



BEYOND THE API STRATEGY

The Platform Nobody Builds On: What Enterprise Unreadiness Reveals About How Organisations Change

On the platform economy, the limits of the API strategy, and the organisational readiness no architecture diagram can supply

Giovanni Leonardi

September 2013

Contents

1. Executive Summary	3
2. The Platform, and What It Is Not	4
3. The Category Error	4
4. Integration by Control, Integration by Interface	5
5. What the Difficulty Reveals	6
6. The Honest Objection	7
7. Readiness as a Quiet Discipline	8
8. What Organisations Learn About Themselves	9

The Platform Nobody Builds On: What Enterprise Unreadiness Reveals About How Organisations Change

Transformation · No. 220

Originally written September 2013 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

*"An organisation that cannot be a platform to itself
will not become one for strangers."*

1. Executive Summary

Somewhere in most large organisations there is now a slide that says the company intends to become a platform. Behind the slide there is usually a programme: an API gateway has been procured, a developer portal stood up, a catalogue of interfaces published. And a year or two later there is a quieter fact that rarely reaches the same slide – almost nobody builds on the platform. A handful of interfaces are called; most integration still runs through the overnight batch and the shared database; the external ecosystem that was going to arrive has not arrived.

This essay is about that gap, and about what it reveals. The platform economy is commonly narrated as a technology story, and enterprise unreadiness for it is commonly diagnosed as a technology deficit – the wrong architecture, the legacy estate, the missing APIs. That diagnosis is not wrong so much as shallow. The reason the enterprise struggles to become a platform is not that it lacks interfaces. It is that a platform is an operating model wearing the costume of an architecture, and adopting the costume changes nothing about the body underneath.

The deeper argument is this: platform models expose, with unusual clarity, the mismatch between how a company's systems are wired and how its decisions are wired. To publish an interface is to make a promise that someone else can build on your capability without asking your permission. Enterprises are constitutionally organised to integrate through control rather than through interfaces – through projects, sign-offs, and ownership boundaries that map to the profit-and-loss statement. The platform asks them to integrate through contracts they do not control. That is not a technical request. It is a request to redistribute authority, and it is answered, predictably, by the organisation's antibodies.

If that is right, then enterprise unreadiness for the platform economy is worth studying for a reason larger than platforms. It is one of the clearest available windows onto how organisations actually change – and onto why the distance between transformation intent and transformation reality is so consistently the distance between changing the diagram and changing who decides.

2. The Platform, and What It Is Not

Begin with the thing itself, because the word has been stretched until it means almost nothing. In the popular telling, a platform is an app, or an ecosystem, or simply any business that is doing well online. That looseness is part of the problem: an organisation that cannot say precisely what a platform is will build precisely the wrong thing and call it done.

A platform, in the sense that matters here, is a business whose primary value comes not from producing a good but from orchestrating exchange between parties it does not own. The classical producer makes something and sells it; its advantage compounds through scale in production. The platform lets others produce, and its advantage compounds through scale in interaction — every additional participant makes the whole more valuable to every other. Economists have described the mechanics of this for a decade now, under labels like the multi-sided market: two or more distinct groups whose demand for one another is mediated by an intermediary that lowers the cost of finding, trusting, and transacting with a stranger.

The producer asks, how do I make more of what I sell? The platform asks, how do I make it cheaper for two strangers to do business through me than around me? These are not two versions of the same question. They imply different organisations.

The distinction matters because the enterprise, faced with the platform, almost always hears the first question when the second is being asked. It reads "become a platform" as "build a large piece of software that other things connect to," and sets about building it with the instincts of a producer: scope it, own it, control it, ship it. But orchestration is not a bigger act of production. It is a different act altogether, and the reflexes that make a firm good at the first are frequently the reflexes that make it bad at the second.

What the successful consumer platforms of the last few years have understood — the marketplaces, the app stores, the developer ecosystems — is that their most important design decision was not what to build but what to let others build. They drew a line around a small core they would own absolutely, and then made the space outside that line genuinely open: documented, stable, and safe to depend upon without a conversation. The value did not come from the core. It came from everything the core made possible for people the platform owner had never met.

