OPERATING MODEL & DECISIONS

The Middle Nobody Owns

Christian Kroschl · 2026 · exact publication date unconfirmed

Article on kroschl.com

A supply chain programme can succeed on both halves and still fail as a whole. Strategy stops at the concept. System integration starts at the scope. Between them sits the question neither one answers: how this supply chain is actually steered. That question belongs to nobody, and it is where performance is won or lost.

Christian Kroschl · Twenty years in pharma and manufacturing networks

A manufacturer runs two programmes back to back. A strategy firm designs the target operating model: capability map, governance model, a roadmap in three waves. Solid work, signed off by the board. A system integrator then delivers the planning platform: scope agreed, data migrated, users trained, go live on schedule.

Eighteen months after go live, service performance sits where it sat before. Inventory is slightly higher. The planners maintain a set of spreadsheets alongside the new system, and the weekly escalation call is still the place where the real priorities get set.

Nobody failed. Both firms delivered exactly what they contracted to deliver. That is the part worth looking at.

Two halves that do not meet

The consulting market for supply chain work is split, and each half is internally coherent.

Strategy work ends at the concept. The artefacts are familiar: an operating model, a capability map, a governance design, a roadmap, a business case. They are useful, and they stop at exactly the point where the interesting questions begin, because the next layer down does not fit on a slide and does not survive a steering committee.

Notice what that sentence does not say. The problem is not the artefacts. An operating model is precisely the right object to build. The problem is where it is allowed to end.

System integration begins at the scope. It takes a defined solution design and makes it exist: modules, configuration, interfaces, data migration, cutover, hypercare. Also useful, and it starts from an assumption that somebody upstream has already decided what the system is supposed to do.

> *Neither half is wrong. Together they are incomplete, and the incompleteness has a specific address.*

See Figure 1 · page 6

Between the concept and the configuration sits a body of decisions that neither side owns. Not because either is negligent, but because these decisions are too detailed for the strategy phase and too consequential to be made during configuration. They fall through, and they get made anyway, by whoever is closest to the keyboard when the field needs a value.

Not a planning problem

There is a reflex to file all of this under planning, and it is worth resisting.

Planning and control is where the gap becomes visible first. It is the layer that produces a number every week, so it is the layer where the absence of a decision shows up as a missed date. That is why most of the examples below are drawn from there.

It is not the boundary of the problem. Steering a supply chain spans the planning and control mechanism, the execution feedback that tells you whether the plan survived contact with the floor,

the data model everything rests on, the system architecture that has to carry the decisions, the operating model and the governance that assigns them, the analytics and increasingly the AI layered on top, and whether any of it is adopted by the people who have to run it on a Tuesday morning.

Treat those as separate workstreams with separate owners and separate vendors and you do not remove the gap. You multiply it. Every seam between them becomes a smaller copy of the same problem.

> *The question is not who plans. The question is how this supply chain is steered, end to end, and who is accountable for the answer.*

See Figure 2 · page 7

What actually lives in the gap

This is not an abstraction. The gap has contents, and they are specific.

1. **Which decisions are taken where, on which horizon, by whom.** Not the org chart. The decision rights. Who may change a confirmed plan, inside which window, against which criteria, and who has to be told. 2. **Where the decoupling points sit.** Every decoupling decision is an inventory decision and a responsiveness decision at the same time. Move it upstream and you buy flexibility with working capital. Move it downstream and you buy cash with lead time. Somebody has to make that trade explicitly. 3. **What the time fences are, and what they cost.** A planning time fence is not an administrative date. It is a statement about how much responsiveness the business is buying and who pays for it in changeovers, expediting and broken promises to other customers. 4. **Which capacity number is real.** Nominal capacity is what the asset could produce under ideal conditions. Demonstrated capacity is what it has actually produced. Plans built on the first are approved with confidence and fail on contact. 5. **Which master data fields carry the plan, and who owns their accuracy.** A lead time set at go live and never recalibrated will misplan every night for years, and the organisation will quietly build workarounds around it rather than fix it. 6. **Whether the architecture can carry the decisions.** A decision that has no home in the data model does not survive the first release. Taking it in a workshop and hoping the platform can express it later is how design turns into decoration. 7. **What happens when plan and reality diverge.** Every supply chain has an answer to this. In most the answer is escalation, and nobody designed it. The feedback path from the floor back into the plan is a design object like any other, and it is usually the one nobody drew. 8. **Whether anyone actually operates it.** A steering model that lives in documents and not in Tuesday morning behaviour is not implemented, whatever the project status says.

