

# Context Is Everything

Applications become data stores, the interface dissolves, and the business runs on context, prompts, agents and personal dashboards: a re-envisioning of the business operating model, and what it costs an organisation not to build one.

**ADRIAN HOLLISTER**

FOUNDER, **ALTLIBRE**

- 1 Argument
- 2 Model
- 3 Controls
- 4 Money
- 5 Sovereignty
- 6 Path
- 7 References

## THE ARGUMENT

**15-20%**TRAINING AND CHANGE  
MANAGEMENT

The industry's own guidance for ERP end-user adoption, as a share of implementation budget.

**3 HRS**A DAY ON MANUAL DIGITAL  
ADMINISTRATION

A OnePoll survey of 10,500 office workers, 11 countries, January 2020; 51 per cent say it obstructs their main job.

**8**

## STEPS IN THE TRAINED CEREMONY

Credentials, claim types, matching, split fares: the ritual the model files with a sentence.

**No. 4**

## ALT-LIBRE PAPERS

October 2026. Released under CC BY-SA 4.0.

# 1. The interface was a training course, and the training course was a tax

Every business application in current use is a negotiation between two parties who cannot speak to each other: a human with intent, and a machine that keeps state. The interface is the treaty. Menus encode the machine's ontology; forms encode its schema; workflow states encode its bookkeeping. The human side of the treaty has always been paid in a currency called training.

The size of that payment is measurable. Enterprise resource planning projects are advised, by the industry's own guidance, to give end-user training and change management fifteen to twenty per cent of the implementation budget, and the same industry's failure analyses consistently find that line underfunded: insufficient training recurs among the leading named causes of ERP disappointment. Training is a compensation payment rather than an investment: the organisation pays, in classroom hours and lost productivity, for the privilege of having made its people learn the machine's language.

Below the training layer sits the administrative layer, the work the interface exists to process and often creates. Asana's Anatomy of Work research has repeatedly found knowledge workers spending more than half their week on "work about work": status updates, chasing information, duplicated effort, searching for records that already exist. Salesforce's State of Sales puts manual data entry at around nine per cent of a salesperson's week, with a further nine per cent on the administrative work that surrounds it. A OnePoll survey of 10,500 office workers across 11 countries, published by Automation Anywhere in January 2020, found employees averaging close to three hours a day on manual digital administration, with 51 per cent saying it obstructs their main job. Together these figures are the standing cost of the treaty, not marginal inefficiencies.

The treaty is also becoming more expensive. Gartner's July 2026 forecast puts worldwide software spending at 1.47 trillion dollars for the year, growing at 15.5 per cent, and Gartner's own analysts identify why the price of software is rising: generative AI features are being added to products enterprises already own, and the bill follows. Organisations are paying more for interfaces, training more people to use them, and employing more people to operate the administrative layer the interfaces generate.

The point of this chapter is not that software is bad. The point is that the interface was never the value but a translation layer between intent and state, and translation has always been the part of the system a sufficiently good language model would make unnecessary. That is now a description of the present, not a forecast. The organisation that sees this first stops building translation layers, stops training people to operate them, and returns the hours to the work it employs people to do.

## 2. The expenses system, retired

|   |             |
|---|-------------|
| 1 | Argument    |
| 2 | Model       |
| 3 | Controls    |
| 4 | Money       |
| 5 | Sovereignty |
| 6 | Path        |
| 7 | References  |

## THE ARGUMENT

Take a concrete process every organisation runs, and run it through the new operating model. Expenses is the right example, because everyone knows the ceremony.

The ceremony today: a new joiner waits for credentials to a system they did not choose. They are booked onto a training session, or sent a forty-slide deck, or handed the number of the one colleague who knows how the thing works. They take photographs of receipts, log into a portal, find the correct claim type in a list of claim types designed by someone who left in 2019, match receipts to card statements, discover that the rail fare and the lunch must be claimed separately, misfile both, receive an automated rejection citing a policy they have never read, resubmit, and eventually receive their money with the enthusiasm of a prisoner released early. Somewhere behind the portal, a finance team re-keys, reviews exceptions, and maintains a configuration that only they understand.

The claim itself, the thing of value, is a five-field record: who, when, what, how much, why. Everything else in the paragraph above is interface.

The model tomorrow: the same person finishes a client meeting, photographs the receipt as they leave the restaurant, and says to their assistant, in whatever words they naturally use, "file that with the Plymouth trip". The context layer knows who they are, what the Plymouth trip is, what the expenses policy says about client meals, what their approval thresholds are, what currency they claim in and what their last three claims looked like. An agent reads the receipt, extracts the record, categorises it against the trip, checks the policy, attaches the evidence, writes the claim to the finance system, and reports back: filed, with the lunch under the client entertainment rule, one item flagged because the wine exceeds the per-head allowance, awaiting your decision: claim the excess personally, or drop it. The ledger is unchanged. The audit trail is better than the old one, because it is generated by the action rather than reconstructed afterwards.

