



BEYOND CHANNEL THINKING

The Platform Promise Most Enterprises Cannot Yet Keep

Why API ambition outruns operating-model readiness in the emerging platform economy

Giovanni Leonardi

September 2013

The Platform Promise Most Enterprises Cannot Yet Keep

Transformation · No. 176

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

“A platform is not a channel with an API bolted onto it; it is an operating model that permits others to create value without waiting in the owner's queue.”

1. The Queue Behind the Screen

The demonstration lasts eleven minutes. A customer selects a service on a smartphone, changes an address, and asks for delivery on Saturday. The screen is clean, the response immediate, and the executive group sees the future: a new mobile channel, ready for launch before Christmas.

Then the questions begin. The address sits in three customer files. Saturday delivery requires an exception in the distribution system. Price eligibility is calculated overnight. The proposed application can display an order, but changing one calls for four internal teams, two outsourced suppliers and a release slot already committed for the quarter. The eleven-minute demonstration conceals a twelve-week queue.

This is the central enterprise problem of the platform economy as it appears in 2013. The visible contest is for better websites, mobile applications and digital partnerships. The deeper contest is for the ability to expose a dependable business capability, safely and repeatedly, without reconstructing the organisation around every new use.

Most established enterprises are preparing for the first contest and remain structurally unready for the second.

That unreadiness is easy to misdiagnose as old technology. Legacy systems certainly matter, but age is not the decisive variable. A twenty-year-old account system with clear ownership and a stable interface may be more useful than a new package surrounded by bespoke connections and ambiguous data. The greater obstacle is that the enterprise has been designed as a chain of controlled hand-offs, while a platform requires a set of capabilities that can be combined by people who do not own them.

The platform question is therefore not, primarily, whether the organisation has an application programming interface. It is whether the organisation can make a promise that survives reuse.

2. From Channels to Capabilities

For much of the past decade, digital investment has meant adding channels to an existing operating model. The website became another front door. Self-service shifted work from the contact centre. More recently, smartphones and tablets have created a demand for services that are immediate, contextual and continuously available. Yet beneath the new front door, the same sequence remains: request, validate, hand off, reconcile, report.

This model can produce attractive digital experiences, but it scales badly. Every channel builds its own interpretation of customer, product, price and entitlement. Every partner connection becomes a separate project. Each new proposition begins by discovering which system is authoritative and ends by creating another set of dependencies.

A platform model reverses the emphasis. It treats selected business capabilities as reusable services: identify a customer, obtain a price, reserve capacity, make a payment, confirm fulfilment. These services can support the website, a mobile application, a call-centre screen or a partner without each consumer re-implementing the underlying rules.

Model	Unit of change	Principal control	Typical failure
Channel	The customer interface	Release and brand approval	A polished front end tied to brittle back-end journeys
Pipeline	The end-to-end process	Functional ownership and hand-offs	Slow change at every boundary
Platform	A reusable business capability	Clear service contracts and accountable ownership	Reuse without reliability, or control without adoption

The difference is not semantic. In a channel model, the mobile team asks several functions for access. In a platform model, an accountable owner publishes a defined capability with conditions of use, service levels and change rules. The former coordinates work; the latter creates an asset.

A platform is not a channel with an API bolted onto it; it is an operating model that permits others to create value without waiting in the owner's queue.

This explains why apparently modern programmes disappoint. They purchase gateways, establish developer portals and wrap selected systems with web services, yet leave decision rights untouched. The interface is new; the queue survives behind it. The organisation has digitised the hand-off rather than removed it.

3. Why Readiness Lags Behind Intent

The unreadiness persists because several sensible management disciplines combine to produce the wrong result.

Capital is approved through projects, so teams are rewarded for delivering a specified output by a specified date. A reusable service asks for something more difficult: investment before every consumer is known, stewardship after the first project closes, and capacity reserved for changes requested by other parts of the enterprise. The project can declare victory when its application launches. The platform cannot, because its value depends upon continuing use.