Read that list again and notice what it is not. It is not strategy: no board approves a scrap factor. It is not configuration either: these are business decisions that must exist *before* anybody opens the system, and they determine what the configuration is for. And it is not planning: half of that list sits outside the planning department entirely.

Parameters are management decisions

The most expensive misunderstanding in supply chain transformation is that planning parameters are technical settings.

A safety stock value is a service promise with a price tag. A lot size rule is a decision about who absorbs variability, the plant or the customer. A planning time fence is a contract between sales and operations about how much chaos is affordable. A scheduling margin is an admission about how reliable a process really is.

All of these end up as numbers in fields. Because they look like configuration, they are treated as configuration, which means they are set once during implementation by someone who was never given the business intent behind them, and then never revisited. The management decision was made. It was just made by accident, by the wrong person, at the wrong moment, and nobody can name it afterwards.

> ***The test.*** *Pick any planning parameter in your system. Ask three questions: who decided this value, what business trade-off does it express, and when was it last checked against demonstrated performance? If the answers are missing*

for the first parameter you pick, they are missing for most of them. That is not a data problem. It is an ownership problem wearing a data costume.

The evidence is not where the attention is

Here is a number that should reorganise a lot of transformation budgets.

Take a material with an eight week lead time, demand varying by about thirty percent period to period, and a supplier whose lead time varies by around two weeks. Put those into the standard safety stock formula. A perfect forecast removes roughly eight percent of the buffer. A lead time you can rely on removes roughly sixty percent. Change the parameters and the split moves, and if the supply is already reliable it reverses. But across the lead times and supply variability this industry actually runs, reliability is the larger lever far more often than not.

Forecast accuracy is where the attention goes, the software is sold and the analytics team is staffed. Lead time reliability is where the leverage is. The gap between those two facts is not a knowledge problem. Most planners know it. It persists because forecast accuracy is a project you can buy, and lead time reliability is a set of decisions somebody has to own across functions that do not report to each other.

The same asymmetry shows up in rhythm. Run a plant to a repeating pattern and a customer order can ship the same day on roughly ninety three percent of days. Run the same plant reactively, same demand and same capacity, and that falls to about forty five. Nothing physical changed. What changed is whether the sequence is a decision or a reaction.

None of this should be taken on trust. Each of the simulators on the main page isolates one of these trade-offs and lets you push it until it breaks: the Safety Stock Synthesizer for which side of the formula is setting your buffer, the Lead Time Cascade for one set of lead times and two consequences, the Pattern Wheel for rhythm against reaction, and the S&OP Simulation Suite for one decision and three bills.

Escalation is a functioning system

When the logic in the middle is not owned, the organisation does not stop. It falls back on escalation, and escalation works. Orders ship. Customers are served. The quarter closes.

What makes escalation expensive is not that it fails. It is that it succeeds while hiding four things at once. It consumes senior attention as its fuel, so the cost never appears in any budget line. It produces no learning, because each intervention is treated as an exception rather than as evidence about the design. It makes the supply chain look robust right up to the moment the two or three people holding it together are unavailable. And it removes the pressure that would otherwise force the underlying decision to be made properly.

A supply chain run by escalation and a supply chain run by design can post the same service number for a long time. They are not the same asset. One of them degrades quietly the moment attention moves elsewhere.

What closing the gap actually requires

Not a third flavour of consulting. A different unit of work.

The deliverable is the operating logic itself, designed as an artefact that survives contact with the system. Concretely, that means three things hold together at once.

Design decisions trace to parameters. Every choice in the operating model has to land somewhere specific: a field, a rule, a horizon, a threshold. If a design decision cannot be traced to the place where it takes effect, it is a preference, not a design.

Parameters trace to a named owner. Not a function. A person, with the authority to change the value and the accountability for what the value produces. Ownership that stops at team level is how a value ends up correct in the document and wrong in the system.

Performance traces back to the decision. When service fails, the organisation should be able to read the failure backwards through execution, plan and data to the decision that caused it. Most

performance reviews stop at the customer facing number, which is the last place a problem appears rather than the first.

