



BEYOND THE API FACADE

The Platform Nobody Joined: Why Enterprises Mistook API Plumbing for a Platform Strategy

On the difference between exposing your services and surrendering your control — and why only one of them is a platform.

The Platform Nobody Joined: Why Enterprises Mistook API Plumbing for a Platform Strategy

Transformation · No. 218

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

“The platform, examined honestly, was a beautifully engineered reading room attached to a building whose doors were still locked from the inside.”

1. The morning the ecosystem did not arrive

There is a particular quiet that settles over a programme in the weeks after a platform goes live. The developer portal is up. The API catalogue is published, versioned, documented, with a getting-started guide and a working code sample someone stayed late to test. The gateway meters every call and the dashboards glow through the night. And then — very little. A trickle of registrations, most of them the team's own engineers checking their handiwork. One or two partners who were going to integrate anyway, contracts already signed. The ecosystem that was supposed to gather around the platform does not gather. Six months on, the register shows a few dozen developer accounts and a handful of live integrations, nearly all of which could have been built as ordinary point-to-point interfaces the old way, for less money.

I have watched this scene play out more than once, and the reaction inside the organisation is always the same: first puzzlement, then a hunt for the missing feature. Perhaps the documentation was too thin. Perhaps we need an evangelist, a hackathon, a slicker portal, a louder launch. None of these instincts is wrong, exactly, but they all answer the wrong question. The platform did not stall for want of a feature. It stalled because the organisation had built the plumbing of a platform without ever making the decision a platform requires.

That gap — between the plumbing and the decision — is, I think, the honest story of the platform economy and the established enterprise. We are living through a period in which software is visibly eating industry after industry, in which a handful of consumer businesses have grown at frightening speed by letting other people build, sell, and transact on top of them. The instruction to the incumbent has been clear and insistent: become a platform, or be intermediated by one. What has been much less clear is what that instruction actually asks of a company that was never built to be a platform in the first place.

2. Two things we learned to call by one name

The confusion of this era, to put it bluntly, is that we gave a single word — platform — to two completely different undertakings, and then measured one by the promise of the other.

The first undertaking is API-enablement. You take the capabilities buried inside your systems — a quote, a policy lookup, an account balance, a shipment status — and you expose them as clean, documented, secured web services behind a gateway. This is real engineering and it delivers real value. It is also, crucially, something an enterprise can do entirely on its own terms. It lives in the technology function, it has a budget line, it can be planned, staffed, and governed like any other programme. Nobody outside the building has to change their behaviour for it to succeed.

The second undertaking is a platform strategy: a deliberate bet that you will create more value by opening a side of your business to outside participants than you can create by keeping that side closed. It rests on network effects — the mechanism by which each new participant makes the whole more valuable to every other — and network effects are not something you build. They are something you attract, by giving other parties a reason to invest their own time and money on top of you. That reason almost always involves ceding something: a margin, a customer relationship, a decision you used to make yourself.

API-enablement is a technology programme you can complete in a room. A platform strategy is a change in who gets to make money from your business — and that is never a decision a gateway can make for you.

These are not two points on a maturity curve, where you do the first and gracefully arrive at the second. They are different in kind. The tragedy of so many "platform strategies" of the last few years is that they delivered the first undertaking flawlessly and were judged against the second, and so a genuine engineering success was recorded, internally, as a disappointment.

3. What the enterprise was built to defend

To see why the second undertaking is so much harder, it helps to remember what a large established enterprise actually is. It is, overwhelmingly, a machine for owning a value chain end to end. It sources, it manufactures or underwrites or originates, it prices, it distributes, it services — and its competitive advantage, its margin, and its sense of itself all sit in the seams between those steps, in the parts of the chain it controls and others cannot easily replicate.

A platform asks you to open precisely those seams. It asks you to let a third party originate the customer you thought was yours, or price a risk you thought only you could price, or assemble your components into a product you did not design and might not approve of. The API is trivial by comparison. The API is just the doorway. The hard part is deciding to leave the door unlocked at the one point in the building you have spent your entire corporate life keeping shut.