Functional accountability creates a second barrier. Customer data belongs to one directorate, pricing logic to another, fulfilment to a third and infrastructure to technology. Each owner is rationally cautious about exposing a capability whose failure will appear on their ledger while much of the benefit appears elsewhere. Reuse creates shared value but concentrated operational risk.

Procurement reinforces the boundary. Outsourcing agreements and package contracts often price change by system or application. A new consumer of an existing service may look trivial on an architecture diagram while triggering a chain of commercial requests. The enterprise discovers that its interfaces are not merely technical connections; they are embedded in statements of work, service definitions and supplier incentives.

Data creates the most revealing difficulty. A platform makes ambiguity visible. A customer identifier that can be reconciled overnight for management reporting may fail when a mobile user expects an immediate answer. A product code that means one thing in sales and another in fulfilment can survive inside departmental reports, but not inside a service promised to several consumers. Platform work does not invent these inconsistencies. It removes the delays and human interpretation that previously concealed them.

- Project funding favours first delivery over continuing stewardship.
- Functional ownership separates the cost of reliability from the benefit of reuse.
- Supplier boundaries turn small interface changes into commercial events.
- Data ambiguity becomes operational at the moment of real-time use.
- Release governance protects individual systems while making cross-enterprise change slow.

These are not defects caused by inattentive managers. They are the predictable consequences of an enterprise optimised for stable processes and controlled change. That is why exhortation fails. Telling leaders to “be more digital” does not alter who may decide, who pays, who carries the risk or how a shared capability is maintained.

The test of platform readiness is not how quickly an enterprise can build its first interface. It is how safely a second team can reuse it without reopening every original decision.

4. The Serious Case for Caution

The strongest objection to platform enthusiasm deserves respect. Enterprises hold sensitive customer information, operate regulated processes and depend on systems that cannot be casually exposed. Opening capabilities to more consumers enlarges the failure surface. A poorly governed interface can spread an error faster than a manual process, allow a partner to exceed intended permissions or bind several channels to a service whose capacity was designed for one.

There is also a strategic concern. A business that makes its capabilities easy for intermediaries to use may weaken its direct customer relationship. It may turn a differentiated service into a component, surrender valuable insight to the party controlling the interface, or make price comparison easier. Not every capability should become a platform, and not every partner should receive the same access.

This caution is right about the hazards and wrong about the alternative. The choice is not between controlled internal systems and reckless openness. Connections are already proliferating through bespoke feeds, screen-scraping, file transfers and one-off web services. In many enterprises, refusing a platform approach does not prevent exposure; it makes exposure inconsistent and difficult to see.

The more disciplined answer is selective openness. Capabilities should be classified by business sensitivity and strategic intent. Some remain internal. Some serve approved channels. Some can be offered to selected partners under contract. A small number may justify broader access. Each class requires explicit rules for identity, volume, data, support and change.

This is slower at the outset than simply building the next connection. It is faster by the third or fourth use because decisions are made once and then governed, rather than repeatedly negotiated in project meetings.

5. What Reuse Reveals

Consider a composite service business preparing to let several distributors quote and book capacity directly. Its first plan treats each distributor as an integration project. One distributor needs a nightly availability file; another requests an online quotation; a third wants booking confirmation returned to its own system. The portfolio contains three projects, each with a sponsor, a budget and a separate supplier estimate.

The architecture review finds fourteen hand-offs between the initial request and confirmed booking. Capacity is recorded in two operational systems. Price exceptions are held in spreadsheets maintained by regional teams. Customer eligibility is checked manually for roughly one transaction in five. The quoted six-week connection can be delivered only by freezing the first distributor's requirements and postponing common data questions.

The enterprise changes the unit of work. Rather than fund three connections, it defines four capabilities: check eligibility, obtain a quote, reserve capacity and confirm a booking. A business service owner is appointed for each, supported by an integration architect, an information owner and an operations representative. The first distributor remains the proving consumer, but its needs no longer define the service alone.

The first connection now takes ten weeks, not six. That apparent delay is politically uncomfortable. Four weeks are spent agreeing what constitutes available capacity, removing two regional price workarounds, defining transaction limits and establishing a support route that does not depend on the project manager. The work seems slower because questions previously deferred until failure are being answered before reuse.