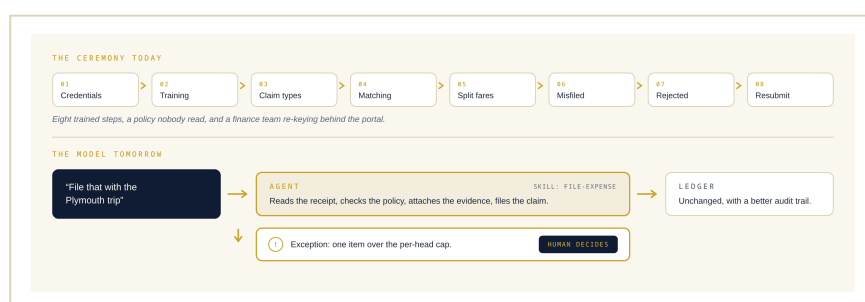


**Figure 1.** The expenses ceremony, before and after. Eight trained steps become two words, an agent, and a human decision.

Nobody was trained. That is the whole point. The training course existed to teach a human the machine's ontology: claim types, field names, workflow states. When the machine holds the ontology instead, the course has nothing to teach. What remains is a policy: meals with clients are claimable, alcohol is capped, rail fares are standard class unless a director approves first. Policy is written for people, kept in one place, and read by the agent that acts.

Two clarifications keep the example rigorous. First, the system of record does not disappear; finance still runs a ledger, and the agent writes to it through a governed interface. What disappears is the system of interaction, the portal, the form, the claim type list, the

- 1 Argument
- 2 Model
- 3 Controls
- 4 Money
- 5 Sovereignty
- 6 Path
- 7 References

## THE MODEL

### 3 PARTS EVERY APPLICATION REDUCES TO

A data store, a policy engine, an interface. The interface is the disposable one.

training deck, and the majority of the support tickets the portal generated. Second, the human is still accountable: the assistant flags the exception and the person decides. Auto assistance is not auto approval. The distinction matters to everything in Chapters 7 and 10.

Expenses is the demonstration case because it is small, universal and uncontroversial. But the pattern is not about expenses. Leave requests, purchase requisitions, timesheets, travel booking, meeting follow-ups, weekly reporting: each is a thin record wrapped in a thick interface, and each reduces to the same three parts described in the next chapter.

## 3. What an application is

Strip any line of business application down to its load-bearing parts and three remain.

A data store. A schema, records, and the quality of both. The claim record, the ledger entry, the contract, the ticket. This is the part that has value after the vendor relationship ends, and the part that outlives every interface built over it.

A policy engine. The rules the organisation has decided to live by: what is claimable, what is approvable, what must be retained, what must never be visible to that role. In most applications this is embedded in configuration, validation logic and code, which is why it is invisible, unauditable and impossible to move.

An interface. Screens, menus, forms, workflows, and the documentation and training required to operate them. This is the part organisations buy, customise, brand, and spend fifteen to twenty per cent of their implementation budgets teaching people to use.

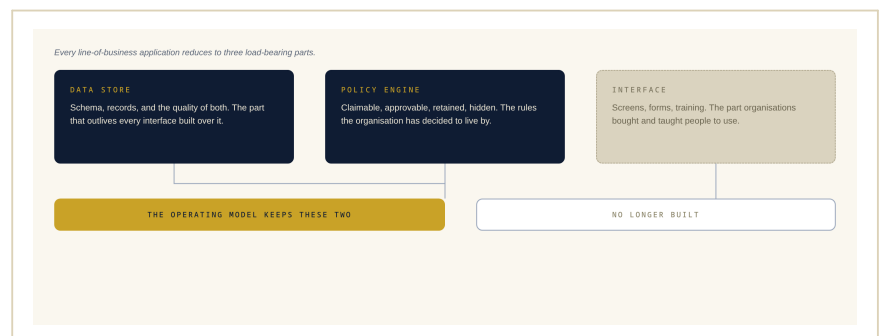


**Figure 2.** What an application is made of. The operating model keeps the store and the policy; the interface is no longer built.

The AI operating model keeps the first two and retires the third. Applications become governed data stores with published schemas, and their embedded policies are extracted into first-class, readable policy that both humans and agents consult. The interface is no longer built, because the prompt is the interface: a surface that already knows the user's language, assembled per person from context. Where a visual surface is still the right tool, such as a timesheet grid, a pipeline board or a financial report, it is generated for the task and the person rather than shipped as a product, and it is disposable.

- 1 Argument
- 2 Model
- 3 Controls
- 4 Money
- 5 Sovereignty
- 6 Path
- 7 References

## THE MODEL

7  
COMPONENTS OF THE CONTEXT  
LAYER

Identity, current work,  
entitlements, obligations,  
history, preference,  
environment.

This inversion changes what development means. Today a new process is a procurement exercise and a build: screens, permissions, integrations, test scripts, training materials, a support queue. In the model this paper describes, a new process is three artefacts: a schema, a policy document, and a skill that acts on both. The build work does not disappear; it changes shape, and Chapter 8 puts a number on the change.

It also changes what the organisation owns. In the application era, the estate was the portfolio: a list of products, versions and renewals, owned mostly by suppliers. In the context era the estate is the data plane and the policy that governs it, and both are ownable in a way an interface never was. This is why the sovereignty question in Chapter 9 lands at board level, and why the asset most worth holding is the one most organisations have never inventoried.

## 4. Context is everything: the seven components

The phrase "context is everything" is used loosely in the industry. This paper uses it precisely, because a blueprint needs a definition that can be built, tested and audited. Context is the total set of facts a capable assistant needs to act on a person's behalf, and it has seven components.