3. The Category Error

Here is where enterprise unreadiness begins, and it begins as a category error rather than a capability gap. The enterprise treats "platform" as a question of architecture — a matter of interfaces, gateways, and integration patterns — when it is first a question of operating model. Architecture is downstream. The interfaces a company can sustain are determined by the way the company is organised to make decisions, and no amount of technical effort survives contact with an operating model that contradicts it.

The oldest observation in our field says as much. Nearly half a century ago it was noted that organisations which design systems are constrained to produce designs whose structure mirrors the communication structure of the organisation itself. We repeat this as a curiosity about software modules. Its real force is broader and more uncomfortable: the shape of what you can build is the shape of how you talk to yourself. An enterprise fragmented into functions that negotiate through committees and escalate through hierarchy will produce systems that integrate through committees and escalate through hierarchy — whatever the architecture diagram promises. You can draw clean interfaces between boxes on a slide. You cannot draw clean interfaces between departments that have never had to honour a contract with one another.

We should be honest that the enterprise has been here before, and recently. A decade ago the same promise arrived under a different name, when service-orientation was going to decompose the monolith into reusable services that any part of the business could compose at will. Much of what was built in that period is instructive precisely because it disappointed. The services arrived, but so did the enterprise service bus, the central integration team, the governance board that had to approve each new consumer, and the canonical data model that took eighteen months to agree and was obsolete on arrival. The technology was capable of openness. The operating model quietly re-imposed control at every seam, because control was the only way the organisation knew how to integrate.

*“We did not fail to build services. We built services
and then surrounded each one with the very
committee it was meant to abolish.”*

I raise the earlier episode not to score a point but because it carries the lesson the platform conversation is about to relearn. The failure mode was never technical. It was that a genuinely modular architecture was asked to run on top of an organisation that reserved every meaningful decision to a centre. The interfaces existed; the permission to use them without asking did not. And permission, not protocol, is the substance of a platform.

4. Integration by Control, Integration by Interface

To see why the enterprise finds this so hard, it helps to name the two ways an organisation can connect its parts, because they are rival philosophies and most large firms are built entirely on the first.

Integration by control is connection through a shared authority. Two systems, or two teams, are made to work together by a project that spans both, a plan that reconciles them, and a governing body that owns the seam between them. Every new connection is an event: it is scoped, funded, staffed, sequenced, and signed off. The strength of this model is that nothing happens without someone accountable having agreed to it. Its cost is that the number of connections a firm can sustain is bounded by the number of

connections its governance can personally supervise — which is to say, not many, and never quickly.

Integration by interface is connection through a published contract. One team exposes a capability behind a stable, documented interface and commits to keeping its promises; any other team may depend on that capability without negotiation, because the contract is the negotiation, settled in advance and once. The strength of this model is that connections no longer require supervision — they compose. Its cost, and this is the part the enterprise cannot easily pay, is that the exposing team must give up knowing, and controlling, who depends on them and why.

Dimension	Integration by Control	Integration by Interface
Unit of connection	The project	The contract
Cost of a new link	High, and paid every time	Paid once, at the interface
Who may connect	Those granted permission	Anyone who honours the contract
Bounded by	Governance capacity	The quality of the interface
Failure looks like	Backlog and bottleneck	Silent breakage of unknown consumers

The platform economy runs entirely on integration by interface, and it runs at a scale that integration by control cannot approach. An ecosystem of thousands of independent builders is possible only because none of them had to ask. The enterprise, meanwhile, is a cathedral of integration by control. Its budgeting is organised around projects, its accountability around ownership, its risk management around the principle that nothing connects to anything important without a named person having approved it. Ask such an organisation to publish an interface and mean it, and you are asking it to do the one thing its entire operating model exists to prevent: to let capability be consumed without a gate.