See Figure 3 · page 8

There is one more requirement, and it is the one that most delivery models get wrong. The people who design the logic have to be the same people who can sit in the configuration workshop and defend it. The moment design and realisation are split across two organisations, the gap reopens in the handover, which is precisely where it opened the first time.

That is not an argument for a specialist of the messy middle. It is an argument against the seam.

See Figure 4 · page 9

Call it engineering

There is a name for the work in the middle, and it is not consulting.

An engineer does not hand over a concept and does not hand over a configuration. An engineer delivers a system that holds under load. That assumption changes everything upstream of it. You design for the disturbance rather than the ideal case. You specify tolerances rather than intentions. You build the feedback path in rather than bolting it on afterwards. You decide the failure modes deliberately instead of discovering them in month nine.

It also changes what counts as finished. An engineered system is commissioned, not delivered. You run it up, you measure it against design intent, and you correct whatever the real world disagreed with. A supply chain that has been designed but never commissioned is the eighteen month story at the top of this piece.

That is what the three requirements above actually are: specification, ownership and test, applied to a supply chain instead of a machine. Very little about that is metaphorical. A planning parameter is a component with a specification and a tolerance. An escalation path is an undocumented failure mode. A master data field with no owner is a part with no drawing. A go live without commissioning is a machine switched on and left alone.

> *Supply chain engineering: designing how a supply chain is steered, and building it until it runs.*

The word matters because it sets the standard of proof. Advice is judged by whether it is convincing. Engineering is judged by whether the thing works.

How we work

We start from how the supply chain is to be steered. Not from the tool scope, and not from the capability map. The first question is which decisions exist, on which horizon, with which information, by whom, and how the answer travels through processes, data, roles and systems until something physically moves.

Planning and control is the mechanism we use most, because it is where steering becomes measurable. It is not the identity of the work and it is not the limit of the scope.

We treat parameters as management decisions and we make people own them by name. We build the measurement so it reads bottom up, from data quality through plan quality to execution and only then to service, because that is the order in which failures actually occur.

And we stay until it runs. Design that is handed over at the point of maximum ambiguity is not design, it is an opinion with a cover page.

> *We are not selling slides and we are not configuring tools in isolation. We design how a supply chain is steered, and we make it executable in processes, data, roles and systems.*

What that means in scope

This is not a claim to a niche between two camps.

We build the operating model, and in the planning environment specifically: S&OP and S&OE, the decision forums, the horizons, the roles. We write the governance that says who decides what and where. We build the roadmap that sequences it. And we do the system realisation, which means we can sit in the configuration workshop and argue about a planning parameter with the person who will have to maintain it for the next five years.

Not because a large scope is attractive. Because the join is where the logic leaks out. Every handover you introduce is a place where an intent becomes a specification, a specification becomes a ticket, and a ticket becomes a setting nobody can explain eighteen months later.

The honest limits are elsewhere. Large strategy firms have more people in more boardrooms than we do. Large integrators can staff a global rollout at a scale we cannot. Those are real differences and we do not pretend otherwise. What we do not do is hand over the design halfway and hope it survives the trip.

> *The point is not that we occupy the middle. The point is that the middle stops existing when one team carries the work from the operating model to the running system.*

Five questions for your own supply chain

None of this needs a diagnostic project. Ask these in your next planning review and listen to how long the answers take.

