without human intervention, monitors counterparty exposures continuously, and forecasts liquidity in real time on the basis of operational commitments rather than weekly check-ins. The treasurer's job becomes setting the policy, monitoring the system, and intervening on the unusual cases. The treasurer's team gets smaller and more

senior. The cost of mistakes goes down because the system catches things humans would have missed; the cost of catastrophic mistakes goes up because when the system gets a policy wrong it gets it wrong at scale. This is the trade. It is worth making, but it is not worth pretending is a free lunch.

What the CFO becomes

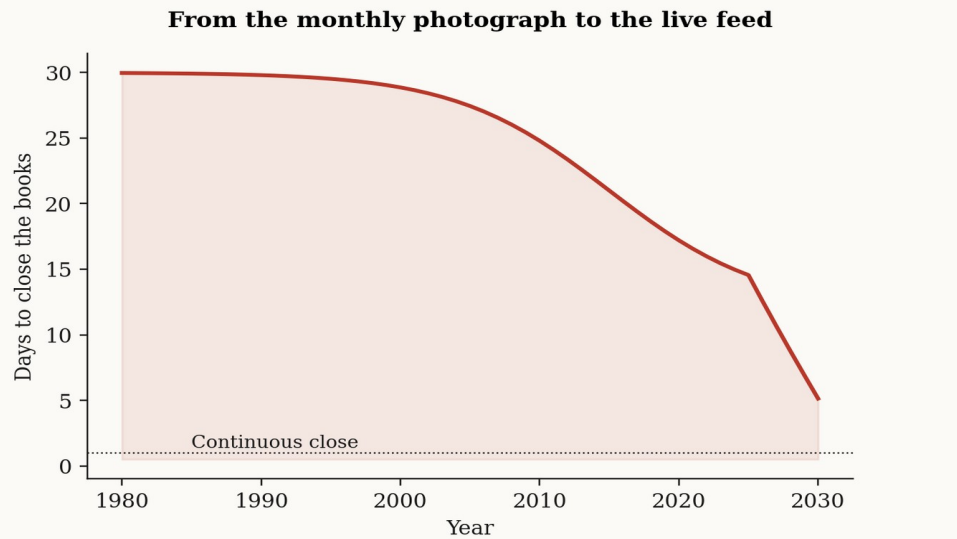
THE CFO OF THE agentic firm is closer in temperament to a chief of staff than to a chief accountant. The historical CFO was, fundamentally, the person responsible for ensuring that the firm's financial reality was accurately captured, properly reported, and credibly defended to investors and regulators. That work does not disappear, but it becomes a smaller share of the role, because it is the part of the work that the system performs continuously and without drama. What grows is the CFO's role as the firm's chief allocator of capital and the firm's chief interpreter of operational reality. The CFO who can tell, on Tuesday, that the European business is starting to slow because the agentic forecast has been quietly drifting downward for two weeks, and who can act on that signal before anyone else in the firm has noticed, is the CFO whose firm wins.

This is, frankly, a more demanding role than the historical CFO had. It requires a deeper engagement with the operating business, a willingness to act on imperfect signals, and a tolerance for being wrong publicly in the service of being right faster than the competition. The CFOs who came up through audit and accounting will find this transition uncomfortable. The CFOs who came up through operational finance will find it natural. The composition of the next generation of finance leadership is going to look noticeably different from the current one, and the transition will be painful for the people whose careers were built on the discipline that is becoming a commodity.

The auditor's dilemma

A FINAL NOTE FOR the audit profession, which is going to find itself in a particularly awkward position over the next five years. External audit was

built around the assumption that the firm's books were a periodic artefact produced by a team of humans, that the audit team's job was to inspect a sample of the underlying transactions and form an opinion on the artefact, and that the artefact and the opinion would be delivered to investors a few weeks after the period end. None of these assumptions survives the continuous close. If the books are continuously reconciled by an agentic system, the audit problem is no longer about sampling transactions; it is about auditing the *system* that reconciled them. This is a different discipline, requires different skills, and is going to be performed by a different mix of firms than the current Big Four oligopoly. The transition will be slow because regulation is slow. It will be unstoppable because the underlying economics are changing.



Finance was always a delayed photography studio. Continuous close turns the conscience of the firm into a real-time organ rather than a monthly tribunal.

The conscience of the firm is more useful when it speaks on Tuesday than when it speaks on the thirty-first.

XI. Legal, Risk, and Compliance by Machine Discipline

The legal department is one of the few places in the firm that is still paid by the page.

Of all the corporate functions facing transformation, legal is the one most actively defending the moat that no longer exists. The defence is intelligent and superficially convincing: legal work requires judgment, judgment requires expertise, expertise requires training, training requires time, and the work product is consequential enough that errors are catastrophic. Therefore — runs the argument — legal cannot be meaningfully automated. This is true of perhaps 5 percent of the work the legal department does. It is wildly false of the other 95 percent, which consists of repetitive, pattern-matching, document-comparison work that lawyers themselves complain about constantly and that has been outsourced to associates, paralegals, contract management vendors, and offshore document review firms for thirty years precisely because nobody at the top of the profession wanted to do it. The agentic transformation of legal is therefore not a threat to lawyers as such. It is a threat to the *organisational structure* that lawyers built to insulate themselves from the work they did not want to do.

This is the chapter most likely to make a general counsel angry. I am going to make the argument anyway, because the alternative is to flatter a profession whose privileges depend on a fiction the profession itself stopped believing in years ago.

The Pareto problem

A typical legal department's queue looks roughly like this. NDAs, lots of them, mostly identical. MSA renewals, mostly small variations on a template the firm signed last year. Data processing addenda, mostly responses to the same handful of customer requests. Order forms and statements of work, mostly populated from a CRM by a sales team that was supposed to follow the template and did not quite. Vendor contracts, mostly inbound paper that

needs to be redlined against the firm's standard positions. A small number of real estate, employment, and litigation matters. An even smaller number of M&A deals or genuinely novel commercial arrangements that require senior lawyers to actually think.

If you plot volume against required judgment, the picture is grotesque. The work that consumes the most pages takes the least judgment. The work that requires the most judgment takes almost no pages. Both flow through the same queue, supervised by the same humans, in the same order they arrive. The senior lawyer who should be spending 80 percent of her time on the M&A deal is spending 60 percent of it on NDAs, because the NDAs are what is on fire today. This is not a failure of the lawyer. It is a failure of how the work has been organised for thirty years.

The agentic firm sorts this out. The high-volume, low-judgment work moves to a substrate that can read, compare, redline, and route at machine speed. The substrate is supervised by a small number of mid-level lawyers whose job is to set the policies the substrate enforces and to inspect the cases that fall outside them. The senior lawyers stop touching NDAs and start spending the day on the work the firm actually pays for them to do — the strategic counsel, the unusual deals, the relationships with regulators, the cases where being wrong has consequences that no system can absorb. The firm gets faster legal work, lower legal cost, and a better-deployed senior bar. The lawyers who lose are the mid-tier ones whose value was being a more careful version of what the substrate now does. The lawyers who win are the ones whose value is judgment — and they end up doing more of the work they trained for.

What the agent reads, and what it does not

A MODERN LEGAL AGENT can do several things competently and several things badly. It is worth being precise about which is which, because the failures are usually in the gap between what the agent can do and what the firm assumed it could do.

The agent can extract structured information from a contract — parties, term, governing law, payment terms, termination rights, liability caps, indemnities, the standard playlist. It can compare a clause against a library of acceptable language and flag deviations. It can produce a redline against the firm's standard positions. It can summarise the negotiating posture of a counterparty across many previous deals. It can answer questions about the contents of a contract that a junior associate would have spent two hours billing for. It can do all of this faster, more cheaply, and often more consistently than the humans who used to do it.

The agent cannot, with any reliability, decide whether a particular deviation from the standard is *acceptable in this case for this counterparty given this strategic context*. It cannot read the room of a negotiation. It cannot tell that the customer is bluffing or that the regulator is angry or that the executive on the other side is about to lose her job. It cannot exercise the kind of judgment that comes from having watched a dozen similar deals go wrong in interesting ways. It cannot, fundamentally, take responsibility — and the legal profession is, at its best, a profession of taking responsibility for judgments that nobody else in the firm is qualified to take.

The discipline of building an agentic legal function is the discipline of being honest about which judgments belong on which side of this line. Mis-classify, and you either bottleneck the firm with too much human review (defeating the point) or you let the system commit the firm to obligations that it should not have committed to (creating consequences the firm will be paying for in two years). The general counsels who do this well will be the ones who treat *the sorting itself* as a senior legal function rather than a delegated administrative one.

Compliance as a system problem

Compliance is a parallel story with its own peculiarities. The compliance function exists because regulators require firms to demonstrate that they are following rules, that they have controls in place to enforce the rules, and that they can produce evidence of the controls when asked. In the legacy enterprise, this work has been done by a combination of policies (documents

that nobody reads), training (modules that everyone clicks through), controls (forms that get signed), and audits (samples that get checked). The whole apparatus is performative in the precise sense that its primary output is *evidence that it exists*, not its enforcement.

Agentic systems make a different kind of compliance possible — and, eventually, mandatory. Instead of policies that nobody reads, the firm has codified rules that the agentic substrate enforces continuously at the moment of the action being controlled. Instead of training, the firm has a system that prevents the action that the training was supposed to deter. Instead of audits, the firm has a continuous record of every controlled action with its justification, its authority, and its outcome. The compliance function stops being the team that produces evidence after the fact and becomes the team that designs the rules the system enforces.

This is good for compliance and bad for compliance officers, because the role is going to require a much more technical and operational kind of person than the role attracted in the past. The compliance officers who came up through audit and law will find the new shape of the work uncomfortable. The compliance officers who came up through operations and engineering will find it natural. The composition of the function is going to shift, and the firms that try to do agentic compliance with a legacy compliance team will produce the worst of both worlds: a system that runs at machine speed but is governed by people who cannot read what the system is doing.

Risk as a continuous discipline

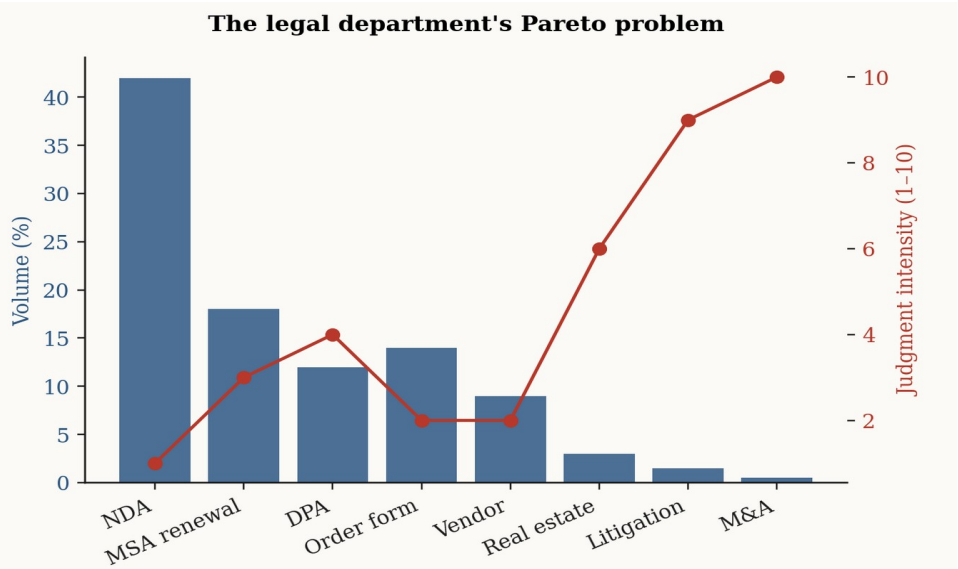
Enterprise risk management has always been an awkward function. It is asked to identify, quantify, monitor, and mitigate risks across every part of the firm, with no real authority over any of them, on a budget that is the first to be cut when things go well and the first to be blamed when things go badly. The historical workaround has been the risk register: a spreadsheet of identified risks, scored on probability and impact, reviewed quarterly, and largely ignored by the people running the businesses the risks belong to. The risk register is to risk management what the dashboard is to operational

management — an honest acknowledgement that nobody is going to do anything continuous about it.

The agentic firm replaces the risk register with continuous monitoring of the *signals* that historically preceded the risks. Concentration risk gets monitored by watching customer concentration in real time, not by recalculating it once a quarter. Supplier risk gets monitored by watching supplier behaviour in real time, not by sending an annual questionnaire that the supplier's intern fills out. Operational risk gets monitored by watching the operations themselves, not by interviewing the operators about what could go wrong. The risk function becomes a small, technical team that designs the signals and the thresholds, and the actual monitoring is done by the substrate. The CRO becomes someone with a credible real-time view of the firm's actual exposures rather than a curator of last year's predictions.

What none of this fixes

A FINAL, HONEST PARAGRAPH for the lawyers reading. None of this fixes the parts of legal work that are genuinely hard, ambiguous, and consequential. The hard cases are still hard. The strategic deals still require senior judgment. The bet-the-company moments still need a human in the room with the standing to make a call and the experience to know what calling it implies. What changes is the *organisational frame* around the hard work. The hard work is no longer surrounded by, and politically subordinated to, an ocean of low-judgment volume. The senior lawyer is no longer a partner who has to pretend to enjoy contract review in order to fund the part of the practice she actually wants to do. The firm gets legal counsel that is faster, cheaper, and — for the work that matters — better, because the senior bar finally has the time and attention to give it. This is good for the firm. It is good for the senior lawyers. It is bad for the middle of the profession, and the profession is going to spend the next decade pretending otherwise. Do not be fooled.