This is why the API programme so often produces the platform nobody builds on. Consider the composite that recurs. A large retailer or insurer decides to become a platform. Money is found, a gateway is bought, and within a year the catalogue lists some forty interfaces. Yet the numbers underneath tell a different story: three external consumers, most of them pilots; internal teams still moving the real data through the nightly batch and a shared database, because the batch does not require them to depend on another team's promises. Each of the forty interfaces was built as a project — scoped, delivered, and handed over. Not one of them changed the fact that to actually use a colleague's capability, you still open a ticket, wait for a governance slot, and engage a systems integrator for six weeks. The interfaces are real. The integration model never moved. The enterprise built the architecture of a platform on top of the operating model of a cathedral, and the operating model won, as it always does.

5. What the Difficulty Reveals

If enterprise unreadiness were merely a technology gap, it would be closing. The tools are cheap and abundant; interfaces are easy to build; the patterns are well understood

and widely published. The gap is not closing, and its persistence is the interesting fact. When a difficulty survives the removal of every technical obstacle, the difficulty was never technical. What the platform exposes, precisely because it is so easy to attempt and so hard to complete, is the true grain of how organisations change.

Three things become visible in the light the platform casts.

- Change is redistribution, not adoption. The rhetoric of transformation is the rhetoric of acquisition: we will adopt the cloud, embrace digital, become a platform, as though change were a thing to be brought inside and installed. But to become a platform is to move authority — from the centre that grants permission to the interface that renders permission unnecessary; from the owners of systems to the consumers of contracts. Adoption is additive and comfortable; you can adopt a technology and change nothing about who decides. Redistribution is subtractive for somebody, and it is resisted not out of ignorance but out of a correct perception of what is being asked. The organisation is not confused about the platform. It understands exactly what it would cost, and declines to pay in the coin of its own authority while paying freely in the coin of programme budget.
- Structure is the real interface. We are fond of saying that culture eats strategy, and it flatters us because culture sounds like a matter of values we could exhort our way out of. The harder truth the platform reveals is that structure eats architecture. The seams in your systems will follow the seams in your organisation, and if the organisation has no clean seam where the platform needs one — if the capability you want to expose is owned by three functions with three budgets and three roadmaps — then no interface you draw across that fault line will hold. The precondition for a good technical interface is a good organisational one: a capability with a single owner, a real internal customer, and a promise it is allowed to keep.
- The internal customer is the test. Every enterprise platform effort should be judged by a single unglamorous question long before any external ecosystem is contemplated: can one team inside this company depend on another team's interface, in production, without a meeting? Where the answer is no, the external ambition is a fantasy, because the external developer is only the internal developer with less patience and no political leverage. An organisation that cannot be a platform to itself will not become one for strangers. The internal customer, treated as a real customer, is where readiness is either present or absent — and it is almost always absent, because the internal customer has never been allowed to behave like one.

The distance between transformation intent and transformation reality is the distance between deciding to change the diagram and being willing to change who decides. Everything else is procurement.

6. The Honest Objection

There is a serious counter-argument, and a reflective treatment owes it more than a nod. It runs as follows: not every business should be a platform, and much of what is diagnosed as enterprise unreadiness is in fact enterprise prudence. Multi-sided markets are winner-takes-most, brutal, and unforgiving; for every ecosystem that flourished, a great many failed expensively. A firm with a defensible producer's business — real products, real margins, real customers — might reasonably decline to bet its operating model on becoming an intermediary in a game it is unlikely to win. On this view the enterprise's integration-by-control is not a pathology but a rational preference for accountability over openness, and the "unreadiness" is simply a firm knowing what it is.

This objection is strong, and it is right about more than it knows. Most enterprises should not try to become two-sided marketplaces, and the ones stampeding to do so because the strategy deck demanded it will mostly waste their money. If the argument of this essay were "every company must become a platform," the objection would defeat it.