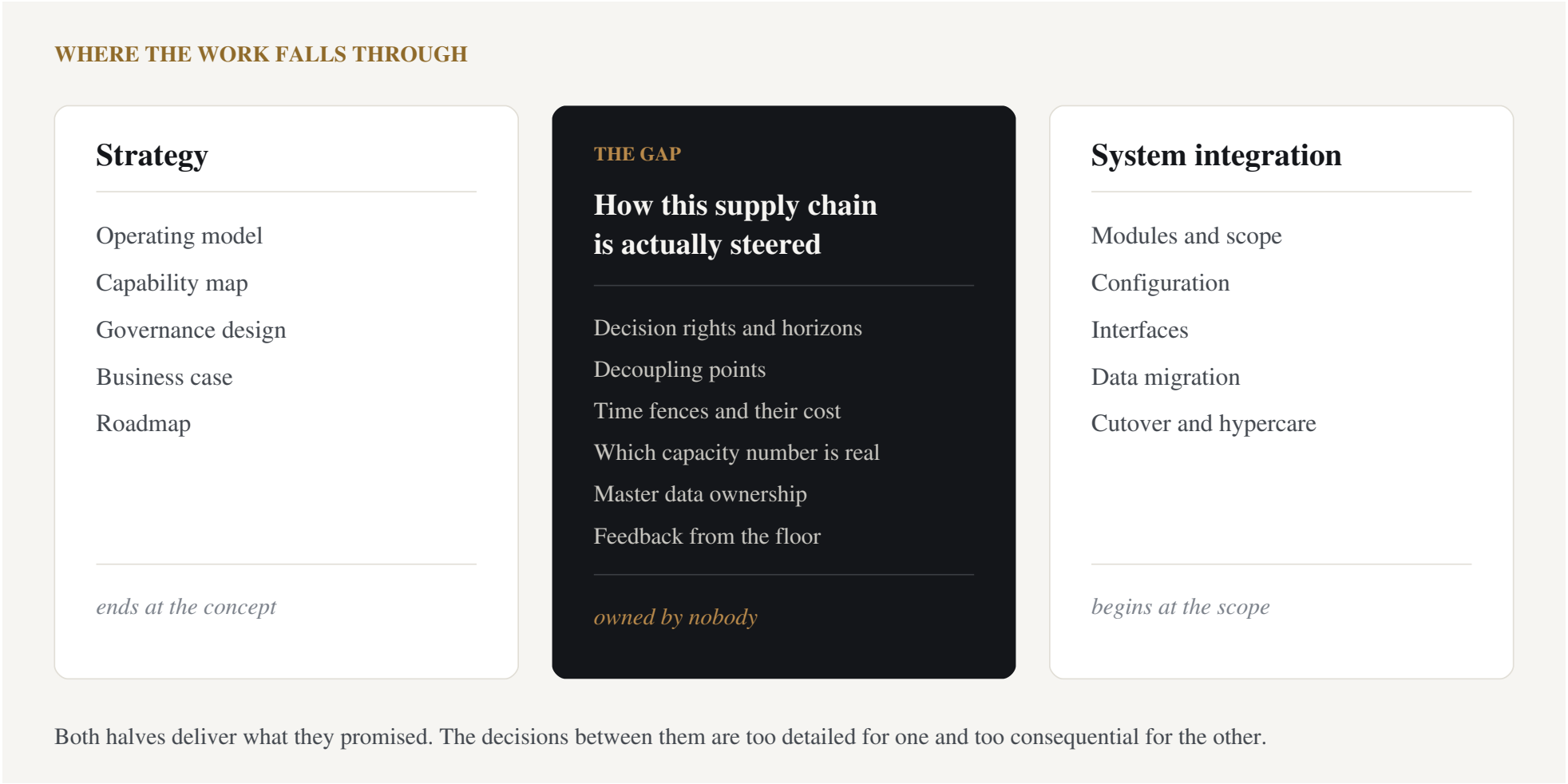
1. Is there one plan, and can everyone in the room name where it lives? 2. Would a capacity gap identified today reach a management decision this month, or would it wait for the exception that forces it? 3. When a delivery fails, does the organisation trace it back to demand, data, plan or execution, or does it assign blame? 4. Who owns master data accuracy by name, and what is their measured performance? 5. What holds this together when the two people who always fix it are both on holiday?

If those questions are uncomfortable, the discomfort is the finding. The work is not to answer them once. It is to build a system in which they have obvious answers, and to keep it that way after the programme energy is gone.

There is a model behind all of this: six principles, a decision chain from structural design through execution feedback, a control layer of architecture, policy, people and proof, and the measurement that tells you which link broke. It is written up in full, with worked examples and four simulations, as Integrated Planning & Control.

July 2026. If you want to argue with any of it, reach me on LinkedIn.