The work that takes the most pages takes the least judgement. The work that takes the most judgement takes almost no pages. Yet both are routed through the same queue.

The legal department's queue was a sorting failure pretending to be a workload. The substrate sorts. The lawyers, finally, get to lawyer.

XII. Product and Engineering as Self-Improving Systems

Software was always supposed to learn. We finally built a substrate that lets it.

For sixty years, the discipline of building software has been organised around a particular asymmetry: writing code is hard, expensive, and error-prone, and therefore the firm should employ a small number of carefully selected humans to do the writing while everyone else uses the result. The asymmetry produced everything we now take for granted about engineering organisations — the hierarchy of titles, the rituals of code review, the cult of the architect, the elaborate apparatus of agile process, the perpetual debate about whether a feature is worth the engineering cost. All of it depended on the assumption that engineering capacity was the binding constraint. For most of the history of computing, it was.

That assumption is no longer reliable, and what comes next is more interesting than the headlines suggest. The interesting story is not that AI writes code (it does, often badly, occasionally brilliantly). The interesting story is that the *cycle* between idea, implementation, deployment, observation, and revision is collapsing in time, and a collapsing cycle changes the discipline in ways that the autocomplete-versus-human framing entirely misses. When the gap between *I wonder if* and *we shipped it* drops from quarters to days, the things you wonder about and the things you ship become categorically different. The product organisation that internalises this will look back at its 2024 self the way an industrial designer in 1990 looked back at the era before CAD: the old workflow was not slow because anyone was bad at their job; it was slow because the substrate could not move at the speed of thought.

What the engineer actually does

It helps, before going further, to be precise about what an engineer's day actually contains. If you sit with a senior software engineer for a week and write down what she does, the list looks roughly like this. Read existing code

to understand a system that someone else built. Try to reproduce a bug that a customer reported. Investigate why a test is failing. Read a design document. Write a design document. Argue about a design document in a meeting. Draft a small change to a piece of code. Wait for the test suite. Wait for the build. Read a code review. Respond to a code review. Approve someone else's code review. Investigate why the deployment broke. Roll something back. Roll something forward. Investigate a metric that moved unexpectedly. Write a runbook. Update a runbook nobody reads. Onboard a new engineer. Sit in a planning meeting. Sit in a retrospective. Write code, occasionally, in the gaps.

Now go through the list and ask, for each item, whether the value comes from the engineer's *judgment* or from the engineer's *patient execution of a known procedure*. Most of the items are mostly the second. The engineer's actual judgment — the part nobody but a human can perform — is concentrated in a small number of moments per week: the architectural decision, the choice of which bug actually matters, the read of why a system is behaving the way it is, the call about what to ship and what to defer. The rest is patient execution of procedures the engineer would be happy to delegate to anyone, or anything, that could be trusted to do them right.

The agentic substrate is exactly that delegate. It can read the codebase faster than the engineer. It can reproduce the bug faster, often. It can draft the small change, run the test suite, respond to half the comments in the review, and surface the engineer's attention only when something requires her to actually decide something. The engineer's week gets shorter and the engineer's contribution gets denser. The engineering organisation gets faster, smaller, and more senior — not because the firm has fired junior engineers, but because the work that used to require a junior engineer is now performed by the substrate, and the firm has stopped hiring three people to grow into one.

The collapse of cycle time

THE MOST UNDERRATED METRIC in modern engineering is the time from an idea being credible to that idea being in front of users. Call it cycle time. Cycle

time has been falling for decades — version control, continuous integration, continuous deployment, feature flags, observability — and each fall has produced an ecosystem of practices, books, conferences, and consultancies that tried to convince the laggards to catch up. The DevOps movement was, in essence, the sociological apparatus around getting industrial-era engineering teams to accept that cycle time mattered.

Agentic systems compress cycle time again, by an amount that the previous compressions did not match. The thing that used to take two weeks — read the bug, find the cause, draft the fix, write the tests, get the review, deploy — now takes an afternoon, when done well, with a human in the loop on the parts that require judgment and the substrate doing the rest. The thing that used to take a quarter — design a feature, build it, integrate it, test it, ship it, observe it — now takes a few weeks. And the thing that used to take a year — rebuild a major subsystem, migrate the data, retire the old version — now takes a few months, because the agentic substrate can perform the kind of mechanical refactoring work that used to require a whole team of engineers to grind through.

The collapse of cycle time is not a productivity story. It is an *epistemic* story. When cycle time is short, the firm can afford to be wrong more often, because being wrong is no longer expensive. The discipline of betting carefully on a small number of expensive features gets replaced by the discipline of running many small bets fast, observing the results, and killing the bets that did not work. This is, in principle, what agile methodology was supposed to deliver. In practice, agile methodology delivered the rituals of fast iteration on top of an infrastructure that was still slow, which produced a great deal of motion and not much speed. The agentic substrate makes the infrastructure as fast as the rituals always pretended it was.

The engineer becomes a curator

In the legacy engineering organisation, the engineer was a producer of code. In the agentic engineering organisation, the engineer is a *curator* of code that the substrate produces. This is not a demotion. It is a different kind

of work, and in many ways a harder one, because the curator has to read more, decide more, and take responsibility for systems she did not personally write. The curator's skills are taste, system thinking, the ability to spot the subtle wrongness in a piece of code that compiles fine and passes its tests, and the willingness to throw away work that the substrate produced fluently but incorrectly.

The hardest part of being a curator is *resisting fluent wrongness*. The agentic substrate is, by construction, very good at producing code that looks right. It is less good at producing code that *is* right, especially in subtle ways that only become apparent at the boundary between systems or under unusual load. The curator's job is to read the code with the assumption that the substrate may have hallucinated, may have copied an obsolete pattern, may have introduced a security flaw the substrate did not understand, may have gotten the concurrency model subtly wrong. This is harder than writing the code in the first place would have been, because writing is constrained by the writer's understanding while reading is not. The curators who do this well will be the senior engineers of the next decade. The curators who do it badly will produce systems that are fast to ship and slow to debug, and their employers will eventually fire them.

Product as a different discipline

THE PRODUCT MANAGER HAS had a curious relationship with engineering for the last fifteen years. She has been simultaneously the person who decides what gets built, the person who has to beg for the engineering capacity to build it, and the person who is blamed when the thing that gets built is not what the user wanted. The role has been popular, well-paid, and quietly miserable for almost everyone who has held it.

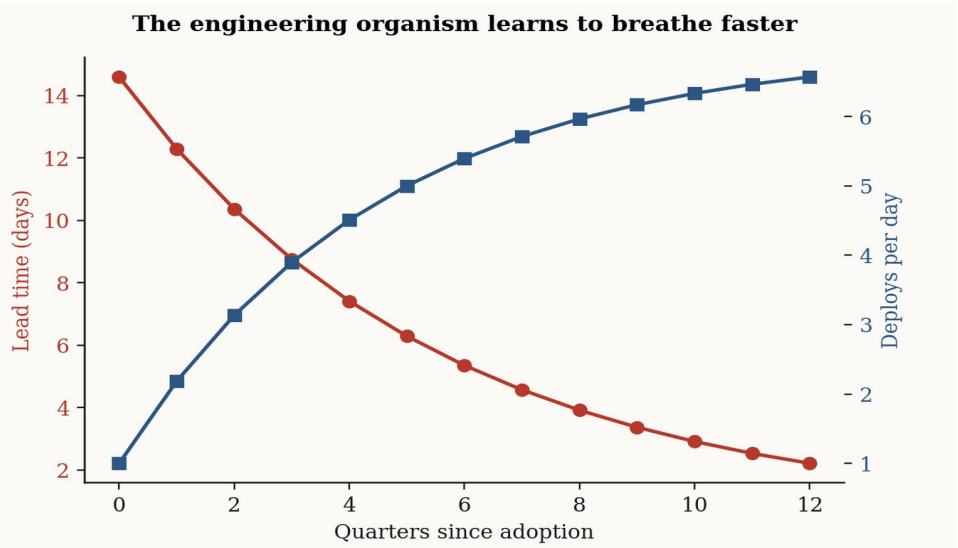
Agentic systems change the relationship. When cycle time collapses, the bottleneck stops being engineering capacity and starts being *good ideas about what to build*. The product manager who can generate, prioritise, and validate ideas faster than the substrate can build them becomes the most valuable person in the room. The product manager who is mostly a

coordinator of engineering capacity discovers that the coordination work has been absorbed by the substrate and her remaining contribution is harder to defend. The product function bifurcates: a smaller number of senior, opinionated, taste-driven product leaders rise in importance, and the larger middle layer of product coordinators thins out the same way the middle layer of every other function is thinning out. The leaders who survive this transition are the ones who can answer, every week, with conviction and evidence, *what should we build next, and why is it more important than the other twenty things we could build*. That has always been the real question. Most product organisations have been answering it badly for years, hidden behind the alibi that the engineering team could not have built the answer fast enough anyway. The alibi is gone.

What good engineering organisations look like in five years

The best engineering organisations of 2030 will look smaller than their 2025 ancestors and will produce more software. They will have fewer junior engineers and more senior ones. They will spend less time in planning meetings and more time looking at what the substrate just built. They will ship code multiple times a day, sometimes multiple times an hour, because the cost of shipping has dropped to the point where the question is no longer *is this worth a release* but *is there any reason to wait*. They will have invested heavily in observability, because a fast-moving system that nobody is watching is a system that fails fast and silently. They will have invested heavily in evaluation infrastructure for the agents themselves, because agents that write code without supervision eventually write the wrong code. And they will have absorbed, painfully, the cultural shift that comes with treating code as a substrate the firm tends rather than an artefact the firm produces.

The worst engineering organisations of 2030 will look like the best engineering organisations of 2020, only larger.



When code, tests, and review become collaborative loops with agents, the DORA curves do not improve linearly. They invert: lead time falls, frequency rises, and the team's identity quietly changes.

The engineer used to write the code. Now she reads it, with the eye of someone who knows the writer never sleeps and never quite tells the truth.

XIII. Operations and Supply Chains That Think

The bullwhip is not a fact of nature. It is a fact of how slowly information travels between tiers.

In 1961, Jay Forrester sat down at MIT and wrote the founding text of system dynamics, *Industrial Dynamics*. The book contained, among other things, a clean mathematical demonstration of a phenomenon every operations manager had felt without being able to name: small fluctuations in end-customer demand produced ever-larger fluctuations in orders as you moved upstream through the supply chain. A 5 percent wobble at the retailer became a 20 percent wobble at the distributor and a 60 percent wobble at the factory. Forrester showed that this was not a failure of forecasting or a failure of management. It was a structural consequence of *information delay* between tiers. Each tier reacted to the previous tier's order signal, which had already been distorted by the tier in front of it, and the distortions compounded. Forrester named the effect; later researchers called it the bullwhip. Sixty years of supply chain optimisation has been a series of attempts to dampen it.

The reason most of those attempts have been disappointing is that they treated the bullwhip as a forecasting problem. It is not. It is a *latency* problem, and forecasting is what you reach for when you have given up on removing the latency. Agentic systems do not improve forecasts. They remove the conditions that produced the bullwhip in the first place, by collapsing the time and the friction between *something happened in the world* and *every relevant party knows about it and has updated their plans*. This is the central insight of agentic operations, and most of the rest of this chapter is consequences.

Operations, but legible

The operations function in a typical enterprise is the part of the business that has been digitised the longest and yet remains, on inspection, the least *legible*. Manufacturing execution systems have been logging things since the 1980s. Warehouse management systems have been tracking inventory since

the 1990s. Transportation management systems have been routing trucks since the 2000s. ERP modules have been planning capacity since the 1970s. There is no shortage of data. There is, however, an enduring scarcity of *coherent live answers* to questions that operations leaders need to ask every day: where is the constraint right now, what is it costing us per hour, what is the highest-leverage thing we could do about it in the next four hours, and what will the situation look like by the end of the shift if we do nothing.

Asking these questions today, in most large enterprises, requires a meeting. The meeting requires preparation. The preparation requires pulling data from several systems, reconciling it manually, and producing a deck. By the time the meeting happens, the situation has changed. By the time a decision is made, the shift is half over. By the time the decision is implemented, the next shift is starting. The operations function spends most of its energy reacting to a version of the world that is several hours stale, and the firm pays for the staleness in the form of inventory it did not need, expedited shipments it should not have made, downtime it could have prevented, and customer commitments it cannot honour.

The agentic operations function does not run meetings about the state of the floor. It runs a continuous, instrumented, model-mediated picture of the floor, with the agentic substrate watching for the patterns that historically preceded problems and surfacing the patterns to human operators before the problems become incidents. The shift supervisor stops being a person who reacts to incidents and becomes a person who is occasionally interrupted by the substrate with a suggestion, a question, or an alert. The plant manager stops being a person who reads yesterday's report and becomes a person who reads the substrate's running interpretation of today.

