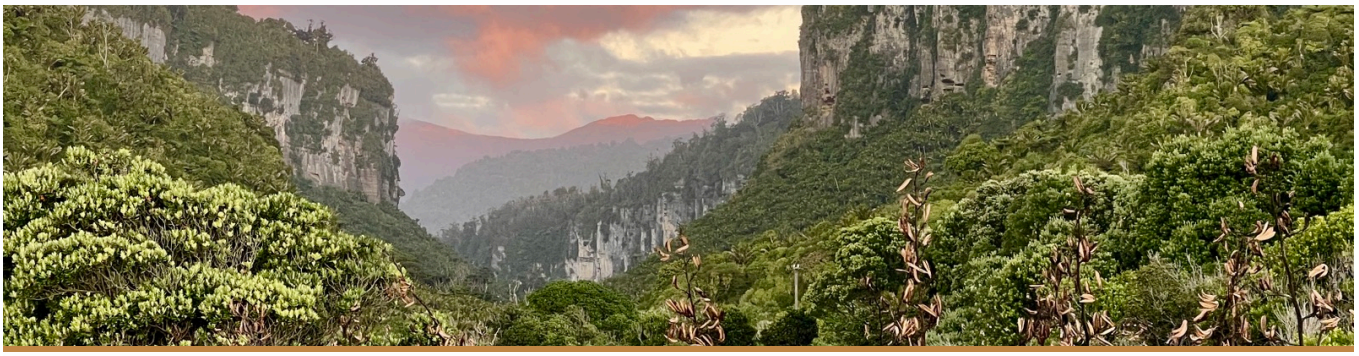


Running an organisation where agents do the work

North Canterbury · © John Stroh

Version 0.4 · revised 21 August 2026 · this version as a PDF

> Drafted with AI assistance, then checked and revised by the author. The judgements and the errors are the author's.

Status of these claims

What this publication does not claim, and what is outstanding against it in the register.

Quoted research without a source record: 5 passages in this publication are quoted verbatim from external research and carry a citation marker that resolves to nothing in this corpus. The quotations are reproduced as written; the sources are not recorded here, so treat the attributions as unverified until they are.

Structure, records and drawings for an organisation whose work is mostly done by software that acts on its own — and the failures that shaped them.

Series part E · Version 0.4 · August 2026

Before you read this

You are going to build something where most of the work is done by software that acts on its own, and you will still be accountable for all of it. Four questions arrive early, and none has an obvious answer.

Who is responsible for what an agent did? Not in principle — in the specific case, at three in the afternoon, when someone asks. Conventional structure answers this by asking a manager. That stops working when agents are making decisions faster than anyone can be told about them.

How do you keep your own claims straight? You will make promises to members, buyers and regulators, and those promises will change. Most organisations end up holding several versions of every claim with no way to tell which one is current.

How do you stop the diagrams lying? Nobody re-reads a diagram when they edit a paragraph. Drawings drift from text faster than text drifts from itself.

And what do you deliberately do badly? An organisation tuned for efficiency is brittle in ways that only appear under stress. Some inefficiency has to be chosen, written down, and defended against people who will always have a good argument for removing it.

This part answers those four. It also records where the answers failed in practice — because a method that reports only its successes is not reporting.

What this is for

Conventional organisational structure will not fit. It assumes that finding out what happened is expensive, that a layer of people exists mainly to gather and verify status, and that accountability is established by asking someone. None of those assumptions survives an organisation where every action is already recorded.

This part stands on its own. The rest of the series describes shared infrastructure a group of organisations might build together, and this was written for whoever runs it. But nothing below depends on that: the problem — you are accountable for work done by software you did not watch — arrives for any organisation deploying agents, whoever owns the infrastructure and however it is constituted.

Four things follow, and one principle runs through all of them: **the boundary you manage should be a boundary the record already has.**

Take what is useful. Argue with the rest.