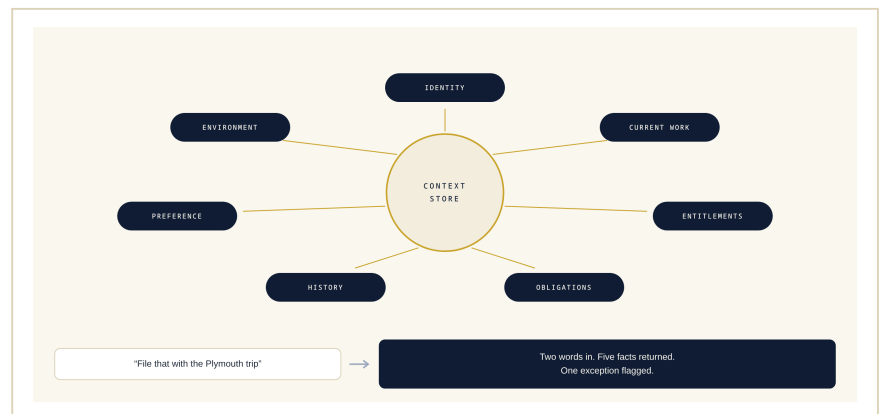


Figure 3. The seven components of the context layer, and the conversation they serve.

Identity and role. Who is asking: their position, their delegations, their approval limits, their manager, their location and their employment terms. Without this, an assistant is a search box with etiquette.

Current work. What the person is doing: projects, deadlines, meetings, commitments, travel. This is what makes "file that with the Plymouth trip" a complete instruction. The two words carry a week of meaning, but only if the context layer holds the week.

Entitlements. What data and actions the person may access, mapped to classification and role. Entitlements are the security boundary of the operating model: the agent plane can only act where context permits, and the permission must live in the context layer, not in the goodwill of a prompt.

- 1 Argument
- 2 Model
- 3 Controls
- 4 Money
- 5 Sovereignty
- 6 Path
- 7 References

## THE MODEL

4  
LAYERS, IN LOAD-BEARING ORDER  
Data, context, agent, human  
surface. Each is necessary; the  
order is not negotiable.

Obligations. The policies that bind the action: expenses rules, procurement thresholds, retention schedules, approval matrices, jurisdictional constraints. Obligations are the organisation speaking to the agent in the same conversation as the person.

History. Precedent and record: past claims, past decisions, past exceptions, the audit trail of what this person and this organisation did last time. History is what makes assistance consistent rather than improvised, and it is the component that makes auto compliance possible, because Chapter 7's evidence is written here.

Preference. How the person works: output format, level of detail, timing, tone, whether they want suggestions or decisions. Preference is a productivity multiplier and a governance afterthought in most organisations; in this model it is a first-class, correctable record.

Environment. Device, channel, calendar, time zone, connectivity. The same person in a car between sites needs a different surface to the same person at a desk, and the operating model should assemble it without being asked.

Worked example: the Plymouth trip, assembled. The expense claim of Chapter 2 exercised all seven components at once. Identity and role: the claimant is a consultant, with a per-head cap on client entertainment and a delegation to their practice lead for claims above their own approval limit. Current work: the Plymouth trip is a client engagement in the calendar, with attendees from both organisations and a project code. Entitlements: the claimant may see the trip record and the expenses policy; the agent may not see payroll. Obligations: the client meals rule, the alcohol cap, the standard class rule, the approval matrix. History: the last three claims, one returned for a missing receipt, so the agent attached the photograph before filing. Preference: the claimant files from a phone, in short sentences, and wants decisions queued for the train home rather than interrupting the meeting. Environment: a restaurant in Plymouth, on a mobile connection, twenty minutes before the next appointment. Two words in, five facts returned, one exception flagged. That ratio is the business case of this paper.

Two claims follow from the definition. First, context is assembled, not bought. Some components come from identity systems, some from mail and calendar, some from the business applications being reduced to stores, some from the person's own corrections. No vendor sells the assembled whole, and the assembly is the work. Second, context engineering supersedes prompt engineering as the discipline that matters. A well engineered prompt is one good sentence spoken to a brilliant stranger. A well engineered context is the difference between a stranger and a chief of staff. The industry's first instinct, longer and cleverer prompts, misses the point: the intelligence was never the missing part; the knowledge of the situation was.

One governance observation before the operating model is described. Every one of the seven components contains personal data about an employee, a client or both. The context layer is therefore the most concentrated personal data asset the organisation has ever assembled, and it is being assembled by accident in most organisations that are deploying AI assistants at

- 1 Argument
- 2 Model
- 3 Controls
- 4 Money
- 5 Sovereignty
- 6 Path
- 7 References

## THE MODEL

4

LAYERS, IN LOAD-BEARING ORDER

Data, context, agent, human surface. Each is necessary; the order is not negotiable.

scale. The organisation that treats the context layer as infrastructure, with an owner, a lawful basis, a retention rule and an audit log, converts that accident into an asset. The organisation that does not has built a liability, with surveillance capability attached.

## 5. The new operating model: stores, skills, agents, dashboards

The target operating model has four layers. Each is necessary; the order is load-bearing.