The supply chain learns to talk to itself

A SUPPLY CHAIN IS, fundamentally, a chain of organisations that are trying to coordinate without trusting each other very much. Each tier holds inventory partly as a buffer against real demand uncertainty and partly as a buffer against the unreliability of its upstream and downstream partners. The buffers

are expensive. They are also rational, given the latency and opacity of the information flow between tiers. Decades of supply chain transformation programmes have tried to reduce the buffers by improving the information flow — EDI, vendor-managed inventory, collaborative planning forecasting and replenishment, and various enterprise integration platforms — and have produced incremental improvements that nobody is satisfied with.

The agentic substrate makes a more radical move possible: continuous, machine-mediated reconciliation between tiers, in which each tier's agents talk to each tier's agents directly, share the relevant pieces of state in the relevant moments, and update their plans together rather than serially. This sounds utopian. It is technically feasible today and is being done by a small number of firms whose names you do not yet know. The economic prize for getting this right is enormous, because the buffers are enormous, and the buffers are mostly there to compensate for an information latency the substrate eliminates.

There is a political problem to be honest about. The firms in a supply chain are not, in the traditional sense, eager to share information with each other. The retailer does not want to tell the supplier exactly how much inventory it is holding, because the inventory level is a negotiating lever. The supplier does not want to tell the manufacturer exactly when it expects to ship, because the shipping window is a negotiating lever. The manufacturer does not want to tell the raw material supplier exactly how much it is producing, because that is a negotiating lever too. Every link in the chain has trained itself, over decades, to treat opacity as a defensive asset. The agentic substrate cannot, by itself, resolve this. It can only make the cost of opacity visible, in the form of buffers that can be quantified and assigned to specific pieces of unshared information. When the cost is visible, somebody renegotiates, eventually.

The reflex layer

Operations is the function in which the reflex / deliberation distinction from chapter four matters most. Most of what an operations function does should be reflex, executed by the substrate without consulting a human,

because the events are frequent, the right responses are well understood, and the cost of waiting for a human is greater than the cost of an occasional automated mistake. The reflex layer of an agentic operations function includes things like rerouting an order around a delayed shipment, escalating a quality alert when a sensor reading exceeds a threshold, releasing a hold on a backorder when stock arrives, adjusting a production schedule when a machine goes down, paging the right engineer when a control loop has gone unstable. None of these require a meeting. All of them are routinely performed in legacy operations by the slowest, most expensive, and most error-prone available mechanism: a human in a control room with a phone.

The deliberative layer of operations — the part that humans should still own — is the part that involves trade-offs across multiple objectives, irreversible commitments, or strategic shifts in how the operation runs. Whether to pull capacity from a customer order to serve a more strategic customer. Whether to absorb a quality variance or stop the line. Whether to renegotiate a supplier contract when the supplier has missed three deliveries in a row. These are the operations decisions that the experienced plant manager earned the right to make over a career, and they are exactly the decisions the agentic substrate cannot and should not make on its own. The art of running an agentic operation is freeing the experienced manager from the reflex work so that she can spend her days on the deliberative work.

Maintenance is the test

IF YOU WANT TO know whether a firm has actually built an agentic operations function or whether it has just bought a few tools and put a banner on the wall, look at maintenance. Predictive maintenance is the canonical agentic operations use case and has been promised, prototyped, piloted, and abandoned by more firms than any other application of machine learning in industry. The reason it is so often abandoned is that it requires the four properties of a real nervous system — continuous sensing, designed routes, codified reflexes, and adaptive memory — and most firms have built one or two of them and called the result a programme. The result is a system that detects

an impending failure, fires an alert into a queue nobody is watching, and is then blamed for being inaccurate when the failure happens anyway.

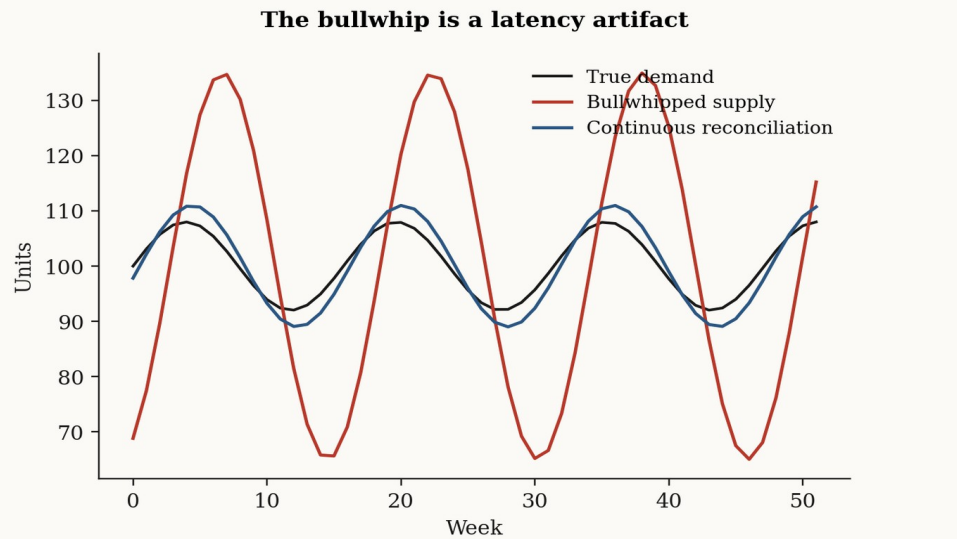
The firms that have made predictive maintenance work are the firms that took the time to design the *routes* — what happens when the system detects an anomaly, who is paged, with what context, with what authority to act, with what feedback loop afterward. The technology is the easy part. The discipline of writing down, in advance and in detail, the operational response to each class of detection is the part most firms skip. Maintenance is the test because it forces all four properties of the nervous system to be present at once. A firm that has built agentic maintenance has built the nerves of an operating organisation. A firm that has not has installed a more expensive smoke detector.

What replaces S&OP

Sales and Operations Planning is the legacy ritual that supply chain organisations use to reconcile demand forecasts with supply capacity, on a monthly or sometimes weekly cadence, in a series of cross-functional meetings that produce a consensus plan that everyone in the room agrees to publicly and quietly disregards in their own day-to-day. S&OP has been criticised for thirty years and survives because no one has known what to replace it with. The replacement, when it comes, will look like a continuous reconciliation engine that watches demand signals and supply capacity in real time, surfaces the gaps as they emerge, and routes the gaps to the human who can decide how to close them — without a monthly meeting, without a consensus deck, without the political theatre that has historically been the actual product of the S&OP process. The cadence ceases to be monthly because it ceases to be a cadence at all. It becomes a continuous state, watched continuously, intervened in occasionally, and reported on after the fact.

This is the hardest of the operations transitions to sell to a supply chain leadership team, because S&OP is not just a process — it is the *political settlement* between sales, operations, and finance, and the meetings are

where the settlement gets renegotiated each month. Replacing the meetings with a continuous engine threatens the settlement, and the people whose authority depends on the meetings will fight the replacement under whatever name they can find for the fight. The leaders who win this battle will be the ones who give the settlement somewhere new to live — typically in a smaller, more senior steering committee that meets less often but with much higher decision quality, on the basis of the substrate's continuous picture rather than a monthly deck.



Forrester's bullwhip is not a fact of nature. It is a fact of how slowly information moves between tiers. Continuous reconciliation does not damp it. It removes the conditions that produced it.

Forrester explained the bullwhip in 1961. We have been damping it ever since. The substrate removes the conditions that produced it.

XIV. Management Without Middle Management

Many managers were not created by destiny. They were created by friction.

The middle manager is the most defended and least examined role in the modern firm. Every transformation programme of the last twenty years has paid lip service to flattening the organisation, and almost every one of them has resulted in roughly the same number of middle managers, with slightly different titles, doing slightly different things. The reason is not that flattening is impossible. The reason is that the work middle managers do is real work — it just happens to be the kind of work that the previous generation of technology could not absorb, so it had to be performed by humans, and the humans accumulated authority along with the work. The defence of middle management has therefore always been, in effect, a defence of the *substrate* on which middle management ran. Take away the need for the substrate and the defence of the role gets harder to articulate, even by the people performing it.

This chapter is the most uncomfortable in the book and I am going to write it carefully, because the loose version of the argument has been used to justify a great deal of cruelty. The middle manager is not the villain. The middle manager is a person who answered an honest job posting, did her job well, and is now finding that the job posting described work the firm no longer needs. That is not her fault, and the firm owes her something better than a redundancy notice with a press release attached. What the firm does not owe her, and what no one is doing her any favours by pretending otherwise, is the perpetual existence of her current role.

What middle management was for

To talk honestly about the role, you have to start by being honest about what it was for. Middle management arose in the late nineteenth century, in the railroads, as the first answer to a particular problem: how do you run an

enterprise too large for a single owner to oversee directly? Alfred Chandler's *The Visible Hand* documents this transition with surgical clarity. The owner could no longer see everything; the workers could not coordinate without supervision; somebody had to stand between them, route information up, route instructions down, and translate the two into the other's language. Middle management was that translation layer. It was indispensable and it was, for the period from roughly 1880 to roughly 2020, the single most important human institution in the development of large-scale organised work.

A middle manager's job, at its best, has consisted of five activities. *Routing*: making sure information that originates in one part of the organisation reaches the part of the organisation that needs it. *Translating*: rephrasing the information so that the receiving party can understand it. *Allocating*: deciding which sub-team or sub-individual should handle a given piece of work. *Chasing*: following up to ensure the work was actually done. *Interpreting*: framing what happened in a way that lets the layer above understand whether to be worried.

If you read this list with fresh eyes, three things become apparent. First, all five activities are mostly *coordination*, not judgment. Second, all five activities are activities the agentic substrate can perform, in many cases better, because the substrate does not get tired, does not have favourites, does not lose track, and does not need to look busy at the end of the day. Third, the parts of management that are *not* on this list — the parts that involve developing people, making strategic calls, providing emotional and political cover, making and defending unpopular decisions — are the parts that the substrate cannot perform at all and that the modern middle manager often does not have time to perform either, because she is too busy doing the five things on the list.

The implication is uncomfortable. The work that consumed middle management's time is now mechanisable. The work that justified middle management's existence was always something else, something the role's day-to-day made it harder rather than easier to do. The agentic firm, by absorbing the coordination work, exposes this. Middle managers in firms that are doing

this transition well are reporting, with some surprise, that they finally have time to do the parts of their job they always thought they were paid for. Middle managers in firms that are not are reporting that their days are getting strange and empty.

Span of control, renegotiated

The classic constraint on the size of an organisation is *span of control*: how many direct reports a single manager can effectively oversee. The number has been studied to death and has hovered, for a century, somewhere between five and ten depending on the nature of the work. The constraint was always a function of how much human cognition one manager could devote to one report — keeping track of their priorities, their progress, their development, their relationships, their problems. With more than about ten reports, the manager simply could not hold all of it in her head, and quality of supervision started to degrade.

Span of control is not a fact of nature. It is a fact of how much *coordination overhead* the manager personally absorbed. If the substrate absorbs most of the routing, translating, allocating, and chasing, the manager's effective span of control rises sharply, because the only thing she has to do per report is the part that genuinely requires her judgment — the development conversation, the strategic call, the moments of friction that need a senior eye. The leading firms in agentic management are running spans of control in the high twenties, sometimes higher, and the managers in those firms are reporting that they are *less* overworked than they used to be, not more, because the work that filled the day was the coordination work the substrate now handles.

This does not mean that organisations should fire two-thirds of their managers tomorrow morning. It does mean that the equilibrium structure of the firm is going to be much flatter, with much wider spans, much fewer layers, and much more senior responsibility per remaining manager. The transition will take five years to play out and will be painful in the years it is playing out. The end state will look, to anyone visiting from 2010, like a different kind of company.

The honest job description

If the five activities that defined middle management are absorbed by the substrate, what is the new job description? It is something like this. The manager of the agentic firm is a *governor of thresholds, an interpreter of exceptions, and a developer of judgment in others*. She decides where the substrate is allowed to act on its own and where it must escalate. She interprets the cases the substrate flags as exceptions, deciding which require action and which require a change to the policy that produced the flag. She develops the smaller team that reports to her into people who can themselves exercise judgment, take responsibility, and grow into the roles above them. She is a person whose value is concentrated in fewer, more consequential moments, and whose authority is correspondingly more visible when it is exercised.

This job is harder than the old one in some ways and easier in others. It is harder because the moments when the manager has to act are higher-stakes and more visible. There is no longer a wall of routine activity behind which to hide a difficult decision. It is easier because the manager is not spending the day on coordination work that nobody respected and everybody was tired of. The managers who flourish in this role are typically the ones who, in their previous roles, were already trying to do this work and were complaining, privately, that the rest of the job was getting in the way. The managers who struggle are the ones for whom the rest of the job was the part they were good at.

Performance management without performance theatre

One of the more visible casualties of this transition is the apparatus of formal performance management. The annual review, the calibration session, the nine-box grid, the forced ranking, the structured goal-setting cycle — most of it was an attempt to impose process discipline on an activity that is, fundamentally, a series of judgments by humans about other humans. The process discipline added consistency at the cost of meaningfulness.

Everybody complains about it. Almost no firm has been brave enough to retire it.