The second distributor connects in four weeks. The third takes three. More importantly, a change to the eligibility rule is made once, tested against all three consumers and introduced with a published notice period. Manual eligibility checks fall from one in five transactions to roughly one in twenty, limited to genuine exceptions. The turning point is not the installation of an interface. It is the decision that “eligibility” is an enterprise promise with an owner, not a fragment of logic inside each channel.

The numbers are modest enough to be credible and important enough to change the investment argument. The first consumer alone would not justify the additional work. Across three consumers, duplicate analysis, testing and supplier change are reduced. Across future consumers, the option value becomes larger still. Yet that value appears only if the services remain dependable. An abandoned interface is not an asset; it is another legacy.

“The first consumer proves that a service works. The second proves whether the organisation has changed.”

6. The Architecture Beneath the Architecture

Service-oriented architecture has already taught enterprises to separate functions and connect systems through defined services. Its technical lessons remain valuable. But the platform economy exposes the unfinished organisational work beneath that architecture.

A service contract cannot compensate for an absent business owner. A message standard cannot settle conflicting definitions of customer. A gateway cannot decide which capabilities should be offered to partners. Monitoring cannot resolve a funding model that pays for construction but not stewardship. These are management decisions expressed through technology.

The temptation is to create a central platform team to solve them all. A small central group is useful for common standards, identity, monitoring, developer support and design discipline. It becomes dangerous when it turns into another queue. If every consumer must petition the centre, the enterprise has concentrated integration without distributing capability ownership.

The better division is federated. The centre supplies the roads and traffic rules; business domains own the destinations. This arrangement is harder than either full centralisation or complete local freedom because it requires leaders to distinguish

what must be common from what may vary. But that distinction is the work.

Platform readiness can therefore be observed through a small set of behaviours:

1. The enterprise can name the business owner of a reusable capability, not merely the system owner.
2. The owner can state who may consume it, under what conditions and at whose cost.
3. Data definitions are operational enough to support immediate use, not merely retrospective reconciliation.
4. New consumers can be added through a known path with predictable commercial and release implications.
5. Reliability is measured from the consumer's journey across systems, not only within each component.
6. The capability has a life beyond the project that first paid for it.

These behaviours sound prosaic beside the excitement surrounding mobile services, cloud delivery and digital marketplaces. That is precisely why they matter. Markets often appear to be transformed by a striking interface, but the durable advantage lies in the machinery that makes repeated participation inexpensive.

7. The Enterprise Choice of 2013

The platform economy is sometimes described as a technology cycle: more connected devices, more software delivered as a service, more partners seeking direct access, more commercial activity organised through digital marketplaces. All of these forces are real. Yet treating them as a technology cycle encourages enterprises to respond with technology acquisition.

The more consequential shift is in the economics of coordination. When a capability can be discovered, accessed and combined at lower cost, smaller firms and external partners can assemble propositions that once required ownership of the whole chain. Established enterprises retain powerful assets: trusted relationships, operational capacity, information, specialist knowledge and distribution. Their disadvantage is not absence of assets but the cost and delay of making those assets usable in a new combination.

That is why enterprise unreadiness can persist even while digital spending rises. Money flows to visible channels because their benefits are legible and their sponsors identifiable. The less visible work of clarifying capabilities, ownership and data crosses budgets and challenges established control. One produces a launch; the other changes the enterprise's grammar.

There is no case for turning every function into an open service or for dismantling controls built through hard experience. The serious requirement is narrower and more demanding: decide which capabilities matter strategically, make their promises explicit, and organise for their repeated use.

The enterprises most likely to prosper will not necessarily possess the newest systems or the largest digital budgets. They will be those that shorten the distance between an

idea at the edge and a dependable capability at the core. In 2013, that distance is still mostly measured in meetings, reconciliations, supplier requests and release calendars.

The platform economy will first be noticed on the screen. Enterprise readiness will be decided in the queue behind it.