Part one · Structure

FIG-11 THE BOUNDARY YOU MANAGE IS A BOUNDARY THE RECORD ALREADY HAS

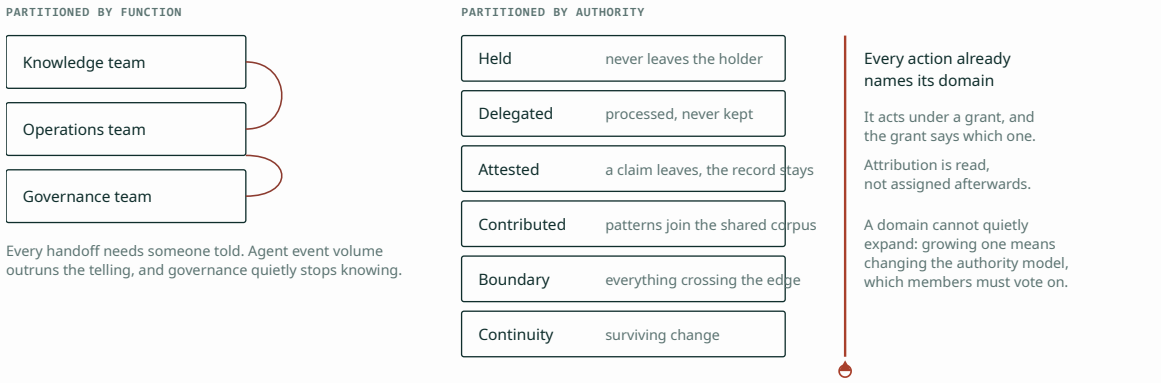


FIG-11 Functional partitions require operations to tell governance what happened, and agent event volume outruns the telling. Authority partitions put the boundary where the record already has one.

Why the obvious partition fails

The instinct is to organise by function: a knowledge team, an operations team, a governance team. Research on agent oversight in public-sector organisations finds a specific reason this fails at scale.

Continuous oversight cannot be guaranteed by segmented structures that isolate governance requirements in separate teams, or by processes that are event-triggered, because the frequency of events produced by agents exponentiates communication costs between operational and governance units — costs that are already prohibitive. Governance responsibilities must therefore be diffused toward the end users: the implementing operational departments whose work is augmented by agents.

Put plainly: agents generate more events than a governance function can be told about. Any structure that requires operations to inform governance will be overwhelmed, quietly, and the first sign will be that governance has stopped knowing things.

But the same research warns against the opposite error. *Some degree of central oversight must remain, because workers taking on dual roles as overseers are themselves influenced by the technology in the exercise of their own discretion.*

So governance cannot be a department and cannot be entirely dissolved into operations. Something has to carry it that is neither.

And why copying a human structure fails

There is a second finding, and it is the sharper one.

Agentic behaviour is sustained by context architecture rather than by human motivation, identity and trust. The implication is that human-imitation organisational forms underperform when they add handoffs without preserving evidence or reducing drift.

Human organisations tolerate lossy handoffs because people reconstruct context by asking each other. Agents do not. **A handoff that does not carry evidence is pure cost**, and an organisation designed by analogy to a human one will be full of them.

That gives the design rule: every boundary in the organisation must be a boundary the record already carries.

The partition: authority, not function

A steward's domain is **a class of authority over records** rather than a business function.

DOMAIN	AUTHORITY HELD	ACCOUNTABLE FOR
Held	Records that never leave the holder's control	That the floor holds — that no path exists by which a held record moves
Delegated	Processing without retention	Destruction attested rather than asserted; processing contexts; agent versioning
Attested	Claims that leave when records do not	Claim accuracy, revocation, and the ceiling on what claims may be composed into
Contributed	The shared corpus	Abstraction quality, source traceability, re-derivation within the published bound
Boundary	Everything crossing the edge	Egress control, supply-chain discovery, inference-path register, jurisdiction review
Continuity	Surviving change	Substitution times, drills, heterogeneity, succession, the body of resolved cases

Two properties follow that a functional partition cannot give you.