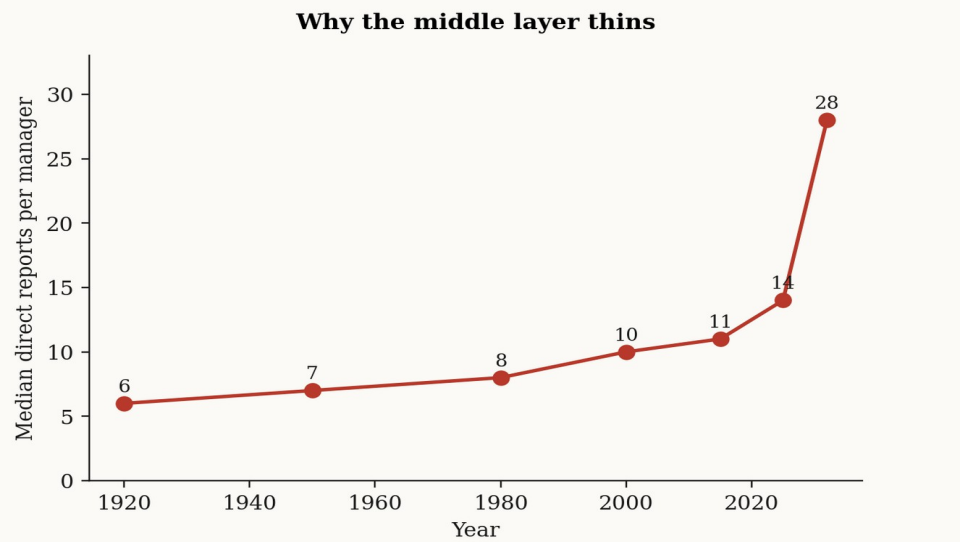
Agentic systems make a different kind of performance management possible — one that is continuous rather than annual, evidence-based rather than memory-based, and concentrated on the small number of consequential observations rather than the long parade of routine ones. The substrate can produce, for any role, a continuous record of the work the person did, the outcomes the work produced, and the patterns that emerged over time. The manager does not have to remember what happened in March; the record is there. The conversation with the employee can therefore be about *interpretation* rather than recollection — what does this pattern mean, what should we do about it, what is the next step in your development.

This is, of course, also the architecture of a surveillance state, and the firms that handle it badly will produce that. The discipline of agentic performance management is the discipline of using the substrate's continuous record without weaponising it — using it to inform conversations rather than to replace them, using it to surface patterns rather than to enforce metrics, and giving employees the same access to the record that managers have. Done well, this is liberating. Done badly, it is a panopticon. The line between the two is not technical. It is cultural and explicit, and the firms that do not draw it explicitly will end up on the wrong side of it by default.

What management theory has to relearn

THE MANAGEMENT LITERATURE OF the last forty years was written about a substrate that no longer exists. Its central concepts — span of control, layers of hierarchy, the manager as a coach, the team as the unit of work, the matrix as a coordination device — were all responses to constraints that the agentic firm relaxes or removes entirely. None of this means the old literature was wrong. It means the old literature was *contingent*, in a way that the people who built their careers around it have been reluctant to admit. The next generation of management theory is going to be written about firms that look very different from the firms in the textbooks, by people who do not yet have

tenure, and the rebuilding of the discipline is going to be slower and more contested than the rebuilding of the firms themselves. The early literature is starting to appear in fragments — in operator essays, in startup memoirs, in the occasional honest case study — and a serious operator should be reading it and ignoring the textbook.



Span of control was a function of how much coordination one human could metabolise. The constraint is being renegotiated by software, not by management theory.

The middle manager was a translation layer for a substrate that needed translating. The new substrate writes its own dictionary.

XV. Human Resources After the HR Department

The HR department was a paperwork institution that wrote love letters to itself about being a strategic partner.

The phrase "human resources" is a euphemism that has been doing more work than it deserved for fifty years. The function it names was never really about humans; it was about the *administrative apparatus around humans* — the contracts, the benefits, the compliance, the policies, the forms, the surveys, the trainings, the workflows that tracked headcount and absorbed grievances and produced the documents that allowed the firm to claim, in litigation, that it had behaved reasonably. The administrative apparatus existed because employing humans is legally and operationally complicated, and somebody had to do the complicated things. That somebody, in most large organisations, was a department that grew steadily larger over the decades and that periodically rebranded itself in increasingly aspirational language: personnel, human resources, people operations, talent, employee experience, the chief people officer's office. None of the rebrandings changed the underlying centre of gravity. The work was, and still is, mostly paperwork.

This is the chapter most likely to be read by HR professionals as an attack. It is not. It is an attempt to draw a clean line between the part of the function that is mechanical and the part that is profound, so that the profound part can finally be done by people who have time to do it, and the mechanical part can be absorbed by the substrate that is going to absorb it whether HR leaders prepare for it or not.

The lifecycle, by what is actually judgment

Take the canonical employee lifecycle — source, screen, interview, offer, onboard, develop, review, exit — and ask, for each stage, what fraction of the work is judgment and what fraction is mechanical. The honest answer is uncomfortable.

Sourcing is almost entirely mechanical. The work of finding plausible candidates from a description of a role, screening for basic qualifications, and surfacing the strongest matches is exactly the kind of pattern-matching the substrate does well. The human contribution is in deciding what kind of candidate the role actually requires, which is a few hours of senior judgment per role rather than a few weeks of recruiter time per req.

Screening is more mechanical than the profession will admit. Most initial screens are checks against criteria the role description already specified, conducted by humans who did not write the description and are working from a checklist. This is the substrate's natural territory, and the firms that are doing it well are running screens at near-zero marginal cost while the recruiters spend their time on the small number of candidates who survive.

Interviewing is the first stage where human judgment becomes irreducible. The interview is where the firm decides whether it wants to spend years working with a particular human, and the substrate cannot replace the felt experience of one human evaluating another. What the substrate can do is prepare the interviewer with a sharper picture of the candidate, generate better interview questions specific to the role and the candidate, and produce more rigorous post-interview synthesis than the typical hand-written feedback form. The interview itself remains human. The apparatus around it does not need to.

Offering is mostly mechanical. The substrate can run the comp benchmarking, draft the offer, handle the initial back-and-forth, and surface to a human only the cases where the negotiation requires judgment.

Onboarding is almost entirely mechanical, despite a generation of HR rhetoric about the importance of the human onboarding experience. The honest reality of most onboarding is administrative: paperwork, system access, benefits enrolment, mandatory trainings, scheduled introductions. The substrate does all of this faster, more consistently, and without making the new hire feel as if she is being processed by a slow bureaucracy. The genuinely human parts of onboarding — the first conversations with the manager, the integration into the team, the early sense of whether the firm is a good fit — are exactly the parts that legacy onboarding programmes have

been doing badly precisely because the administrative parts consumed all the available attention.

Developing is where human judgment matters most and where the substrate can help the most. The substrate can produce, for each employee, a continuous picture of what they have worked on, what they have produced, what they have learned, and where their next growth edge probably lies. The development conversation is then between two humans who are looking at the same evidence rather than between two humans who are trying to remember the same six months. This is one of the few areas where the agentic transformation makes the human work *more* important, not less.

Reviewing — the formal performance management ritual — is, as discussed in the previous chapter, mostly theatre that the substrate can replace with continuous evidence-based observation.

Exit is mostly mechanical, including the compliance-heavy parts that nobody likes doing but that have legal consequences. The human part of an exit is the conversation, and the conversation is better when the administrative parts are not consuming the time the conversation needed.

If you sum up the mechanical and the judgment fractions across the lifecycle, the conclusion is that something like 70 percent of the work an HR department does today can be moved to the substrate without any meaningful loss of quality, and something like 80 percent of the *attention* that is currently spent on mechanical work could be redirected to the parts of the job that actually matter. The HR department of 2030 will be smaller, more senior, and concentrated on a much narrower set of activities. The activities that remain will be the ones that justified the function's existence in the first place, and it will be done by people who have time to do it well.

Talent acquisition becomes a different game

Recruiting is the part of the HR function that the substrate has already started to absorb most visibly, and the first wave of agentic recruiting tools is producing exactly the predictable failure modes. Firms are using agents to flood candidates with personalised outreach, which has produced a candidate experience that is, if anything, worse than the previous generic outreach

because candidates can now smell the algorithm even when they cannot prove it. The agentic recruiting tools that work are the ones that use the substrate to do the unglamorous parts of the work — sourcing, screening, scheduling, paperwork — and reserve the human contact for the moments when human contact is actually warranted. The agentic recruiting tools that do not work are the ones that try to use the substrate to fake intimacy at scale.

The deeper change is that the recruiter's job is no longer about throughput. It is about *judgment about what the firm needs*. The recruiter who can sit with a hiring manager, push back on a vague job description, articulate the actual capability the role requires, and design an interview process that surfaces the relevant signal — that recruiter is now the most valuable person in the talent organisation, because the substrate can absorb everything else. The recruiter who is mostly a coordinator of pipeline is performing work the substrate now does better.

The employee experience problem

THE DISCIPLINE OF "EMPLOYEE experience" arose in the 2010s as a recognition that the employee was, in some meaningful sense, the customer of the HR function. This was a good idea. It was also undermined, almost from the start, by being implemented as a layer of additional surveys, additional engagement initiatives, additional forms, and additional internal communications, all of which the employees experienced as more administrative friction rather than less. The substrate offers a way out: the parts of the employee experience that are administrative friction can be made invisible, and the parts that are about the relationship between the employee and the firm can finally be the focus of the function.

A well-built agentic HR system handles the administrative side of employee experience in the same way a well-built agentic finance system handles the close — continuously, in the background, without anyone having to ask. Benefits questions are answered by the substrate. Policy questions are answered by the substrate. Logistics around expenses, time off, equipment, and access are handled by the substrate. The employee notices the function

only when something requires actual human attention, and at that moment the human is available because she is not buried in tickets about parking passes.

Culture as the only durable HR product

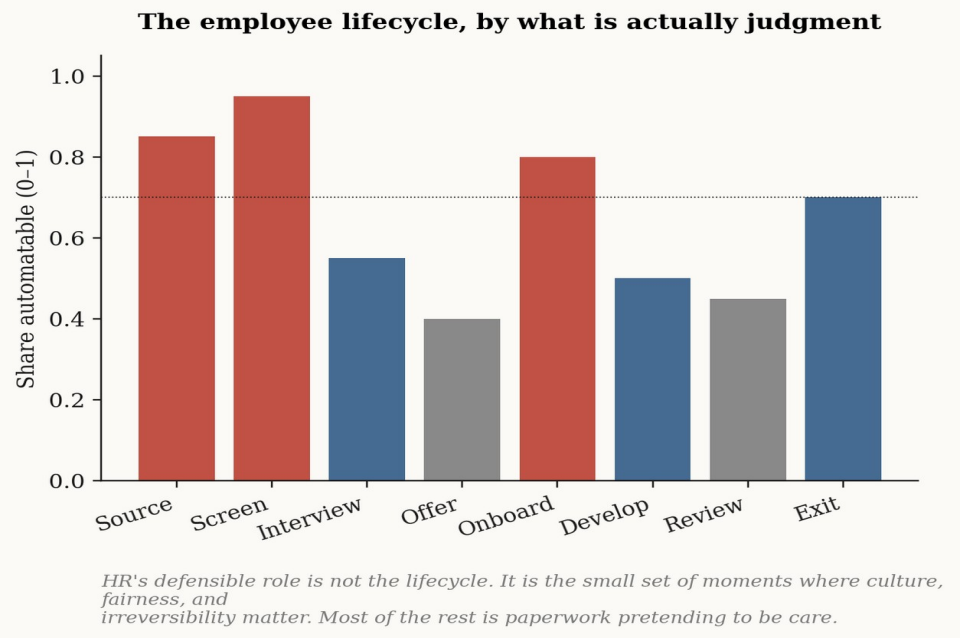
If the lifecycle is mechanical, the talent function becomes a small senior team, and employee experience is mostly substrate, what is the HR department of 2030 actually for? The answer, slightly old-fashioned and increasingly obvious, is *culture*. Culture is the one HR product that the substrate cannot manufacture, because culture is the residue of a long history of consistent decisions about how the firm treats its people, what behaviours it rewards, what behaviours it refuses to tolerate, and what kind of moments it makes possible between the people who work there. An agent can produce employee communications in the firm's voice. It cannot produce the firm's voice itself. The voice was earned, slowly, by the small number of decisions a real human team made about what the firm would stand for and what it would not.

The chief people officer of the agentic firm is closer in temperament to a cultural editor than to a function head. Her job is to define, defend, and continuously refresh the small set of cultural commitments that make this firm a place worth working in, and to make sure those commitments survive the substrate's constant production of more, faster, and cheaper everything. The CPOs who survive this transition are the ones who can articulate, in plain language, what their firm is *for the people who work there* and how that is different from what the substrate would produce by default. The CPOs who continue to treat HR as a process function will be running smaller departments every year and explaining, in increasingly anxious terms, why nobody seems to take them seriously in the executive meeting.

The honest part, again

A final paragraph for the people whose jobs this chapter has just described as mostly mechanical. The work you have been doing was not unimportant. It was load-bearing, in exactly the way that the work of every clerk and every

typist was load-bearing in earlier waves of automation. The fact that it can now be done by a substrate does not erase the contribution it made or the skill it required. What it does mean is that the firm no longer needs as much of it, and the people who built careers around it deserve to be told the truth about that, given a real transition with real money, and given the chance to move into the parts of the function that the substrate has just made more valuable rather than less. That is the honest deal. Anything else is dishonest in a way that the people leaving and the people staying will both feel.