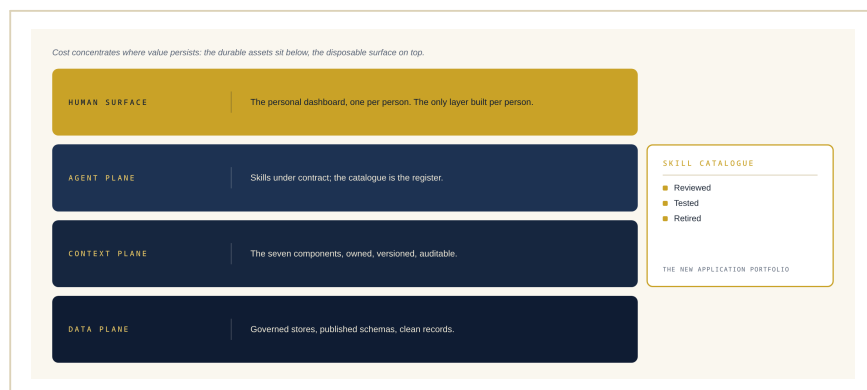


Figure 4. The four layers of the operating model. Each is necessary; the order is load-bearing.

The data plane. The business applications, reduced to what they always were underneath: governed data stores with published schemas and clean records. The disciplines of this plane are data quality, retention and classification, which most organisations will discover they have underinvested in precisely because interfaces used to compensate for bad data with patient humans. This plane is also where the application portfolio simplifies: when the interface no longer matters, products compete on data quality and integration, and rationalisation becomes possible for the first time in a decade.

The context plane. The seven components of Chapter 4, assembled into an owned, versioned, auditable asset: the context store. It has a schema, an owner, a lawful basis under the GDPR, a retention rule per component, and access controls at least as strict as the identity plane, because it subsumes the identity plane. Everything the business knows about a person and their work meets the agents here, and everything the agents do writes back here as history.

The agent plane. The unit of delivery is the skill: a named, versioned capability with a contract. The contract states what the skill does, what data classifications it may touch, what policies it must apply, what evidence it must write, what it costs per use, and who owns it. An agent is a composition of skills operating over context: file this expense, book this travel, draft this report from the meeting record, chase this approval, raise this requisition. The skill catalogue is the new application portfolio: reviewed for lifecycle like software, tested before release like software, and retired like software. Where AltLibre Paper No. 3 defined the governance stack for AI inside the IT department, this chapter is the same discipline generalised: nothing runs unregistered, everything runs tiered by risk, and the catalogue is

- 1 Argument
- 2 Model
- 3 Controls
- 4 Money
- 5 Sovereignty
- 6 Path
- 7 References

## THE CONTROLS

6

## DESIGN RULES GOVERN THE DASHBOARD

It is a surface, not a system of record: it renders context, it does not own it.

## 2 WORDS

## IN, FIVE FACTS RETURNED

One exception flagged. That is the whole interface.

the register.

Worked example: the contract for one skill. Name: file-expense, version 1.3. Function: file a client meal expense against a trip, with evidence attached. Data classifications: internal financial records; no personal data beyond the claimant's own. Policies consulted: the expenses policy, the client entertainment rule, the approval matrix. Evidence written: the claim record, the receipt image, the policy clauses consulted, the exception report. Cost per use: measured, and reviewed at each catalogue review. Owner: the finance operations manager. Human gate: claims above the claimant's approval limit route to the practice lead. Six fields and a gate: the entire specification of what the application era called an implementation.

The human surface. The personal dashboard: one per person, assembled from the context layer, described in Chapter 6. This is the only part of the operating model built per person, and it is the part that replaces the portal, the intranet, the eleven tabs and the two-day induction.

The economic logic of the four layers is that cost concentrates where value persists. Data and policy persist for decades; interfaces lasted for product cycles; training lasted until the next release. In the application era, the organisation's spend followed the interface: procurement, customisation, training, support. In the context era, spend follows the durable assets: the store, the policy, the context layer and the skill contract. The interface economy was a rental economy, in which the organisation paid continuously for translations. The context economy is an ownership economy, in which the organisation holds the assets and the translations stop.

## 6. The personal dashboard

Every person in the organisation gets a dashboard. Not a generic portal, and not a consumer chat window pasted over corporate data: a working surface assembled from their own context, showing what matters to them, doing what they ask, and declining where policy forbids. A chief of staff for every person, assembled from the context layer the organisation has already built.

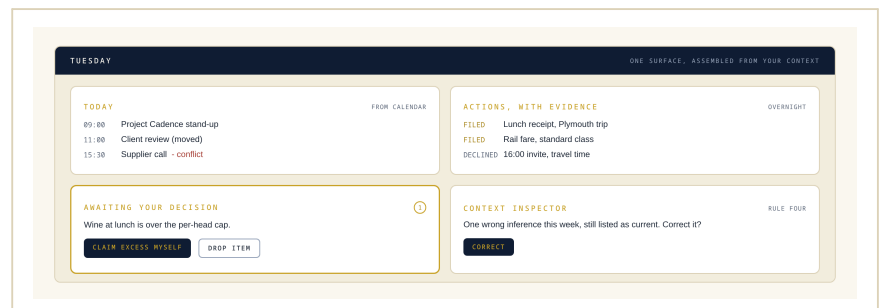


Figure 5. The personal dashboard. One surface per person, rendered from context the person can inspect.

- 1 Argument
- 2 Model
- 3 Controls
- 4 Money
- 5 Sovereignty
- 6 Path
- 7 References

## THE CONTROLS

5

LINES EVERY EVIDENCE RECORD  
CARRIES

What was asked, what was assembled, what was consulted, what was done, and what a human reviewed.

The dashboard is the operating model's most visible artefact and its most easily corrupted one, because it is where convenience and surveillance converge on the same screen. Six design rules govern it.