Attribution is read, not assigned. Every agentic action operates under an authority grant, and the grant names the domain. Nobody decides after the fact whose responsibility something was.

A domain cannot quietly expand. Functional domains grow by absorbing adjacent work — that is how organisations acquire a fourth layer without anyone deciding to. An authority domain can only grow if the authority model changes — and that should be a decision reserved

to whoever owns the organisation, not to whoever runs it.

Names are yours to choose. Six is what the authority model produced; a different authority model produces a different number, and if yours produces three you have either a simpler system or an incomplete one.

The way this fails. The domains and the authority model have to stay aligned, and nothing forces them to. Change the authority model without moving the domains and you have an organisational chart describing a system that no longer exists — accountability allocated for a boundary that has moved. Nothing announces it. Declare which domains depend on which parts of the authority model, and check the two against each other whenever either changes.

Three layers, and what pays for the third one's absence

Board. Domain stewards. The people doing the work. No fourth layer under any name — no programme office, no deputy stewardship.

This is a consequence rather than a preference. Where every action is signed and timestamped, the ledger performs continuously the function that a middle layer performed periodically. The layer whose main product was status has had its product automated.

But that layer did something else as well, and this is the part most flat-structure arguments miss. **Middle management is also redundancy.** It absorbs disagreement, holds informal knowledge, and provides somewhere for an objection to go that is not the person you are objecting to. Removing it for throughput and replacing nothing produces an organisation that is coherent in the way that fails suddenly.

Two replacements, and both cost something:

Dissent has a structural route. A steward may formally object within or adjacent to their domain. The objection is recorded, answered in writing by the board, and both stand permanently. Not a veto and not a delay — the decision proceeds. What changes is that the disagreement survives the decision, so that when a problem is later identified there is a record of who saw it coming.

A sampled fraction of routine work is checked by a person, always. Not only escalations. Where people see exceptions alone, they lose the calibration required to judge an exception, and the exception path degrades precisely as the routine path improves. The sample rate has a published floor, and **it may not fall as accuracy rises** — that is switching off the alarm because it has stopped sounding.

Roles inside a domain

Practice is converging on four supervisory roles: *Agent Supervisor, Eval Owner, Exception Handler and Human-in-the-Loop Reviewer — held even where a single person holds several of them in a small organisation, with the owner being a named human in the function that consumes the agent's output rather than IT or the vendor.*

Here those are roles held *inside* a domain, not posts beside it. The steward is the named owner; the four roles distribute across the people in the domain and collapse into one person in a small cluster. Three layers stay intact and accountability stays granular.

One human can be accountable for a much larger volume of output, but only if the structure clearly defines what they are watching and what they are allowed to ignore. The authority grant is what defines that. It is the job description, and it is machine-readable.

Advancement, and pay

No vertical ladder exists, so advancement cannot be promotion. It is competency-based: the same person handles more demanding work as attested competency grows, with agentic support scaling to match, and pay following competency rather than headcount managed. Attestations live in the record, so competency is evidenced rather than claimed.

Pay shape. A strong base — enough that no one is negotiating from need — with competency-scaled additions, and no significant spread across the three layers. Board, steward and coal face are different accountabilities, not different classes.

This is a deliberate departure and should be argued as one. The conventional justification for a wide spread is that senior roles carry more leverage over outcomes. In an organisation where the leverage sits in the authority model and the ledger, that justification is weaker than it looks.