This is why the same organisations that ship an API programme on time freeze when the platform question becomes real. The instinct that made them successful — protect the margin, own the relationship, keep the decision inside the committee — is exactly the instinct a platform punishes. And no amount of engineering resolves an instinct.

“We exposed our services with great discipline. What we could not bring ourselves to expose was a decision.”

4. The honest ledger

I want to be concrete, because the abstraction flatters both sides. Consider a composite that is true to life: an established, multi-line organisation that ran a two-year “platform” programme. On the engineering ledger, it worked. Partner onboarding, which had reliably taken around ninety days of bespoke integration, dropped to roughly nine once partners could self-serve against documented APIs. Internal reuse rose sharply; three separate channels that had each maintained their own copy of the same account logic collapsed onto one service. The cost of the next integration fell every time, because the marginal integration was now configuration rather than construction. These are not vanity numbers. They are the genuine, bankable return of doing API-enablement well.

And on the platform ledger — the ecosystem, the outside innovation, the network effect that was going to bend the growth curve — the result was close to nothing. The external developers did not come, because there was nothing on offer they could turn into a business of their own. Every API returned data; none of them conferred the ability to do anything the organisation had not already decided to allow. You could read a price but not set one. You could see a product but not compose a new one. The platform, examined honestly, was a beautifully engineered reading room attached to a building whose doors were still locked from the inside.

The lesson is not that the programme failed. It is that it succeeded at the thing it was actually doing and failed at the thing it was named after — and that the naming did real harm, because it set an expectation no gateway could ever meet and quietly discredited a piece of work that deserved to be called a win.

5. The objection worth taking seriously

There is a strong version of the opposing view, and it deserves a real answer rather than a straw one. It runs like this: most established enterprises have no business trying to become platforms at all. Two-sided markets are for the digital natives who were born with no chain to protect. For everyone else, the sensible move is simply to modernise integration — expose the services, onboard partners faster, cut the cost of

every future connection — and get on with the actual business. On this view, API-thinking is not the easy half of the job; it is the job, and the talk of surrendered control and inverted value chains is consultant theatre.

I have a good deal of sympathy for this, and where it is right it is very right. Many organisations genuinely do not need to be platforms, and would waste years and reputations chasing network effects they will never earn. For them, disciplined API-enablement is not a consolation prize; it is the correct strategy, honestly named.

But that is the whole point. The failure of this era was not that enterprises attempted platforms and fell short. It was that they attempted API-enablement — the sensible, achievable thing — and called it a platform, importing an expectation of ecosystems and network effects that the work was never designed to produce. The remedy the objection proposes is exactly right; it simply has to be stated plainly. Do the integration work, bank the ninety-days-to-nine, and stop dressing a competent modernisation programme in the borrowed language of disruption. Honesty about which of the two undertakings you are actually doing is worth more than any amount of platform vocabulary.

6. What readiness would have meant

If "enterprise unreadiness" means anything useful, then, it was never a readiness of technology. The gateways worked. The services were clean. The engineering, by and large, was the strongest part of the whole endeavour. What was missing sat one floor up, in the questions the technology allowed everyone to avoid.

- Are we willing to let someone outside this company originate a customer, and share the value when they do?
- Are we willing to let a decision we have always made — a price, a term, an eligibility — be made by a rule that a third party can invoke without asking us first?
- And if opening that seam lets a partner capture part of the chain we used to own, do we believe the larger, faster market is worth the margin we give up?

An organisation ready to answer those questions honestly, in either direction, was ready. An organisation that treated them as engineering questions and pushed them down into the API backlog was not — and no portal, however polished, was ever going to make it so. The platform economy did not catch the enterprise short of technology. It caught it short of the willingness to decide who, in the end, gets to build on top of what it owns. That is a boardroom question wearing an engineer's overalls, and until we take the overalls off and see it for what it is, the next portal will launch to the same particular quiet as the last.