One: the dashboard is a surface, not a system of record. It renders context; it does not own truth. The record lives in the data plane, and any dashboard can be rebuilt from the stores and the context layer in hours. This rule is what prevents the operating model from re-creating the application era's most expensive mistake, the interface that became the estate.

Two: everything the assistant does is visible. The dashboard shows its own actions: what was filed, what was flagged, what was declined and under which policy clause. An assistant that acts invisibly is a liability in an operating model, whatever its productivity numbers.

Three: consequential actions are contestable. Any action with financial or personal consequence can be undone within a defined window, and the person can always see the decision that was made for them, by which skill, under which policy. This is the dashboard's implementation of the accountability principle from the EU AI Act and the GDPR: the human retains the power to overrule, and the power is real rather than nominal.

Four: the person sees the context, and can correct it. Preference and current work are wrong surprisingly often, because they were inferred. The dashboard lets the person inspect and amend what the assistant believes about them: their projects, their delegations, their defaults. This is both a data protection duty and a quality mechanism; the person is the cheapest auditor the context layer will ever have.

Five: no hidden differentiation. The context layer is capable of ranking, scoring and prioritising people, and therefore of quiet algorithmic management. Any differentiation between people that the dashboard performs must be declared, published and contestable, and employment decisions remain human decisions, with the dashboard's inputs available to the human who makes them. An assistant may schedule; it may not decide who is underperforming.

Six: symmetry of use. The context layer serves the person, not only the employer. If the organisation gains analytical visibility from its employees' context, the employees gain the same visibility in return, and any monitoring use is disclosed and lawful. The fastest way to kill the operating model is for staff to discover that the chief of staff was also a supervisor.

- 1 Argument
- 2 Model
- 3 Controls
- 4 Money
- 5 Sovereignty
- 6 Path
- 7 References

## THE CONTROLS

5

LINES EVERY EVIDENCE RECORD  
CARRIES

What was asked, what was assembled, what was consulted, what was done, and what a human reviewed.

Worked example: a Tuesday with the dashboard. 08:10 The dashboard assembles the day from the calendar: three meetings, one conflict flagged, a suggested swap sent for approval. 08:25 Overnight, the agent filed two receipts from yesterday's photographs, both under threshold, both shown in the actions feed with the evidence attached. 09:00 The stand-up runs from the dashboard's own summary; the project view is one screen, not eleven tabs. 11:40 A purchase requisition is drafted from the conversation in the meeting record; the person checks it, amends the quantity, approves. 14:05 The agent declines a meeting invitation, citing the travel time the calendar shows; the person would have accepted, and regretted it. 16:30 The context inspector shows one wrong inference from the week: a project the person left in March, still listed as current. Corrected in one click, logged as context quality work. Nothing on this list required training, and every consequential action carried a human decision.

Built this way, the personal dashboard stops being a feature and becomes the organisation's statement about what it believes work is: intent expressed in the person's own language, met by a system that knows the situation, bounded by policy everyone can read.

## 7. Auto compliance: evidence as a byproduct of action

Compliance in the application era was archaeology. The organisation acted through interfaces, then reconstructed what happened from logs designed for debugging, screenshots assembled by hand and interviews conducted under time pressure. The evidence was expensive, incomplete and late.

In the operating model this paper describes, evidence is a byproduct of action. Every skill is required, by its contract, to write an evidence record at the moment it acts: what was asked, by whom; what context was assembled, from which sources; which policy clauses were consulted; what was done, to which store; what a human reviewed and when. None of this is compiled afterwards, because all of it is generated while the work happens. The audit trail moves from sampling to completeness, and the compliance function moves from reconstruction to inspection.



Figure 6. The evidence record, written at the moment of action.

- 1 Argument
- 2 Model
- 3 Controls
- 4 Money
- 5 Sovereignty
- 6 Path
- 7 References

## THE MONEY

4  
LINES OF SPEND FALL, ONE  
RISES

Development, licences,  
integration and training fall;  
context stewardship rises.

Three legal instruments turn this from convenience to obligation. The GDPR's Article 22 restricts decisions made solely by automated means with legal or similarly significant effect, so the operating model requires a decision classification scheme: which decisions an agent may take alone, which require a human in the loop, and which a human must make without machine input at all. The EU AI Act requires transparency: people must know when they are interacting with an AI system, and deployers of high-risk systems carry duties of oversight, logging and information towards the workers affected. NIS2, where it applies, obliges the organisation to treat model and inference suppliers as part of its assessed supply chain, which means the agent plane's dependencies are auditable, named and jurisdictionally assessed. Paper No. 3 mapped these instruments to controls for the IT department; the mapping is identical here, with the scope widened to every process the agents touch.

Auto compliance also does quiet, structural work in the data protection layer. Labelling happens at the point of creation, because the skill that generates a document applies the organisation's classification scheme as it writes. Retention is applied by the store, not remembered by the employee. Records are complete, because the agent files as it acts rather than when someone remembers to. Most of the GDPR's operational pain is the gap between what the policy says and what the systems do; an operating model in which the policy is machine-readable and the agents must consult it closes that gap by construction.

One warning belongs here rather than in Chapter 10, because it is a design decision rather than a failure mode: evidence by design does not occur by accident. If the evidence requirement is not in the skill contract at version one, it will not be retrofitted at version four, and the organisation will have automated its processes before it automated the evidence of what they did. The skill contract in Chapter 5 exists principally for this reason.