The HR department was paperwork pretending to be care. The substrate does the paperwork. The care, finally, can be the work.

XVI. Customer Service After the Queue

The customer was always trying to talk to someone who could fix the problem. The queue was the firm's way of making sure she could not.

Customer service is the function that has been promised AI transformation longest, has been disappointed by it most often, and is, in 2026, on the edge of a transition that will be more profound and less painful than the previous false starts. The previous false starts — the IVR systems of the 1990s, the first generation of chatbots in the 2010s, the promise of AI-powered self-service that absorbed most of the customer service investment of the early 2020s — all failed in roughly the same way. They tried to use machines to keep customers *out of contact* with the firm. The customers noticed, resented it, and rated the firm down. The firms responded by hiring more humans to handle the cases the bots had refused to escalate. The net effect of two decades of "AI in customer service" was, for many enterprises, a more expensive customer service operation that customers liked less than the one it replaced.

The agentic transition is different because it inverts the goal. The previous goal was to *deflect* contact. The new goal is to *resolve* contact, fully and at the first attempt, with whatever combination of substrate and human is best suited to the case. The difference is not subtle. Deflection optimises for cost per interaction, which trains the firm to make the interaction as unpleasant as possible so that fewer customers have it. Resolution optimises for *the customer's problem being gone*, which trains the firm to make the interaction as effective as possible regardless of who or what handles it. The economic logic of resolution is better than the economic logic of deflection, because customers whose problems get solved spend more, churn less, and tell other customers. The firms that figured this out a decade ago and built around it — the ones whose service organisations are remembered as legendary — were not the ones with the best technology. They were the ones with the right goal.

The substrate does not change the goal. It makes the right goal much cheaper to pursue.

The end of average handling time

EVERY LEGACY CUSTOMER SERVICE organisation is measured, in some form, on average handling time. The metric is universal and almost universally misleading. Average handling time treats customer interactions as if they were a homogeneous flow with a meaningful central tendency. They are not. The distribution is heavily right-skewed, with most contacts being short and routine and a small number being long, complicated, and consequential. The interesting story lives in the long right tail, where customer trust is destroyed and where the firm's reputation is actually being made or unmade. The average tells you nothing about the tail. Worse, optimising for the average actively damages the tail, because the agents who are penalised for long interactions learn to cut off the complicated cases, which are the cases that mattered most.

A serious customer service organisation in 2026 should be measured on the *shape of the distribution*, not on its mean. The relevant questions are: how short can the routine cases be made (the substrate is going to absorb most of these), how much shorter can the long tail be made (this is where the substrate-plus-human model wins), and how much can the *failure tail* — the interactions that ended without resolution — be eliminated entirely (this is the metric the customer actually cares about). A firm whose median handling time has gone up and whose failure tail has disappeared is winning. A firm whose average handling time has gone down and whose failure tail has held steady is losing without knowing it.

What the agent should and should not do

A WELL-BUILT CUSTOMER SERVICE agent does several things that are obviously useful and a few things that require more care. The obvious uses: answering questions for which the answer is unambiguous and exists somewhere in the firm's knowledge base, executing routine actions on the customer's account,

gathering the information that a human will need before the human gets involved, drafting responses for human review on cases that require it, and following up on previous interactions to confirm the resolution stuck. These are not controversial and are being deployed, with varying success, in most large enterprises today.

The more delicate uses are about *escalation*. The agent has to know when to get out of the way, and the threshold for getting out of the way is one of the operational levers that determines whether the system feels good or bad to the customer. Set the threshold too high — the agent tries to handle too much on its own — and the customer feels trapped in a loop with a machine that is failing to understand her. Set the threshold too low — the agent escalates too readily — and the human queue gets clogged with cases the agent could have resolved. The right threshold is not a constant. It depends on the customer (a high-value enterprise customer should be escalated faster than a low-value one), the topic (an angry customer about a billing dispute should be escalated faster than a question about feature availability), and the agent's own confidence in the case (a low-confidence interaction should be escalated regardless of customer or topic). Tuning these thresholds is the daily operational work of running an agentic service organisation, and it is what separates the firms that get this right from the ones that buy a tool and call it a strategy.

There is also a class of interactions where the agent should *never* be the front line, regardless of what it is technically capable of. Safety incidents. Allegations of fraud or harassment. Cases involving regulated populations. Cases where the customer is in distress in a way that is recognisably human. The cost of getting these wrong is so high, and the value of human contact is so concentrated in exactly these moments, that the system should be designed to route them straight to a human and to do so reliably. Building this routing well is harder than building the agent's response logic, because it requires the system to recognise the case before it has heard it out — and the recognition has to be calibrated against both false positives (humans wasting time on routine cases) and false negatives (the system failing to recognise distress and continuing to robotically reply). Most firms will get this wrong at first. The

firms that take it seriously will get it less wrong each quarter. The firms that do not take it seriously will be the subject of news stories.

The human agent's new job

THE HUMAN CUSTOMER SERVICE agent of the agentic firm is a different role from the human customer service agent of 2020. She is not measured on call volume. She is not measured on average handling time. She is not asked to follow a script that the substrate has already followed more cheaply. She is the person to whom the substrate escalates the cases it cannot handle, which means, by construction, that her cases are the harder ones. They are emotionally heavier, intellectually more demanding, and more consequential per case for the firm's relationship with the customer.

This is not a worse job than the legacy one. It is, in many ways, a much better one — more skilled, more meaningful, more respected, and better paid. The agents who do this job well are people who can read a situation, build trust quickly with a stranger, hold the firm's interests in mind while honouring the customer's emotional reality, and exercise judgment in situations where there is no clean right answer. These are the qualities that the legacy customer service industry treated as bonus traits and rewarded inconsistently. The agentic transition makes them the entire job, and the people who possess them — many of whom are currently working in legacy queues for poverty wages — are about to discover that they are more valuable than they were told. The firms that figure this out first will assemble the strongest service teams in their industries by quietly hiring the people their competitors are still treating as commodities.

The honest counterpoint: there will be fewer of these humans than there are customer service agents today. Probably much fewer. The firms that try to pretend otherwise will pretend by keeping the existing org chart intact and gradually making the work meaningless until the people quit. The firms that handle it well will be honest about the shape of the transition, give the people who are leaving a real off-ramp, and concentrate the remaining roles on the people best suited to the new shape of the work.

The knowledge base as a living document

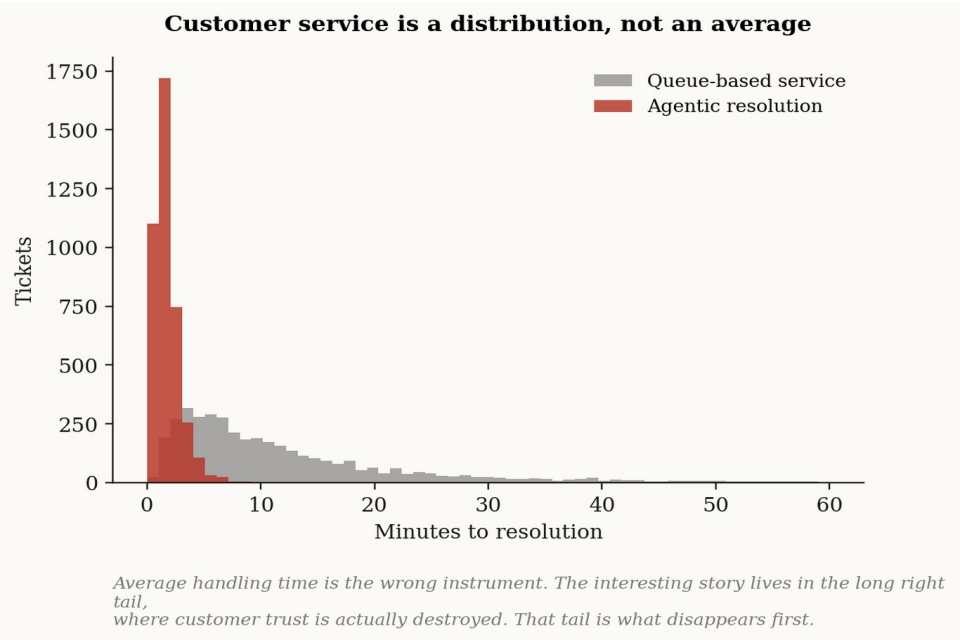
A SUBTLE BUT IMPORTANT consequence of the agentic transition is that the firm's knowledge base — the set of articles, policies, and procedures that the substrate consults when answering customer questions — becomes the most important document the firm maintains. In legacy customer service, the knowledge base was a backwater. It was written by a small team, updated reluctantly, and read mostly by new agents during onboarding and by frustrated agents during difficult cases. Its accuracy was tolerable rather than excellent.

In an agentic service organisation, the knowledge base is the substrate's source of truth. Every error in the knowledge base produces a thousand wrong answers per day at machine speed. Every gap in the knowledge base produces a thousand cases of the agent making something up. The discipline of maintaining the knowledge base has to become continuous, evidence-based, and treated as a first-class engineering activity. Every customer interaction that the substrate handled badly should be treated as a knowledge base bug, traced to its source, and used to improve the source. This work used to be done by nobody and is now load-bearing. The firms that build a serious operational discipline around it will have substrates that get better every week. The firms that do not will have substrates that hallucinate confidently and consistently, in ways that the customer notices before the firm does.

A note on tone

THE LAST THING TO be said about agentic customer service is that the tone of the substrate matters more than people expect. Customers are tolerant of a lot when the firm is solving their problem. They are much less tolerant of a substrate that is solving their problem in a tone that feels insincere, robotic, or — worst of all — inappropriately cheerful when the customer is upset. Tuning the tone of the substrate is one of the small, unglamorous craft skills of agentic service work, and it is one of the things the legacy AI vendors are worst at. The default tones of off-the-shelf chat systems are uniformly bad — chirpy, performatively helpful, full of exclamation points and emojis that signal a kind

of enthusiasm no real human ever displays in a customer service interaction. Strip them out. Let the substrate speak in a neutral, competent, slightly formal voice that takes the customer seriously. The customers will notice the difference, even when they cannot articulate it.



Average handling time is the wrong instrument. The interesting story lives in the long right tail,
where customer trust is actually destroyed. That tail is what disappears first.

The queue was the firm's apology for not being able to listen at scale. The substrate listens. Now the firm has to mean it.

XVII. Security, Trust, and the Right to Stop the Machine

Power becomes legitimate only when it can be halted.

Every powerful system creates a new category of failure that did not exist before it. The steam engine created boiler explosions. The railway created the level crossing accident. Electrification created the household fire from faulty wiring. The automobile created the pedestrian fatality. Civil aviation created the mid-air collision. Nuclear power created the reactor meltdown. The internet created the data breach. Each of these failure categories was invisible to the people designing the technology that produced it, because the failure mode emerged from properties of the technology that were not visible until the technology was deployed at scale. The firms and the regulatory regimes that handled each transition well were the ones that took the failure category seriously *before* the failures became frequent enough to demand it. The firms and regimes that handled it badly waited until the failures forced their hand, by which point the cost was paid in lives, money, and trust.

Agentic systems are about to create their own category of failure, and the firms deploying them are, almost universally, not yet taking it seriously enough. The category does not have a single name. It is the family of failures in which an autonomous system, acting at scale, takes a series of locally reasonable actions that produce a globally unreasonable outcome, at a speed and breadth that no human in the loop could have intervened in. The failure is not malicious. The failure is not, in any meaningful sense, the substrate's fault. The failure is what happens when capability is deployed before the controls that should govern it, by people who could not imagine the failure mode because they had never seen one. The right way to think about this is not as a security problem in the legacy sense. It is as a problem of *legitimacy* — the question of what gives the agentic system the right to act, who has the standing to revoke that right, and what mechanisms are in place to revoke it before the consequences become irreversible.

The blast radius problem

A USEFUL CONCEPT HERE is the *blast radius*. The blast radius of an action is the scope of what the action affects if it turns out to be wrong: how many records, how many customers, how much money, how much downstream activity. Legacy enterprise security architectures were built around the assumption that the blast radius of any single action was small, because the actor was a human and the human could only do so much in an hour. The discipline of security was about keeping the wrong humans out, controlling what the right humans could see, and auditing the actions after the fact. The model worked because the temporal and physical limits of human action were the implicit ceiling on the blast radius.

Agentic systems break the model. An agent can take a thousand actions in an hour. Each action might be tiny in isolation, but the cumulative effect of a thousand small actions in an hour is a blast radius that no human control was designed to constrain. A misconfigured agent can send the wrong message to ten thousand customers before anyone notices. A confused agent can update the same record incorrectly across a portfolio of accounts at a rate the audit system was never built to flag. A subtly compromised agent — one whose underlying instructions have been quietly altered by an attacker — can exfiltrate data at a pace and pattern that the legacy data loss prevention tools were not built to detect, because the legacy tools assumed the actor was human and slow.

The architectural answer is to make the blast radius an explicit, designed property of every agent. Every action the agent can take should have a maximum scale beyond which the action requires explicit human authorisation, regardless of how confident the agent is. The maximum scale can be financial (no transaction above $X), reputational (no message to more than Y customers), structural (no change to a record in this set of tables), or temporal (no action that cannot be reversed within Z minutes). The caps are crude and that is the point. Crude caps catch the failures that subtle controls miss. The agentic firm that gets this right will look, from the outside, slightly less impressive than the firm that gets it wrong, because the firm with caps will occasionally pause its agents while the firm without caps will run faster —

until the first morning the firm without caps wakes up to a problem that has propagated overnight to a thousand customers, and discovers that there was no point in the chain at which any human had the chance to say no.