Part two · How the organisation holds its own knowledge

FIG-10 THE LEDGER REPLACED THE LAYER THAT REPORTED STATUS

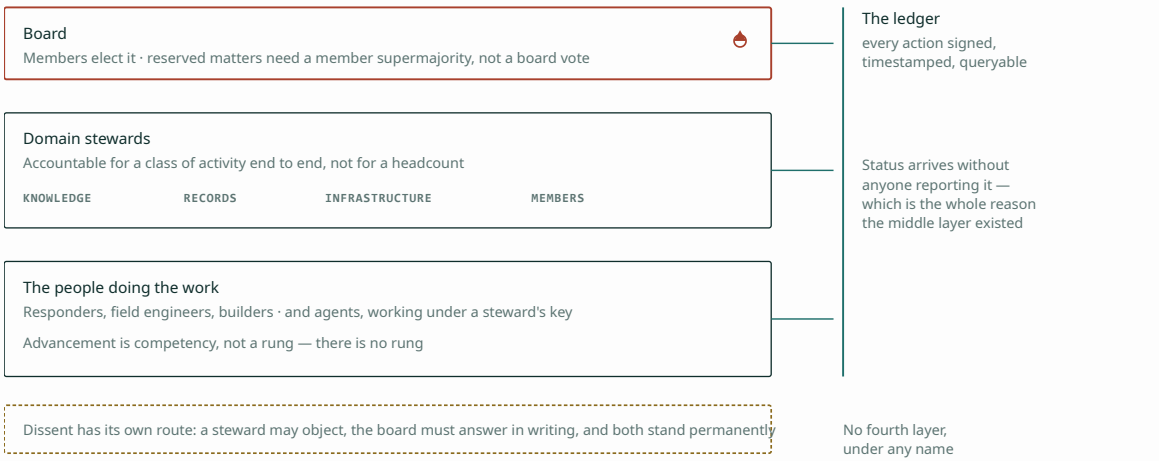


FIG-10 Middle management exists largely to aggregate and verify status. Where every action is signed and timestamped, the ledger does that continuously — so the flat structure is a consequence of the record architecture, not a preference.

Structure decided, the next problem is that this organisation makes claims — to members, to buyers, to regulators — and has to keep them straight while they change.

The conventional answer is a document set: a strategy, a business case, a rollout plan, an operations manual. It fails in a specific way. Each downstream document restates upstream context so it can stand alone, the restatements drift, and within two revisions you are maintaining several versions of every claim with no way to tell which is current.

One corpus, many renderings

The unit is the **assertion**, not the document. One claim, one file, one identifier, one accountable steward. A document is a *rendering* — a query over assertions, ordered for an audience. Nothing is written twice, so nothing can drift.

Each assertion carries:

- A stable identifier, never renumbered, never reused
- One steward — unowned assertions rot
- **Typed links**, which is what makes the corpus queryable
- A review sunset, so authority is renewed rather than revoked
- A disclosure level
- Evidence links, or a visible mark that there are none

A plain hyperlink says two things are related. A typed link says *how*: this is authorised by that; this measurement supports that claim; this procedure realises that rule; this limits what that may assert; **this contradicts that.**

That last one earns its place. Every organisation holds claims in tension. Most discover it in front of a reviewer. Recording it makes the tension a managed item with an owner rather than an ambush.

Two queries that replace remembering

Staleness. A publication declares which assertions it renders. When one of them changes after the publication was last edited, the publication is flagged. The prose no longer says what the organisation says.

Coverage. Which assertions are carried by no publication and no drawing? This is the query that catches work being lost — an insight that exists in the record and reaches nobody. It is the one to run before declaring any revision complete.

What it caught

The claim that this works is only worth making with evidence.

- A business-case line asserting costs that had never been measured. Tagged as requiring evidence, no evidence attached, reported on every run. **It is still failing on purpose**, because the alternative is a document where unmeasured figures look identical to measured ones.
- A constitutional clause discussed at length in a publication that never declared it, so coverage still reported it as reaching no reader. The prose changed and the declaration did not.
- An amendment-consequence query: change a clause, get back every assertion that would need review, with each one's steward. Members can vote on a change knowing what it breaks.
- A drawing showing five items where the text had moved to six. Caught by a reviewer, not by the tool — see below.

What it failed to catch

Staleness masked by cosmetic edits. Figures declared what they depicted and the check compared commit dates. But a theme synchronisation rewrote every drawing file, resetting every date, silently clearing every staleness flag in the system. A tool that reports "nothing is stale" because something unrelated touched every file is worse than no tool: it produces confidence without checking.