## 8. The money: what the business stops spending

The operating model's economics are specific enough to enumerate. Four lines of spend fall, one rises, and the net is strongly positive for most mid-sized organisations.

Development spend falls. A new process in the application era was a build: screens, permissions, integrations, test cycles, training materials. A new process in the context era is a schema, a policy document and a skill. The organisation stops building translation layers, which is to say it stops building most of what it previously called development. It does not stop development; it stops building interfaces, a distinction the industry blurred for forty years.

Training spend falls, and mostly disappears. The training budget of the application era was a payment for the interface, as Chapter 1 established. When the ontology moves into the agent, the course has no subject. Induction does not disappear; it changes content, from "how to operate the expenses system" to "what the expenses policy says", which is the conversation the organisation should have been having all along.

|   |             |
|---|-------------|
| 1 | Argument    |
| 2 | Model       |
| 3 | Controls    |
| 4 | Money       |
| 5 | Sovereignty |
| 6 | Path        |
| 7 | References  |

## SOVEREIGNTY

## I

## QUESTION DECIDES THE RISK

Residency is not sovereignty:  
which legal regime can compel  
the assembled context?

Support spend falls. The first line of the application era's support desk was interface failure: users lost in menus, forms rejected, workflows stuck. Chapter 2 of Paper No. 3 described a first line substantially operated by AI inside the author's own department; the same pattern generalises. When there is no menu, there is no way to get lost in one, and exceptions are handled by the same agent plane that handles the work.

Administrative hours are returned to the business. The figures in Chapter 1 are the size of the prize: hours per day of manual digital administration per employee, work about work crowding out work. The context operating model does not recover all of it, but a mid-sized organisation that recovers even a third of the administrative time its people currently spend operating interfaces has reduced the headcount devoted to paperwork without touching a single job.

Governance spend rises, deliberately. Context engineering, data quality, skill testing, evidence review, model evaluation, sovereignty assessment: these are the new cost lines, and they are real. The accurate framing is that the savings are a transition dividend, not a permanent state: the organisation stops paying for translation and starts paying for trustworthiness. The net remains strongly positive, because trustworthiness is a bounded engineering cost, while translation was an unbounded annuity paid to whoever owned the interface, per user, per year, forever. The sovereignty question decides where much of the dividend accrues, which is the argument of the next chapter.

Worked example: the five lines of the spreadsheet. Line one, development: the translation-layer builds disappear; the schema, policy and skill work remains. Line two, training: the interface courses go; policy induction remains. Line three, support: the menu-walking first line goes; exceptions are handled by the agent plane at a fraction of the old cost. Line four, administrative hours: the figures of Chapter 1 are the prize; recovery starts with the pilot and is measured per participant. Line five, governance: context engineering, testing, evidence review and sovereignty assessment are new lines, entered deliberately. Four lines fall, one rises, and the transition dividend is the gap between them. The spreadsheet does not need invented figures to be read; every line names money the organisation already spends.

## 9. Who holds the context holds the business

The context layer is the organisation's working memory: its people's identities, work, entitlements, obligations, history and preferences, assembled in one place. That is precisely what makes it valuable, and precisely what makes its jurisdiction a board-level question, because Paper No. 1's analysis of AI dependency applies here with its force undiminished.

- 1 Argument
- 2 Model
- 3 Controls
- 4 Money
- 5 Sovereignty
- 6 Path
- 7 References

## THE PATH

7  
FAILURE MODES, IN ORDER  
The engineering section: what breaks, plainly stated.

Consider what a prompt contains in this operating model. Not a search query; the query "file that with the Plymouth trip" is harmless. But the context assembled around it is not: client relationships, project plans, personal data, commercial terms, the organisation's intent at the moment of action. When the assistant runs on a hosted model under a foreign provider, that context is processed inside a legal regime that includes the United States CLOUD Act, which permits US authorities to compel production from US-controlled providers regardless of where the data physically sits. Residency, as Paper No. 3 argued, is not sovereignty; a data centre in Dublin operated under a foreign legal entity is a data centre in Dublin operated under a foreign legal entity. In the application era, that argument applied to the IT estate. In the context era, the organisation's entire operating memory has acquired a jurisdiction, and most organisations deploying consumer AI assistants have not noticed that the border moved.

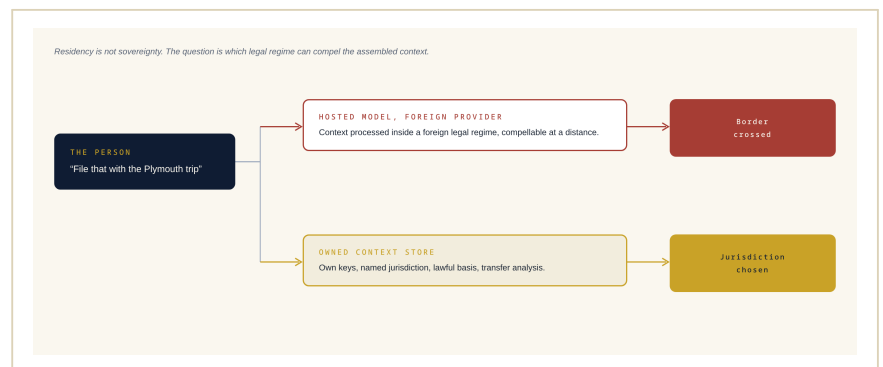
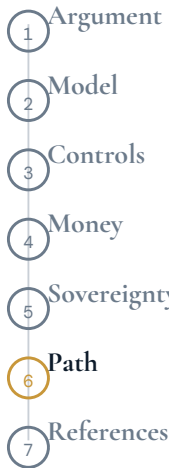


Figure 7. A two-word prompt crosses a border. Unless the context layer is an owned asset, the assembled context is processed inside a foreign legal regime.