Identity is the new perimeter

The legacy security model treated identity as a thing humans had — names, accounts, badges, access cards. The agentic firm has to extend this model to non-human actors. Every agent in the system should have its own identity, distinct from the identity of the human who deployed it and distinct from the identity of the system it is acting on. The agent should have its own credentials, its own permissions, its own audit trail, and its own ability to be revoked. This sounds obvious. It is, in 2026, very rarely done. Most enterprise agent deployments give the agent the credentials of a service account that was created in 2014 for a purpose nobody remembers and that has ambient access to half the firm's systems because the original engineer who set it up did not feel like configuring permissions properly. The agent then inherits the over-privilege, and the security team discovers, six months later, that the audit trail of recent changes is opaque because everything looks like it was done by the same service account.

The discipline of agent identity is the discipline of treating each agent as a first-class actor with a name, a purpose, a permission set, and an owner who is responsible for what it does. The permission set should be the minimum required for the agent's stated purpose, and the agent should be unable to take actions outside its permission set even if instructed to. The audit trail should record, for every action, the agent that took it, the human or system that authorised the agent to act in this case, the reasoning the agent gave for the action, and the outcome. None of this is technically hard. All of it is operationally tedious and most teams skip it under deadline pressure. The teams that do not skip it will be the teams whose agentic deployments survive their first contact with the legal department, the regulator, or the press.

The right to stop the machine

Of all the controls that an agentic firm needs to build, the most important is the easiest to articulate and the hardest to actually implement: a *kill switch*. The right to stop the machine. The mechanical, immediate, no-questions-asked ability to halt an agent or a class of agents and unwind their recent actions. The kill switch should be exercisable by a defined set of humans — not a single individual, because that creates a single point of vulnerability, but not a committee, because committees deliberate and the kill switch is for moments when there is no time to deliberate. A small group, with clear authority, with practiced procedures, and with the technical ability to actually halt the system rather than just send an angry email to the team that deployed it.

The kill switch is the hardest control to implement because it cuts against every other instinct of the engineering organisation that built the system. The engineers built the system to run autonomously and to be reliable. Halting it feels like a failure. The product team built the system to deliver value, and pausing it pauses the value. The finance team budgeted for the system on the assumption that it would run continuously. The kill switch is, in effect, a standing veto on all of these expectations, exercisable by people who may not have built the system, in moments when the people who did build it will be telling them everything is fine. This is exactly the position that air traffic controllers, nuclear plant operators, and emergency room physicians have always occupied, and it is the position that the agentic firm needs to create for the first time inside the corporate management hierarchy. Some of the firms that build this role will discover that they have hired the wrong personality into it. The right personality is unusual: someone with the technical depth to understand what the system is doing, the institutional standing to halt it without being overruled, and the temperament to do so calmly under pressure. The firms that find this person early will save themselves the kind of incident that ends careers.

Adversarial agents

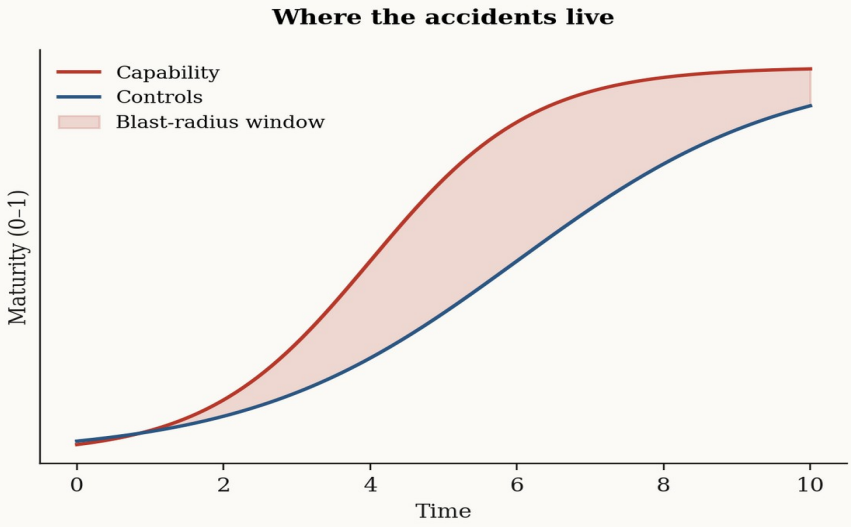
A SPECIFIC SUBCATEGORY OF the security problem deserves to be called out, because it is going to become more important than most security teams currently realise. The agentic substrate is, by construction, susceptible to instructions that come from outside the firm. A customer service agent reads emails from customers; some of those emails will contain instructions from attackers trying to manipulate the agent into doing something it should not. A document-processing agent reads documents from suppliers; some of those documents will contain instructions hidden in ways that the supplier did not intend or that an attacker has injected. A web-browsing agent visits web pages; some of those web pages will contain instructions designed to redirect the agent from its intended task to an attacker's task. This class of attacks — sometimes called prompt injection, sometimes called indirect prompt injection — is the closest equivalent the agentic firm has to the SQL injection attacks that plagued the early web. The defences are, in 2026, immature.

The right posture is to assume the attacks will succeed at some rate, design the system to limit the damage when they do, and build detection and response procedures around the assumption that the substrate is occasionally going to be lied to by something it is reading. This is uncomfortable for engineering teams that are used to thinking of security as a problem of keeping bad inputs out. In an agentic system, the bad inputs are inside the perimeter by definition, because the agent's job is to read inputs from sources the firm does not control. The discipline that works is a combination of reduced privilege (the agent cannot do much damage even if it is compromised), human review thresholds for high-stakes actions (the cost of confirming a few extra times is less than the cost of being deceived once), and continuous monitoring for the patterns that indicate the agent has gone off-task. None of these are perfect. All of them are necessary. None of them excuse the firm from taking the threat seriously, which is the most common posture in 2026 and the one that will be remembered in 2028.

Trust is a one-way ratchet

A FINAL, SOBERING OBSERVATION. Trust in autonomous systems is a one-way ratchet. It accumulates slowly, over months of uneventful operation, and it is destroyed completely in a single visible incident. The firms that have built the agentic systems are going to discover that the customers, the employees, the regulators, and the board do not care how many thousands of correct actions the system took before the incident; they care about the incident. This is unfair. It is also accurate, and it is the political reality the agentic firm operates inside.

The implication is that the discipline of security and trust in the agentic firm is not the discipline of avoiding all possible failures — that is impossible — but the discipline of *handling failures well when they occur*. The firms that will survive their inevitable first incidents are the ones that have built, in advance, the procedures for noticing the incident quickly, halting the system promptly, communicating honestly with affected parties, and remediating in a way that is visible and credible. The firms that will not survive are the ones that built impressive systems with no incident-response capability and that, when the incident comes, react with denial, delay, and the kind of corporate language that makes everyone trust them less. The first instinct of most large organisations under pressure is the second one. The work of building the first one has to be done in advance, by people who are willing to think clearly about what might go wrong while everything is going right.



Every powerful technology produces a window in which capability outruns control. In agentic systems
that window is shorter and the consequences are wider. The window must be governed, not denied.

A system that cannot be stopped is not a system. It is a hostage situation in which the hostages are paying a subscription.

XVIII. Change Management for People Who Hate the Word Change

"Change management" is what we call it when we have to talk people into something they would not have chosen on their own.

The phrase *change management* arrived in the corporate vocabulary in the 1980s, peaked in the 1990s as a kind of secular religion, and has since become one of those terms that everyone uses and no one quite respects. The reason is that the discipline, as it is usually practiced, is mostly an apparatus for *softening the blow* of decisions that have already been made by people who never considered the option of not making them. The change is a fact; the management of the change is the part where the consultants are paid to convince the people affected that the change is good for them. Most large transformations fail to deliver their stated benefits, the post-mortems blame the change management, the next transformation is announced, and the same consultants are hired again, often with the same slides. The pattern is so reliable that one suspects the failure is not a bug.

This chapter is about how to do something better, in the specific context of agentic transformation, where the stakes and the resistance are higher than in most of the changes the discipline was built for. I am going to make the unfashionable claim that the standard playbook is wrong about almost every important question and should be replaced by a smaller, harder, more honest set of practices. The standard playbook is wrong because it was built to manage the deployment of new IT systems, which were experienced by employees as marginal additions to their existing work, and is being applied to a transition that is experienced by employees as a credible threat to their livelihoods. The two situations require different politics. Pretending they require the same politics is the central mistake.

Why the standard playbook fails here

THE STANDARD CHANGE MANAGEMENT playbook has roughly five elements. *Communicate the vision*, repeatedly and from the top, until everyone has heard it. *Identify champions* in each affected group, who will model the new behaviour and bring their colleagues along. *Train everyone* on the new tools, processes, and expectations. *Measure adoption* and intervene where it is lagging. *Celebrate wins* and tell the success stories. The playbook is not stupid. It works, modestly, when the change is modest. It fails, sometimes catastrophically, when the change is deep enough that the affected people understand correctly that the official story is not the full story.

The agentic transformation is one of those deeper changes. The official story is *we are augmenting your work to make you more productive*. The actual story, in many cases, is *we are absorbing the work that some of you do, and there will be fewer of you in two years*. The employees can usually tell the difference between the two stories. They can read the headcount projections, watch which roles are being backfilled, notice which functions are getting investment and which are getting "natural attrition", and infer the trajectory long before the leadership team is ready to discuss it openly. The standard playbook addresses the official story, which means it is addressing a story the employees do not believe, which means the playbook makes things worse rather than better. The champions feel like collaborators. The training feels like a final paycheck wrapped in a tutorial. The communication feels like spin. The metrics get gamed. The success stories get resented. By the time the leadership team realises the playbook has failed, the trust deficit is too large to repair without the kind of honesty the leadership team was avoiding in the first place.

The honest version

THE HONEST VERSION STARTS with telling the truth about the trajectory. Not in slogans, not in carefully drafted statements, not in town halls where the questions are screened in advance, but in plain language, early, and to everyone affected. *Here is what we are building. Here is what it does today. Here is what it will be able to do in a year. Here is which roles we expect to*

grow, which to shrink, and which to disappear. Here is how we will treat the people in each category. Here is what we are not yet sure about. Here is when we will be more certain and how we will tell you. The leadership team that says these things, in this order, with specifics, will get a strange and immediate response. Some employees will be relieved that someone is finally talking to them like adults. Some will be furious. A few will quit. Almost none will be apathetic, which is the failure mode the standard playbook produces.

The honest version is hard for two reasons. The first is that leadership teams genuinely do not know the future with the precision that the honest version implies. They have hypotheses, plans, and fears, but they do not have certainty, and they are reluctant to say things in public that they may have to walk back. The standard advice in such situations is to say less. The honest advice is to say more, but to be explicit about the level of certainty attached to each claim. *We are confident that the support team will be smaller. We are uncertain whether the engineering team will grow or stay flat. We do not yet know what the marketing function will look like in three years.* The employees will respect this. They will not respect a spin that pretends to know what the leadership does not know.

The second reason it is hard is that the honest version requires the leadership team to commit, in advance, to how it will treat the people whose roles are going to disappear. This is the part that the standard playbook is designed to defer. *We will figure out the people side as we go.* The people side does not figure itself out. The people side has to be designed, funded, and announced in advance, because the announcement is the credibility test of everything else the leadership says. A firm that says *we will treat the affected employees with dignity* and then offers two weeks of severance and a LinkedIn workshop is a firm that has destroyed its own credibility for the next ten years. A firm that announces, in advance, generous severance, real retraining funding, real hiring preference for internal moves, and a public commitment to the people whose jobs are absorbed, will find that the same announcement does most of the change management work for it. The remaining employees will be willing to participate in the transition because they will believe the firm will treat them fairly if they end up on the wrong side of it.

The diffusion is not uniform

A SECOND OBSERVATION, WHICH complicates the honest version. The transition is not going to happen evenly across the firm, and the parts of the firm where it happens fastest will not be the parts where the leadership team predicted. Diffusion of any new capability inside an organisation follows a pattern that has been documented for sixty years and that the agentic transformation will not exempt itself from. There are early adopters (a small minority who try the new thing the moment it is available), early majority (the larger group who try it once a few early adopters have made the case), late majority (the larger group still who adopt only when not adopting is becoming embarrassing), and laggards (the people who never adopt and have to be either reorganised around or replaced).

The interesting and underappreciated fact is that the early adopters in any agentic transformation are almost never in the functions or departments that the leadership team expected. The leadership thinks, naturally, that engineering will move fastest because engineers are technical, or that finance will move fastest because finance is rigorous, or that customer service will move fastest because customer service is process-driven. In practice, the early adopters are usually individuals scattered across the firm whose personal dispositions favour experimentation, who are politically able to try things without permission, and who happen to be working on a problem the substrate is well-suited to absorb. They are almost never the people the change management programme identifies as champions. They are usually the people the change management programme is trying to manage around.