A gap in the disclosure gate. The check verified that no *publication* referenced restricted material. Drawings were never checked, and drawings go into published decks directly. A restricted specification was referenced by a drawing that appeared in two decks, and nothing objected.

Caveat carriage blind across one axis. Where a drawing exists in a plain-language version, the tooling enforces that every caveat is carried into it — simplification may not drop what makes a claim honest. But drawings also have progressive layers, and truncating to two layers silently dropped a caveat that lived in layer four. The rule was enforced on one axis and blind on the other.

The lesson in all three is the same, and it is the reason this section exists: **checking machinery creates confidence out of proportion to what it actually verifies.** Every check should be tested against a deliberately planted defect. A check that has never failed has never been shown to work.

Part three · How the organisation draws

Drawings drift from text faster than text drifts from itself, because nobody re-reads a diagram when they edit a paragraph. Four rules keep them honest.

Geometry is the argument; words are the register. One drawing per idea, carrying the structure, with text keyed rather than baked in. A plain-language pack overrides the words without touching the drawing. If a version needs different geometry, the two versions are making different arguments, and that is a finding about the argument.

Simplification may not drop caveats. A language pack declares which keys are caveats, and the check fails if one falls back to the technical version. Without that rule, “plain language” quietly becomes “the version without the uncomfortable sentence.”

Colour carries meaning, not decoration. One colour marks only what cannot be moved — constitutional commitments, fixed boundaries, sealed records. Another marks what travels. A third marks what is unresolved and parked. If the fixed-point colour starts appearing on things that are merely important, the scheme has stopped carrying information and should be reset.

A drawing declares what it depicts. Which claims does this illustrate? Recorded in a manifest, so a claim changing makes the drawing checkable rather than quietly wrong.

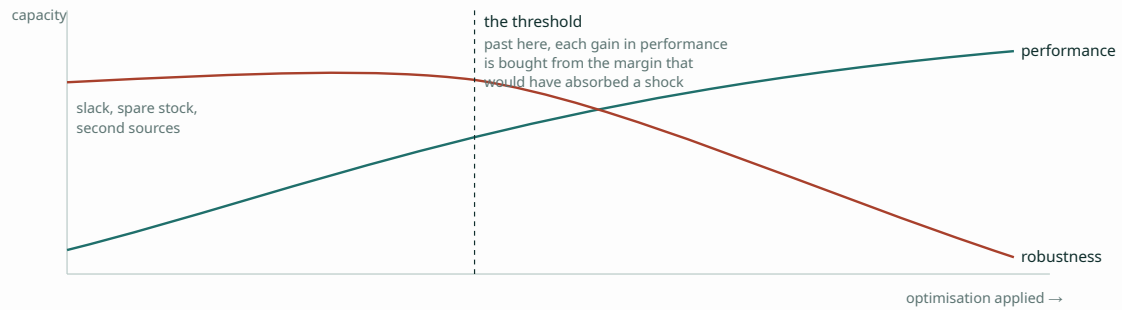
Two practical notes. Drawings are fixed-width bands with transparent backgrounds, so they stack into any combination without adjustment — a set for a minister and a set for a board draw from the same file. And **drawings are made alongside the prose, not after it.** Made

after, they decorate. Made alongside, they shape the argument: the five-versus-six error above was caught because someone had to draw the row that did not exist.

Photographs, if used, are a separate matter. One per publication, from your own collection, credited with place and photographer, never captioned with a line telling the reader how to feel. Measure the photograph before placing it: a warm image sitting beside a status colour in the same hue family destroys the meaning of the status colour.

Part four · What the organisation gives up on purpose

FIG-03 COUNTERPRODUCTIVITY · SCHEMATIC, NOT MEASURED



Shape, not magnitude. The threshold position is illustrative — where it sits for any given system is an empirical question we have not answered.