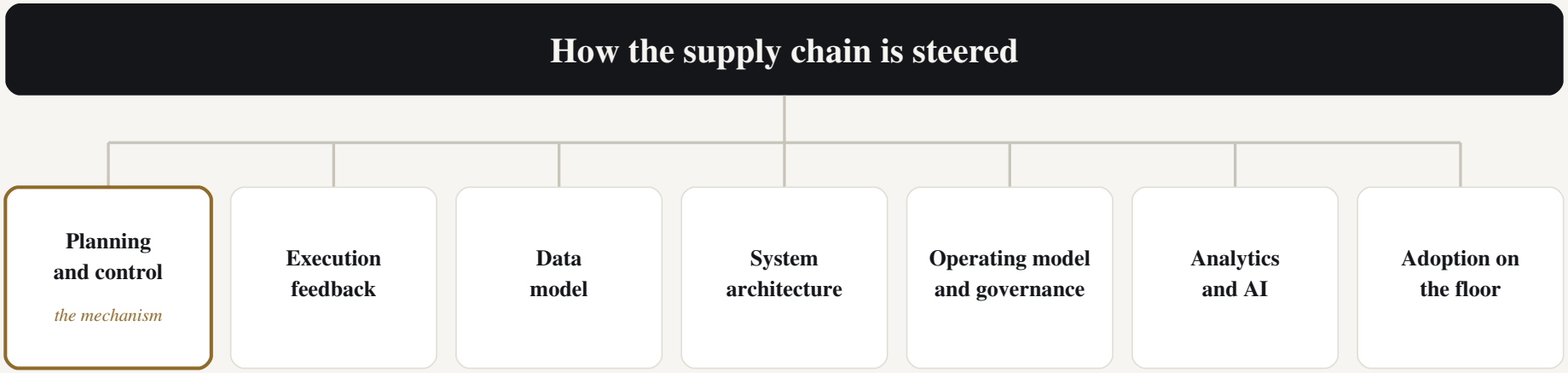
Figure 1



Back to text · page 1

Figure 2

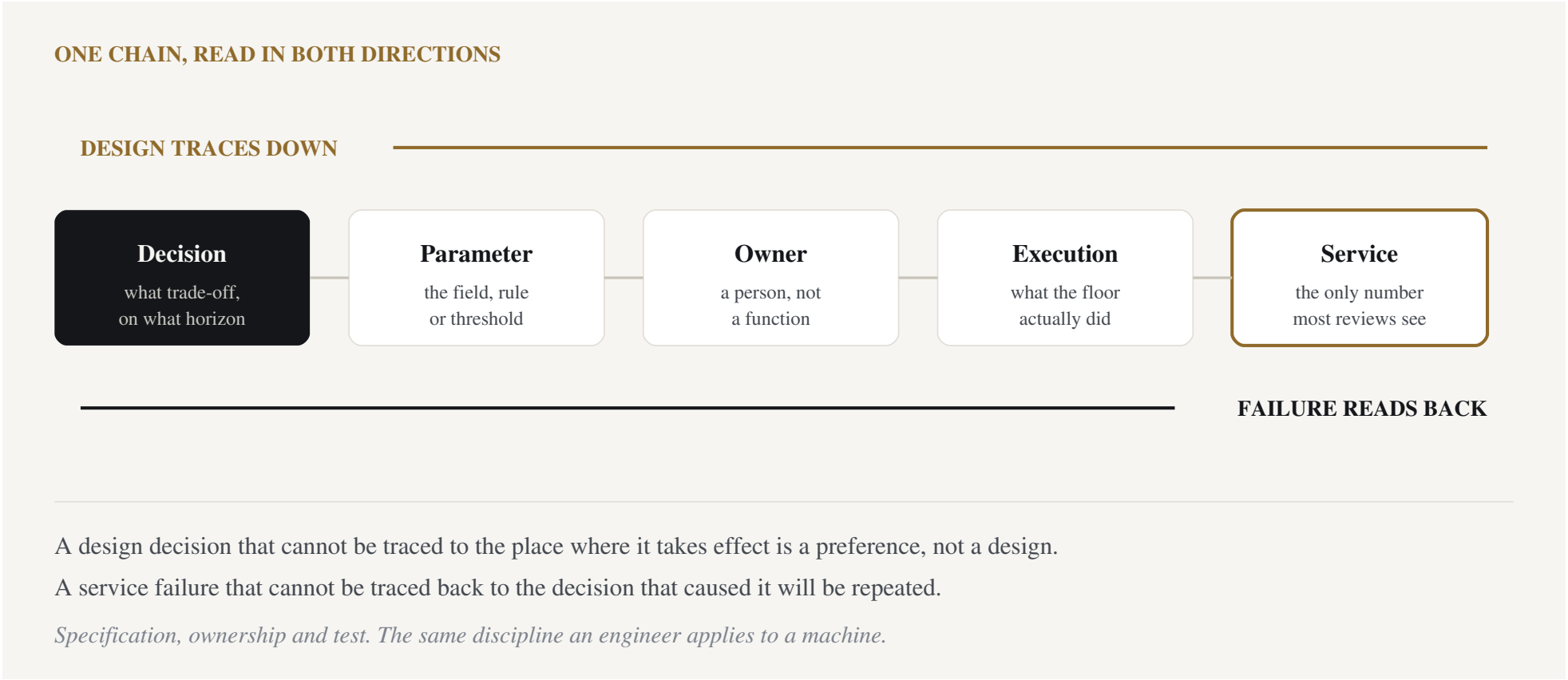
PLANNING IS THE MECHANISM, NOT THE SCOPE