The implication is that the firm should look for these natural early adopters and invest disproportionately in them, even when they are not in the organisationally convenient places. Find the senior engineer in marketing who is quietly building agents on her own initiative. Find the operations analyst who has automated half her own job and is afraid to tell her manager. Find the salesperson who has been running her own agentic outreach loop in defiance of the official policy. These people are not problems to be managed. They are evidence of where the transformation is actually happening, and they are usually two years ahead of the formal programme. Promote them, fund

them, give them air cover, and let them show the rest of the firm what the work looks like when it is going right. The rest of the firm will copy them faster than any official communication strategy can produce.

What to measure

The standard playbook measures *adoption* — how many people have started using the new tool, how many have completed the training, how many have logged into the new system this week. Adoption metrics are the change management equivalent of vanity metrics in startup measurement. They are easy to collect, easy to game, and easy to misinterpret. A team that has been told it will be measured on adoption can produce excellent adoption numbers without changing anything important about how it works. The substrate gets clicked on. The work continues as before.

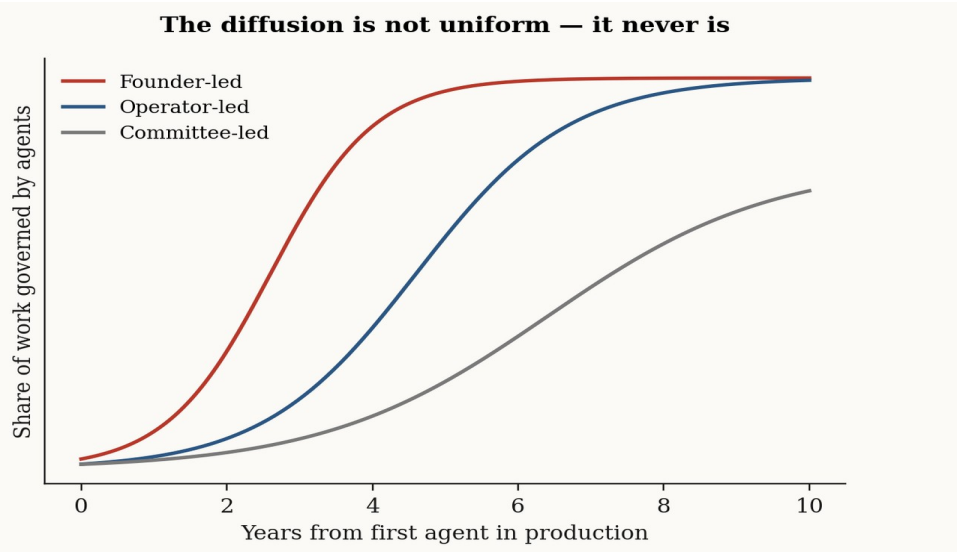
The right thing to measure is *outcome compression*: how much faster the function is producing the outcomes the function is supposed to produce. Cycle time on the relevant operational loops. Cost per outcome. Quality of outcome. Time from signal to action. These are harder to measure than adoption, more politically loaded, and more revealing. They are also the only metrics that tell you whether the transformation is actually happening. A function that has perfect adoption and unchanged outcome metrics is a function that has installed the substrate as a layer of decoration. A function that has imperfect adoption and dramatically improved outcome metrics is a function that is being transformed by a small number of people who matter.

The discipline of small, irreversible commitments

The last principle is the most uncomfortable, especially for leadership teams that came up through consulting or finance. The agentic transformation is best run as a series of *small, irreversible commitments* rather than as a single large programme. A small commitment is something the firm has decided to do that it cannot easily walk back. Closing the books continuously instead of monthly. Eliminating a particular layer of approval. Retiring a particular legacy system. Each of these, in isolation, is small. Each of them,

once made, is hard to reverse without losing face. The accumulation of small irreversible commitments is what the transformation actually consists of.

The reason this works better than the alternative is that it forces the political and operational details to be confronted in the moment when the commitment is made, rather than deferred to a future programme that will be cancelled when the politics get hard. The small irreversible commitment also has the property that it produces a result the firm can see immediately and learn from, which means each commitment makes the next one easier and better-informed. The alternative — the large multi-year transformation programme with quarterly steering committees and a published roadmap — has the opposite property: the politics get worse over time as the original committed leaders move on, the costs accumulate before the benefits are visible, and by year three the programme is being run by people who are mostly trying to survive long enough to put it on their CV. Almost every large transformation programme in the history of corporate IT has failed in roughly this way. The agentic transformation will too, if it is run that way. Small. Irreversible. Honest. In that order.



Carlota Perez's diffusion logic applies in miniature. Adoption is not a function of evidence. It is a function of who in the firm is allowed to commit irreversible action.

Change management is the apology the firm makes for not having had the conversation earlier. Have it earlier.

XIX. The Zero-Person Startup

The new minimum efficient scale is astonishingly close to zero. The interesting question is what survives when there is almost nothing left to subtract.

Every founder has fantasised about this at least once: a company that does not need a company in the old sense. No HR department. No middle managers. No quarterly planning rituals. No coordination tax on every decision. Just a small core of irreducible humans, a great deal of cheap and attentive cognition, and a customer base that experiences the firm as if it were much larger than it is. The fantasy is older than software. It is what every entrepreneur has wanted, in one form or another, since the earliest joint-stock companies. The interesting thing is not that the fantasy is now being marketed to founders. The interesting thing is that it is, for the first time, technically possible.

The zero-person startup is an extreme case, and extremes are analytically useful precisely because they reveal which assumptions about the firm were actually structural and which were merely habitual. The exercise of designing one — even if you do not actually intend to run one — forces a clarity that no normal strategic planning exercise can produce. You discover, very fast, which functions are essential, which are legacy, and which exist only because the previous technological environment required them. You discover which roles were really about judgment and which were really about coverage. You discover which decisions actually had to be made by a human and which had only ever been made by humans because no alternative existed. The clarity is uncomfortable. It is also the most useful diagnostic available to a founder building a company in 2026, regardless of how many people the company eventually employs.

What "zero-person" actually means

Let me be precise, because the phrase is misleading on its own. A zero-person startup is not literally a company with zero humans. It is a company in which the *default* assumption has been inverted. Instead of asking *whom do we hire next*, the founder asks *what autonomous capability do we build next, and what human judgment remains irreducible*. The default is automation; the exception is human work; and the human work that remains is the work that was always supposed to be the point — vision, taste, edge cases, capital, partnerships, the moments where the firm's character is on the line. The humans who remain are not the humans who were doing the routine work. They are the humans whose presence is the company.

A useful frame: imagine a company with one human, then ask which functions could not be performed by that human plus a well-built agentic substrate. Some of them — closing a complicated deal, hiring the second human, sitting in a difficult board meeting — are obvious. Some of them — running customer service, processing orders, drafting contracts, shipping product, writing marketing copy, monitoring operations, paying bills, filing taxes — are not obvious at all, and on inspection it turns out that almost all of them can be performed by the substrate, with the human in the loop only at the moments where the substrate is uncertain or where the stakes are unusual. Now imagine the same exercise with two humans, then five, then twenty. At each step, the question is what each additional human is for. The honest answer, in 2026, is usually that the marginal human is for *judgment under uncertainty in a specific domain*, and that the firm needs only as many such humans as it has irreducibly human domains. This is often a very small number.

The economics

The economics of the zero-person startup are not subtle. A traditional venture-backed software company in 2020 spent something like 70 percent of its operating costs on salaries, with the remainder going to infrastructure, marketing, and the various overheads of running a corporate entity. The salary line is what determined how much capital the company needed, how

aggressively it had to grow, how big the eventual exit had to be to justify the round, and how much of its time it spent recruiting, managing, and replacing humans. The salary line is what produced the venture capital model as we know it. Every detail of how startups are funded, valued, and exited is downstream of the assumption that building a software company requires hiring a lot of people.

When the salary line collapses — not to zero, but to a fraction of what it was — the rest of the equation rearranges itself in ways that the venture capital industry is still adjusting to. The capital required to reach a given level of revenue falls. The time required falls too. The optimal team size at each stage falls. The pressure to grow into a justification for the last round of funding falls. And, perhaps most consequentially, the calculus of who should start a company shifts. The kind of person for whom the salary-driven model was prohibitive — the operator with deep domain expertise but no appetite for managing a hundred-person organisation, the engineer who wants to build something serious without becoming a CEO, the second-time founder who is tired of the recruiting treadmill — finds that the kind of company they actually want to build is, for the first time, financially viable. The result is a Cambrian explosion of small, durable, profitable companies built by people who would never have started a company under the old economics. Some of these companies will become large. Most will not. None of them will look like the ones in the textbooks.

Three categories of zero-person company

IT HELPS TO DISTINGUISH three patterns, because they have different requirements and different ceilings.

The *micro-SaaS at maximum extension* is the simplest case. A single founder, or a tiny team, builds a software product whose entire operations — customer acquisition, support, billing, infrastructure, content, even most engineering — runs through agentic systems. The founder spends her time on the parts that matter: the product roadmap, the rare difficult customer conversation, the strategic decision about which adjacent market to enter

next. Revenue per founder in these companies is the most extreme on record, sometimes in the millions of dollars per year per human. The ceiling is real: at some point the product needs to grow into a category that requires actual humans to defend, and the founder either accepts the ceiling or hires the first non-founder. Many will accept the ceiling. The economics are good enough that there is no reason to grow into a large company unless the founder personally wants to.

The *agent-native vertical* is the more ambitious pattern. A small team builds a company whose core offering is not software at all but an *outcome* — a piece of work that used to be done by a service firm full of humans. Legal research, contract review, financial analysis, market research, customer support outsourcing, lead generation, content production, certain kinds of consulting. The agent-native vertical company offers the same outcome as the legacy service firm, at a fraction of the cost, with a tiny fraction of the human team. Its competitive advantage is not the model — the models are commoditised — but the operational discipline of running the agentic substrate well, the domain expertise of its small human team, and the trust it has built with its customers. The ceiling on these companies is much higher, because they are competing in markets historically defined by labour-intensive services, and the labour cost gap is large enough to justify a great deal of growth.

The *infrastructure for the previous two* is the third pattern, and it is where most of the venture capital is currently going, partly because it is the pattern most familiar to investors who came up in the SaaS era. Tools for building agents. Platforms for orchestrating agents. Marketplaces for agent components. Eval frameworks. Memory systems. Identity layers. Some of these will be enormous companies. Many will be acquired by the dominant model providers within five years. The interesting observation is that the infrastructure layer is the *least* characteristic of the new era — it is the layer that looks most like the previous era's businesses, just with newer logos. The genuinely new businesses are in the first two categories.

What does not work

A FEW THINGS SHOULD not be attempted in the zero-person form, and the founders most likely to attempt them are the ones who have read the most enthusiastic versions of this story without the caveats.

Anything that requires *trust at the moment of the transaction* — high-stakes professional services, certain kinds of healthcare, certain kinds of legal work — does not work as a zero-person company, because the customer is paying, in part, for the presence of a credentialed human at the moment when something matters. The substrate can do the work. It cannot, in the eyes of the customer, *take responsibility* for the work in the way the customer needs.

Anything that requires *physical presence* — installation, repair, hands-on operations, certain kinds of construction — has an obvious human floor that the substrate does not raise. The substrate can dispatch, schedule, route, optimise, and follow up, but somebody still has to climb the ladder.

Anything that requires *political navigation* — large enterprise sales to regulated industries, government contracting, certain kinds of partnership development — requires senior humans whose presence is the value being purchased. The substrate can prepare them and follow up for them. It cannot replace them.

The successful zero-person founders are the ones who pick the markets where these constraints do not bind, build the company tightly around the substrate, and avoid the temptation to grow into the kind of business they actually wanted to escape from in the first place.

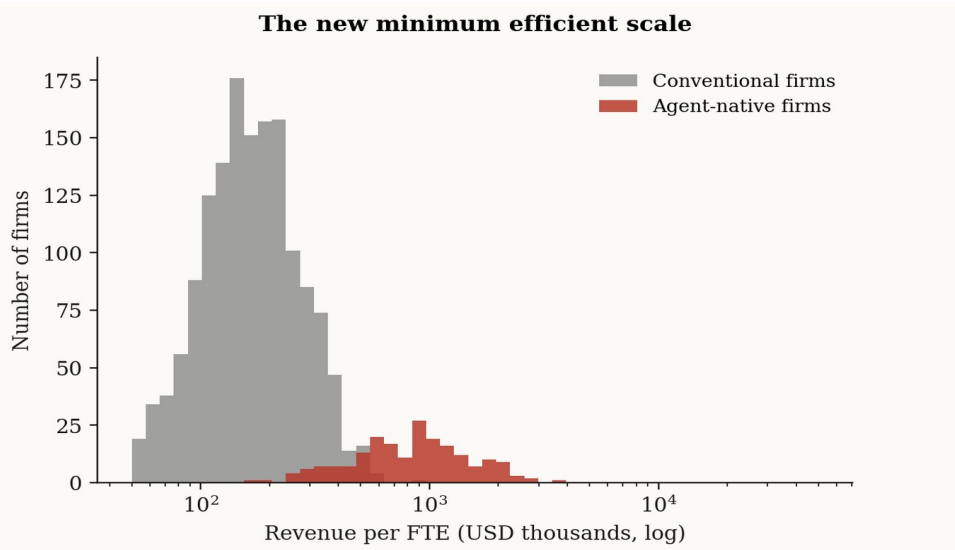
What the founder becomes

THE FOUNDER OF A zero-person startup is not, in any meaningful sense, doing less work than the founder of a traditional company. She is doing different work. She spends almost no time on recruiting, no time on management, no time on the political work of holding a hundred-person organisation together. She spends much more time on the things that founders always wanted to spend time on but rarely could: deeply understanding the customer, designing the product, making the strategic calls, building a small number of consequential relationships with partners and investors and the few

employees she does hire. The job is more concentrated and, in many ways, harder, because there is no organisation to hide behind. Every decision is hers. Every mistake is hers. Every success is hers, in a way that it never quite is in a larger company.