But that is not the argument. The confusion in the boardroom — and it is worth naming plainly — is between the platform as a business model and platform-readiness as an operating discipline, and these come apart. Whether to enter a multi-sided market is a genuine strategic choice on which caution is often correct. But the underlying disciplines that platform-readiness demands — capabilities with single owners, stable interfaces honoured over time, the ability of one part of the firm to build on another without negotiation — are not a bet on any particular market. They are simply what it looks like for a large organisation to be able to change at the speed its environment now changes. A producer that will never run a marketplace still benefits enormously from being able to recombine its own capabilities without a project for every seam. The prudent firm is right to refuse the business model and wrong to use that refusal as an excuse to avoid the discipline. Declining to become a platform is a defensible strategy. Being unable to integrate without control is not a strategy at all; it is a limit, and the platform simply makes the limit impossible to keep ignoring.

7. Readiness as a Quiet Discipline

If the platform is a test of how an organisation changes, the encouraging part is that passing the test does not require the grand programme — and may be actively hindered by it. The grand programme is integration-by-control in its most ambitious form: a very large project to abolish the need for projects, which naturally fails, because it tries to supervise its way to a model whose entire point is the absence of supervision. Readiness is built the other way, in the small and the unglamorous, and it looks less like a transformation than like a change of habit.

1. Give every capability a single owner and a real internal customer. Before an interface can be stable, someone must be accountable for the capability behind it and answerable to someone who depends on it. Where ownership is split across functions, no contract will hold; the first work is organisational, not technical — resolving who owns the capability, so that a promise about it can be made by someone entitled to make it.

2. Make the interface a product, not a project. A project ends; a product is kept. The moment an interface is treated as a deliverable to be handed over, it begins to rot, and its consumers learn not to trust it. The disciplines that make interfaces dependable – versioning, backward compatibility, deprecation with notice, a commitment to not breaking the people who rely on you – are product disciplines, and they require a team that stays.
3. Let internal teams consume without asking, and count who does. The truest measure of readiness is not the size of the interface catalogue but the number of production dependencies that were established without a meeting. Remove the gate for internal consumption first; then watch what gets used. Consumption is the only honest signal, and it cannot be faked by a programme.
4. Resist the urge to govern the openness back out. The predictable enterprise response to interfaces that people actually depend on is alarm, followed by a governance board to bring the new connections under control. That instinct, indulged, rebuilds the cathedral one committee at a time. Governance of a platform is the discipline of the contract – clear, stable, enforced – not the supervision of each act of connection.

None of this is a technology programme, and that is precisely why it is hard. It asks the centre to hold less, the owners of systems to be answerable to their consumers, and the organisation to accept dependencies it did not individually approve. These are constitutional changes dressed as engineering practices, and they are undertaken, when they succeed, not by decree but by a slow accumulation of small refusals to integrate the old way.

8. What Organisations Learn About Themselves

I have come to believe that the platform economy will be remembered less for the businesses it created than for what it taught the enterprise about the enterprise. It functioned as a kind of assay – a reagent dropped into the organisation that turned a particular colour wherever authority was hoarded and integration was purchased one project at a time. Firms went looking for a technology and found a mirror.

The lesson generalises well beyond platforms, which is the reason to dwell on it. Every wave of change that the large organisation has met in recent years – the move to the cloud, the turn toward faster and more iterative delivery, the pressure to become in some fashion digital – has produced the same pattern: an intent sincerely adopted at the top, a programme dutifully funded, and a reality that stubbornly refuses to match the slide. In each case the explanation offered is a capability gap, and in each case the deeper truth is that the change required a redistribution of authority that the organisation was willing to describe but not to perform. The technology was only ever the occasion. The subject was always the operating model, and the operating model does not appear on the transformation roadmap because the people who write the roadmap are the operating model.

So the enterprise that wants to know whether it is ready for the platform economy – or for whatever the next reagent turns out to be – need not commission a study. It can ask itself the small question and watch its own face as it answers: can one of our teams build on another of our teams, in production, without a meeting? The honest answer to

that question is the whole of the matter. Everything the platform economy has to teach the enterprise is contained in the discomfort of giving it.

The gap between intent and reality, in the end, is not a gap in capability, budget, or technology. It is the gap between an organisation deciding to change and an organisation being willing to become the kind of thing that has changed. The platform did not create that gap. It only, and usefully, made it impossible to look away from.