Split these across separate owners, separate budgets and separate vendors and you do not remove the gap.
Every seam between them becomes a smaller copy of it.

Back to text · page 2

Figure 3



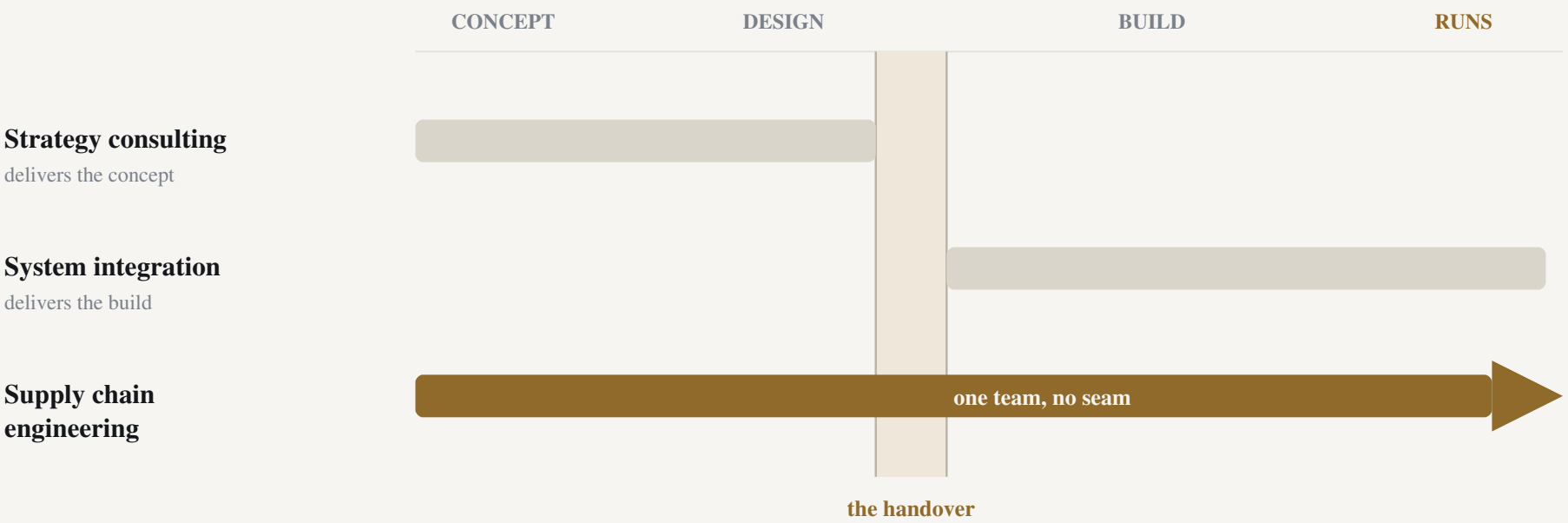
A design decision that cannot be traced to the place where it takes effect is a preference, not a design.

A service failure that cannot be traced back to the decision that caused it will be repeated.

Specification, ownership and test. The same discipline an engineer applies to a machine.

Back to text · page 4

Figure 4



We build the operating model and we build the system.
Not because the whole scope is attractive, but because the join is where the logic leaks out.

Back to text · page 4