The dependency compounds silently, for the same reason it did in the IT estate. The assistant is convenient; the assistant is adopted; the assistant's provider also supplies the mail platform, the identity platform and the productivity suite; and within eighteen months the organisation's collective memory is processed end-to-end inside a single foreign legal regime, with no exit that does not include institutional amnesia. Sovereignty washing makes detection harder, because the brochure mentions European residency while the control plane, the model weights and the legal entity sit elsewhere.

The remedy is the portfolio from Paper No. 3, applied to the business rather than the department. The context store itself must be an owned asset, under the organisation's own keys, in a jurisdiction the organisation's lawyers can name, with a lawful basis and a transfer analysis under the GDPR, because prompts and assembled contexts are personal data. The model portfolio follows the same tiers: open weights, self-hosted, for the sensitive domains; European jurisdiction providers for general workloads; a documented exception process for anything else. And the dependency map from Paper No. 3, extended once, covers this case: for each AI capability in the business, record where the context is assembled, where inference occurs, which authority can compel it, and what happens if the service is withdrawn.

The strategic formulation is short enough for a board pack. When the interface dissolves, the last visible border of the IT estate dissolves with it. What remains is context, and context has a jurisdiction. An organisation that does not choose its context layer's jurisdiction has already chosen it, and chose a foreign one.



## THE PATH

18

MONTHS, FOUR PHASES

Each phase closes with exit evidence, and each can be stopped without loss.

## 10. What breaks: failure modes, plainly stated

A blueprint that only lists benefits is a sales document. This chapter is the engineering section, and it describes the seven failure modes that will occur, in the order organisations will meet them.

Hallucination becomes action. In the assistant era, a fabricated sentence is an embarrassment. In the agent era, a fabricated transaction is a payment, a filing or a record. The mitigation is structural: agents propose consequential actions, evidence records accompany them, and a human approves within the decision classification scheme. Skills that act without that gate are scoped out of the catalogue.

The context layer becomes an attack surface. Whoever can inject into the context can steer the business: a poisoned calendar entry, a doctored document, a prompt injection riding inside an email the agent reads on your behalf. Prompt injection stops being a curiosity of chat interfaces and becomes an operating risk in the same class as malware. The context plane needs integrity controls, provenance on assembled context, and the same defensive engineering the data plane has received for twenty years.

Model drift changes the operating model overnight. A provider updates a hosted model and the skill that worked on Thursday behaves differently on Friday. The mitigation is the evaluation harness: skills are tested against fixed cases before release and periodically thereafter, and drift is detected in the catalogue rather than in the business.

Surveillance creep. The context layer that powers the chief of staff also powers employee monitoring, and the distance between the two is one procurement decision and a quiet configuration change. Design rule six in Chapter 6 exists because this is the failure mode that corrupts the model culturally: the day staff conclude that the assistant is management's instrument, the context layer degrades, because people will not correct a system they do not trust, and Chapter 4 established that the person is the cheapest auditor the context layer will ever have.

Skill sprawl. Every team builds skills; no one retires them; the catalogue becomes shadow IT with better grammar. The mitigation is the same one that governed applications: registration, tiering, ownership, lifecycle review. The catalogue is the register or it is nothing.

Loss of shared reality. When every dashboard is personal, the organisation risks losing the common picture that portals, for all their faults, accidentally provided. The mitigation is architectural: the record remains singular in the data plane, and every dashboard renders the same stores. Personal surfaces; shared truth.

Accountability diffusion. The phrase "the assistant did it" is not a defence in any statute this paper cites, and it will not become one. The decision classification scheme, the evidence record and the named skill owner exist to keep accountability legible. Where they fail,

- 1 Argument
- 2 Model
- 3 Controls
- 4 Money
- 5 Sovereignty
- 6 Path
- 7 References

## THE PATH

**18**

MONTHS, FOUR PHASES

Each phase closes with exit evidence, and each can be stopped without loss.

the operating model produces its most dangerous artefact: an organisation that acts without deciding.

None of these failure modes is a reason not to build the operating model. Each is a reason to build it deliberately, with the controls in Paper No. 3 generalised from the department to the business, rather than discovering the failure modes in production, which is the alternative.

## II. The eighteen-month migration path

The migration is sequenced so that each phase produces evidence, and each phase can be stopped without loss. The phases are described for a mid-sized organisation; scale the calendar, not the sequence.

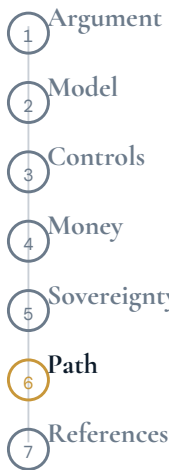


**Figure 8.** The eighteen-month migration. Four phases, each closed by exit evidence; the record migrates last.

Phase one, months one to three: inventory, and two processes. The organisation inventories its context sources: where identity, work, entitlements, policy and history live, and what state they are in. In parallel, two administrative processes are selected for migration, expenses and leave requests being the canonical pair, chosen because they are universal, low risk and universally disliked. Exit evidence: a context source inventory with owners; two processes selected with named business owners; a decision classification scheme drafted.