FIG-03 Past a threshold, each increment of performance is bought from the margin that would have absorbed a shock. Nothing announces the crossing. Schematic: the shape is the claim, not the position of the threshold.

An organisation optimised for efficiency is brittle in ways that appear only under stress. The vocabulary is worth stating because a performance culture and a robustness culture use the same words for different things.

A PERFORMANCE CULTURE CALLS THIS	A ROBUST ORGANISATION CALLS IT
Idle capacity	Margin — what absorbs a shock nobody modelled
Duplication	Redundancy — more than one of what matters
Inconsistency between sites	Heterogeneity — parts that do not all fail the same way
Friction	Slowness — time for a correction to land
Internal disagreement	Incoherence — dissent that has not been optimised away
A single best supplier	A single point of failure

The weak link inverts. In a stable world the weak link in a chain is the least capable supplier, because it slows everyone down. Under fluctuation it is the *most* capable one, because it is the most optimised and therefore the most brittle. Dependency on the fastest, cheapest, most capable supplier arrives on the procurement form looking like prudence.

Past a threshold, performance is bought from margin. The system keeps improving at what it was tuned for while losing the capacity to absorb anything it was not tuned for. Nothing announces the crossing.

Four commitments make an organisation deliberately less efficient, and each should be structural rather than intentional — written where management must persuade the owners rather than simply decide:

1. **Every site holds complete state.** More expensive than centralising, and the premium appears as a defended line rather than absorbed into overhead. A guarantee optimised into overhead is unfunded and therefore false.
2. **Hardware and models differ across sites.** Standardising is cheaper and produces correlated failure. The reference is not the optimised fighter but the airliner flying at half capacity with three independent autopilots.
3. **A person checks a sample of routine work, always**, with a published floor that may not fall as accuracy rises.
4. **Dissent is preserved rather than resolved away.**

Each will come under pressure. The pressure will always be well argued and it will always arrive as a proposal to reduce cost per site. Naming the shape of the erosion in advance is most of the defence against it.

The question test

A robust question stays stable when you shake it with interactions. *How do we reduce cost per site* invites optimisation and produces a cheaper installation. *What conditions let this still be running in year seven, after the person who understood it has left* invites redundancy, repairability and skills.

Carry both. State which is which.

One principle

Put every boundary where the record already has one, and give every boundary an owner.

In a human organisation, rules are guidance. A person reads them, judges, sometimes departs from them, and accountability is reconstructed afterwards by asking what happened and why. That works at a few dozen decisions a day.

An agentic organisation makes orders of magnitude more, and reconstruction stops being available. There is nobody to ask, and far too much to ask about.

Drawing the boundaries this way changes what a rule *is*. The organisation's values, and the rules derived from them, stop being guidance an agent is expected to follow and become the limits of what it is able to do. An agent acts inside a grant of authority or it does not act at all. **Accountability is settled before the action rather than established after it** — domain, owner and permission are fixed at the moment the work happens, and the record only has to state them.

That is the structural difference, and it is why an organisation built this way can be small, flat, and still checkable.

It only holds if the constraint lives in the substrate. A rule enforced by policy is guidance again, and an agent that can be argued past has no boundary. That is why there is no override in the design, including for the board.

How you would know this is wrong

This is a design claim and should be treated as one. Three things would show it false:

1. An agent acting outside a grant.
2. An action whose domain cannot be determined from the record alone.
3. A case where accountability still had to be reconstructed by asking someone.

Any of those means the constraint is living in policy rather than in the substrate, and the argument above does not hold. If you build this, they are the first three things to test — and the results are worth publishing whichever way they go.

Cite this version

```
Running an organisation where agents do the work, version 0.4 (August 2026). agenticgovernance.digital/downloads/built-to-be-checked-v0-4.pdf
```


PREVIOUS
D · What it does for you on a Tuesday

Series contents

NEXT
F · What it costs

Alongside: sources and provenance · slides