This is a kind of work that suits some people enormously well and some people not at all. The founders who flourish in this form are the ones who were always frustrated by the management overhead of a traditional company and who find the agentic substrate liberating in the same way that a craftsman finds a good set of tools liberating. The founders who struggle are the ones whose temperament needed the social structure of a team, the ritual of management, the daily distraction of running an organisation. The latter are not wrong to need these things. They are just not, in the current technological moment, building zero-person companies.

The most useful framing for any founder considering this form is not *can I run a company with no employees* — almost no one literally does that — but *what is the smallest team that can credibly hold the business together, and what would each person on it be for*. If you can answer that question with five names and a clear description of each person's irreducible contribution, you are ready to build the company you actually want to build, regardless of what the market expects a company of your kind to look like. If you cannot, you are still in the era of headcount as strategy, and you should reread chapter one before raising money.



The efficient frontier is not a story about cost. It is a story about how few people are needed to hold a business together when most of the cognition is mechanical.

The minimum efficient scale of a firm used to be set by how much labour the firm needed to coordinate. The new minimum is set by how much judgment cannot be delegated.

XX. After the Firm

The future of the company begins when we stop assuming the company is a natural form.

The deepest question raised by this book is not what AI will do inside the company. It is what the company *becomes* once intelligence is cheap, mobile, continuously available, and capable of acting on the world without being routed through a paid human. This is the question that the previous nineteen chapters were trying, in various ways, to circle. It is the question that the consulting class is least equipped to ask, because the consulting class is paid by the existence of the firm in roughly the form it currently occupies, and a question that calls the form itself into doubt is a question that calls into doubt the people asking it. So it has to be asked from the outside, by people who have nothing to lose by asking it and a great deal to lose by not. This chapter is my attempt to ask it honestly, knowing that any honest answer will be partial, contestable, and almost certainly wrong in the ways that matter most. The point is not to arrive at the right prediction. The point is to take the form of the firm seriously as a variable rather than as a constant, because the people who treat it as a variable are the people who will design what comes next.

For most of the history of organised commerce, the firm has been the dominant container of productive activity. The reasons are well-understood. Markets are expensive to coordinate at the level of detail that complex production requires; firms internalise the activity to avoid the transaction costs; the boundary of the firm settles, more or less, where the marginal cost of internal coordination equals the marginal cost of buying the same thing from the market. This is the Coasean account, refined by Williamson, Demsetz, Hart and Holmstrom, and a generation of economists who took the firm as a given object whose boundary needed explanation. Their work was excellent. It also took for granted, because the era took for granted, that the only available substrate for coordination inside the firm was a hierarchy of paid humans with limited cognitive capacity, slow reaction times, and political incentives that did not always align with the firm's. The agentic substrate

violates that assumption, and the violation is the part of the story that the existing theory of the firm cannot easily absorb.

Coase, revisited

Imagine the Coasean equilibrium graphically. On the horizontal axis, the size of the firm. On the vertical axis, the marginal cost of producing one more unit of activity inside the firm versus buying it from the market. The internal cost curve rises with size, because larger firms accumulate coordination overhead and bureaucratic friction. The market cost curve falls with size, because larger firms have more options and more bargaining power. The intersection of the two curves determines the optimal size of the firm. This is a simplification, but it captures the spine of the theory.

Now ask what happens when the agentic substrate enters the picture. The internal coordination cost curve falls — not slightly, but substantially — because the coordination work that used to be performed by humans is now performed by the substrate at a fraction of the cost and a fraction of the latency. The market cost curve falls too, but for different reasons: it falls because the firms on the other side of the market are also using the substrate to coordinate themselves, which makes them cheaper to deal with. Both curves shift downward. The interesting question is which falls faster, and the answer is that the internal coordination curve falls faster in some kinds of activity and the market curve falls faster in others. The equilibrium point — the optimal size of the firm — does not move uniformly. It moves up for some kinds of activity and down for others, sometimes dramatically in either direction, and the result is not a single new equilibrium but a fragmentation of equilibria across different kinds of work.

In activities where the substrate makes internal coordination radically cheaper — software development, customer service, marketing, finance, certain kinds of operations — the optimal firm gets *smaller*, because much of what the firm used to do internally can now be done by a tiny team with the substrate doing the rest. In activities where the substrate makes the market radically more efficient — sourcing components, finding suppliers,

contracting for specialised services — the optimal firm also gets smaller, because activities that used to be internalised for transaction-cost reasons can now be safely externalised. In activities where the substrate enables coordination at a scale that was previously infeasible — global supply chains, multi-party financial settlements, distributed manufacturing networks — the optimal firm gets *larger*, because coordination is now cheap enough that the limits to scale that used to bind no longer bind. The result is a world in which the average firm is smaller, the largest firms are larger, and the middle of the distribution is being hollowed out. This is the underlying shape of the next decade. Most predictions about the future of the firm are predictions about which slice of this shape the predictor happens to be looking at.

Three futures

It is useful to lay out three plausible futures, not as forecasts but as scenarios that bound the space.

The *concentrated future* is the one in which a small number of very large firms — the model providers and the early agentic platform owners — capture the bulk of the value created by the transition. They do this by owning the substrate everyone else builds on, by controlling the data that the substrate learns from, and by leveraging the network effects of being the default platform. In this future, the smaller firms enabled by the substrate exist, but they exist at the platform's discretion, in much the way that the sellers on a marketplace exist at the marketplace's discretion. The platform sets the rules, takes the rents, and decides which categories of activity flourish and which are quietly displaced. This is the future the model providers are betting on, and there are credible reasons to believe they are right. Network effects are real. Data accumulates. Switching costs grow over time. The dominant model provider of 2030 may have a moat that looks, in retrospect, indistinguishable from the moats of the dominant operating system providers of 1995.

The *fragmented future* is the one in which the substrate becomes a commodity, models become interchangeable, and the value migrates downstream to the firms that use the substrate well in specific verticals. In

this future, the dominant companies of 2030 are not the model providers but the vertical operators who have built durable, agent-native businesses in specific markets — legal, financial, healthcare, education, logistics, professional services — and who have made themselves the default destination for customers in those markets. The model providers continue to exist and to make money, but they make less of it than the platform play would have produced, because the value of the model is increasingly subordinated to the value of the operational discipline and the customer relationships that the vertical operators have built. This is the future that most independent operators are betting on, because it is the future in which their own work has the most leverage. There are credible reasons to believe they are right too. Commoditisation of models is well underway. Vertical specialisation is hard to displace. The customer relationship is, historically, more durable than the technology relationship.

The *re-statified future* is the one nobody wants to talk about and everybody privately considers. In this future, the substrate becomes powerful enough and consequential enough that governments start to regulate it heavily, not at the level of model safety (which is the public conversation) but at the level of *market structure* — antitrust against the platform owners, mandatory interoperability between substrates, public infrastructure for the most consequential agentic services, and limits on how much of the firm's coordination can be outsourced to entities outside the firm's national jurisdiction. This future looks unlikely from the vantage point of 2026. It looked unlikely for cloud computing in 2010, for social media in 2015, for cryptocurrency in 2018. It usually looks unlikely until the moment it becomes inevitable. The historical pattern is that every technology that becomes deeply infrastructural eventually attracts the attention of the state, and the agentic substrate is on a trajectory to become deeply infrastructural by 2030. Anyone betting on the previous two futures should hold a small position in this one as a hedge.

The honest answer about which of these futures will arrive is that all three will arrive, in different proportions, in different sectors, in different countries. The interesting strategic question is not which one to predict but which one your own firm is structurally exposed to, and whether you have positioned

yourself to benefit from the version that actually shows up in your particular slice of the economy.

What does not change

I WANT TO END with the part that the optimistic versions of this story tend to skip. A great deal will change about the firm. Some things will not, and the firms that respect the things that do not change will do better than the firms that imagine the substrate has retired the older problems.

Trust does not change. The firm that customers trust is still the firm that customers buy from. Trust is built by long histories of consistent behaviour, and the substrate cannot manufacture it any more than it can manufacture brand. The firms that try to substitute fluent agentic interactions for actual trustworthiness will discover that the substitution does not work, and will spend the next decade trying to undo the damage of having tried.

Judgment does not change. The hard decisions are still hard. The substrate can prepare them, frame them, simulate them, and execute them once they are made, but it cannot make them. The firms that pretend the substrate has absorbed judgment will accumulate a quiet record of bad decisions made fluently and at scale, and they will pay for it in ways that take years to become visible.

Politics does not change. The firm is still a coalition of humans with conflicting interests, ambitions, fears, and loyalties, and the work of holding the coalition together is still political work that has to be done by humans who understand the people involved. The substrate does not absorb politics. It changes which politics matter, which is a different thing, and the politics that matter in the agentic firm are, if anything, sharper and higher-stakes than the politics that mattered in the legacy one.

Responsibility does not change. When something goes wrong — and something will go wrong, often, because the substrate makes mistakes at the same rate as any sufficiently complex system — the question of who is responsible for the mistake is not abstract. It is concrete, legal, and consequential. The firms that have built clear lines of responsibility for agentic

actions will absorb the mistakes and continue to function. The firms that have not will discover, after the first significant incident, that the absence of clear responsibility is itself a kind of failure that the legal system, the regulators, and the press are happy to assign retroactively.

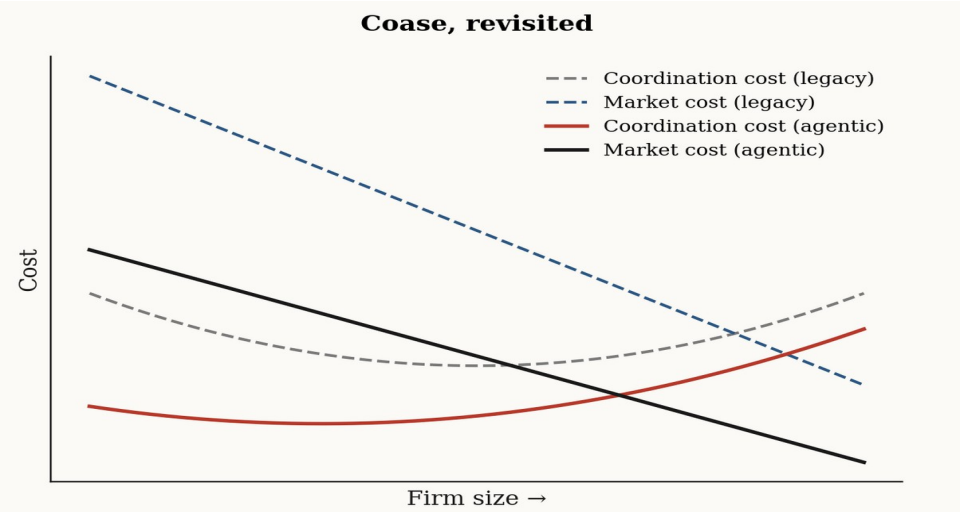
The scarcity of attention does not change. The customer's attention is still scarce. The employee's attention is still scarce. The leader's attention is still scarce. The substrate produces more outputs per unit of input than any previous technology, but the bottleneck has moved to who can decide what is worth attending to. The firms that win the next decade are not the firms that produce the most. They are the firms that decide best what not to produce.

A closing observation

THE QUESTION OF WHAT the firm becomes is, in the end, the question of what work is *for*. The historical answer was that work was the thing humans did in exchange for the means to live. The firm was the institution that organised the work, captured the surplus, and distributed the wages. The agentic substrate complicates this by making more and more of the work performable without humans, which forces a question that every previous wave of automation eventually forced and that the current wave is going to force more sharply than any of its predecessors. *If the work can be done without us, what are we for?* The question is older than the loom and is not going to be answered in this book. What this book has argued, and what I want to leave the reader with, is that the question is now a question for operators and founders, not just for philosophers. The decisions you make about how to build your company over the next five years will shape, in your small slice of the economy, what kind of answer becomes available to the people who work in and around it. That is a heavier responsibility than the one most founders signed up for. It is also a more interesting one, and the founders who take it seriously will build companies that are worth building, in a way that the firms of the previous era did not always manage to be.

The future does not belong to the firm with the biggest payroll, or the smallest payroll, or the most sophisticated substrate, or the cleverest model. It

belongs to the people who can hold a clear picture of what they are trying to build, the discipline to build it honestly, and the willingness to be wrong in public when the picture turns out to need revision. That has always been what good operators have done. The substrate just makes the stakes higher, the pace faster, and the consequences more visible. The job, finally, is the same job it always was. We just have new tools, and new responsibilities, and a much shorter list of excuses.



Coase explained the firm as a response to transaction costs. When both curves fall but coordination
falls faster, the equilibrium size of the firm is renegotiated. That is the deep story.

The firm is not a fact of nature. It was a contingent answer to a particular cost structure, and the cost structure has changed. What we build next is up to us.