Phase two, months four to nine: the first skills live. The selected processes are rebuilt as schema, policy and skill. The context store is formalised with its schema, lawful basis, retention rules and access model. A pilot group of fifty people receives personal dashboards, and their corrections are captured as context quality work. Exit evidence: skills under contract in the catalogue, versioned; evidence records generated by default and reviewed by the compliance function; pilot satisfaction and defect rates measured; a data protection impact assessment for the context layer completed.

Phase three, months ten to fifteen: retire the first interfaces. The old portals for the migrated processes are withdrawn for the pilot population. The training materials for those processes are retired and replaced with the policy documents the agents consult. The



## THE PATH

## 6

## QUESTIONS FOR THE BOARD

If the answers are not known, the operating model is being built by accident.

capability catalogue is reviewed for the first time, and skills are tiered by risk per the governance stack of Paper No. 3. Exit evidence: zero training courses remaining for migrated processes; measured administrative hours returned per pilot participant; the first catalogue review completed.

Phase four, months sixteen to eighteen: the operating model becomes formal. The governance of the context layer and the skill catalogue is assigned to named owners, extending the governance stack from Paper No. 3 beyond the IT department. The sovereignty assessment of Chapter 9 is completed: where the context is assembled, where inference occurs, which authorities can compel it. Board reporting is established with the six questions of Chapter 12. Exit evidence: dependency map for all business AI capabilities; context layer under the organisation's own keys or a dated plan to that effect; the migration's second tranche of processes selected on the strength of the first tranche's numbers.

The sequencing embodies one principle: migrate the ceremony first, the record last. The portals, forms and training courses can be retired quickly because they were never the asset. The data plane and the context plane are the asset, and they are hardened, not replaced.

## 12. The questions the board should ask

Six questions, in the order they should be asked. If the answers are not known, the operating model is being built by accident, and accidents in this domain compound.

First: which of our training courses are compensating for our software? The answer is the list of processes to migrate first, and the size of the answer is the size of the opportunity.

Second: where does our context layer physically and legally live, and who can compel it? The follow-up, verbatim from Paper No. 3: name the jurisdictions. If the assembled context of the organisation's working memory is processed under a regime that permits extraterritorial compulsion, the board has a sovereignty exposure, and it is now a business risk rather than an IT one.

Third: who owns the skill catalogue? If the answer is no one, the organisation has an ungoverned software estate wearing a new name.

Fourth: what is our decision classification scheme? Which decisions may agents take alone, which require a human in the loop, and which a human must make without machine input? Under Article 22 of the GDPR and the EU AI Act, this is not optional, and it is the control that prevents accountability diffusion.

Fifth: can we show an evidence record for yesterday's automated actions? If the answer requires more than an afternoon, the compliance regime is still archaeology.

Sixth: what would it cost never to build another screen? This is the strategic question, because it forces the organisation to see the interface as a cost line rather than a capital asset. The organisation

- 1 Argument
- 2 Model
- 3 Controls
- 4 Money
- 5 Sovereignty
- 6 Path
- 7 References

## THE PATH

that internalises the answer stops buying translation layers, and starts owning the two things that were always worth owning: its data and the rules it has chosen to live by.

The closing argument is the same one this paper opened with, stated once more in its plainest form. For forty years, organisations taught their people to speak machine. The machines have learned to speak people. Everything that was built to manage the translation, the portals, the courses, the support queues, the fifteen to twenty per cent training tax, the hours of administration, was scaffolding for a compromise that has ended. What remains when the scaffolding comes down is the organisation itself: its data, its policy, its people's intent, and the context that connects them. The organisations that build deliberately around those four assets will run at a cost their competitors cannot follow, with a compliance position their competitors cannot fake. The rest will rent the same future from a foreign provider, and pay for the privilege in the old currency: their context, someone else's jurisdiction, forever.

## 13. References

---

1. Gartner, Inc., "Gartner Forecasts Worldwide IT Spending to Grow 14.2% in 2026, Totaling \$6.37 Trillion", press release, 27 July 2026; and Gartner, Inc., "Gartner Forecasts Worldwide IT Spending to Grow 9.8% in 2026, Exceeding \$6 Trillion For the First Time", press release, 22 October 2025.
2. Asana, Anatomy of Work Index, research series on time knowledge workers spend on "work about work", 2021 to 2023 editions.
3. Salesforce, State of Sales, research series on time sales representatives spend on manual data entry and administrative work.
4. Automation Anywhere, "Global Research Reveals World's Most Hated Office Tasks", OnePoll survey of 10,500 office workers across 11 countries, published 21 January 2020; fieldwork conducted autumn 2019.
5. Hollister, A., "The Sovereign Machine: A Blueprint for the AI-Enabled IT Department", AltLibre Papers, No. 3, 7 October 2026.
6. Hollister, A., "Your AI Answers to a Foreign Government: US and Chinese Law", AltLibre Papers, No. 1, 7 September 2026.
7. Regulation (EU) 2016/679 (GDPR), in particular Articles 13 to 15, 22, 25 and 35.
8. Regulation (EU) 2024/1689 (EU AI Act), in particular the transparency obligations applicable from 2 February 2025 and the deployer obligations for high-risk systems applicable from 2 August 2026.
9. Directive (EU) 2022/2555 (NIS2), Article 21(2)(d), supply chain security obligations.