Avoid these 5 anti-patterns

when going platform-first

Manish Kumar
Head of India Domestic Business

Mujiruddin Shaikh
Market Technology Principal

Smitha Hemmigae
Head of India Marketing

ThoughtWorks®

Content

Platform-first has become such a common business-tech phrase that HBR has coined a term for this phenomena; "'platformania'...a land grab, where companies feel they have to be the first mover to secure a new territory, exploit network effects, and raise barriers to entry."

But, the excitement is not without reason. 7 of the world's 10 largest global companies are underpinned by some of the most evolved technology platforms we've seen. In this paper, we discuss what extremely successful businesses have in common i.e. platform technology.

In a reality where every business wants to either build or become one, it's a good idea to look at what platform thinking is.

The nonlinear growth curve achieved by technology platforms is centered on:

- **Removing friction within delivery teams.** This is achieved through high quality, self-service access to foundational technology.
- **Creating an ecosystem of technology and business capabilities.** This is achieved by leveraging domain-driven principles and carefully-factored business capability interfaces (APIs) that are critical to an effective and modern Service Oriented Architecture (SOA) strategy.
- **Explicitly investing in a foundation for experimentation.** This is achieved by empowering delivery teams to easily test new ideas and validate learnings.

IDC predicts, " By 2020, 60% of all enterprises will be in the process of implementing an organization-wide digital platform strategy." This is indicative of India Inc. not being too far behind when it comes to 'platformania.'

Infact, the Indian government has been treading this path with the trinity of JAM (Jan Dhan accounts), the Aadhaar ID system and mobile technology. India's digital backbone, another moniker for this trinity, is fueling rapid digital transformation across the country and fostering a competitive innovation agenda that both the public and private sector can partake in.

For instance, fintech is one of the biggest beneficiaries of the Aadhaar platform with, now, countrywide access to banking and finance services through banks, loans, microinsurance, lending and more.

Another example is Jio Digital Services, a new digital services company that is set to act as an umbrella platform for all the Reliance-owned digital businesses including Reliance Jio, MyJio, JioTV, JioCinema, JioNews and JioSaavn.

The platform is investing in emerging technologies such as blockchain, artificial intelligence and machine learning, and is also embracing virtual, augmented and mixed reality, natural language processing alongside voice-enabled services.

More recently, Facebook's purchase of a nearly 10% stake in Reliance Industries' digital business unit; Jio Platforms is valued at $5.7 billion. This deal places Reliance's digital operations at around $66 billion and pushes the Indian conglomerate ahead in its plans to de-leveraging its balance sheet while accelerating the launch of its new commerce business.

Experts say the arrangement between Reliance Retail, Jio Platforms and Facebook-owned WhatsApp can offer consumers access to the nearest kiranas, or grocery stores.

Are all companies experiencing such success, alike?

At ThoughtWorks, the digital native company that has pioneered tech expertise for more than 25 years, we have seen a fair share of large enterprise platform initiatives falter. Incidentally, our services are often sought to rescue such failed initiatives. And, it's based on this experience that we have put together the 5 anti-patterns that organizations should avoid when they go down the inevitable platform journey.

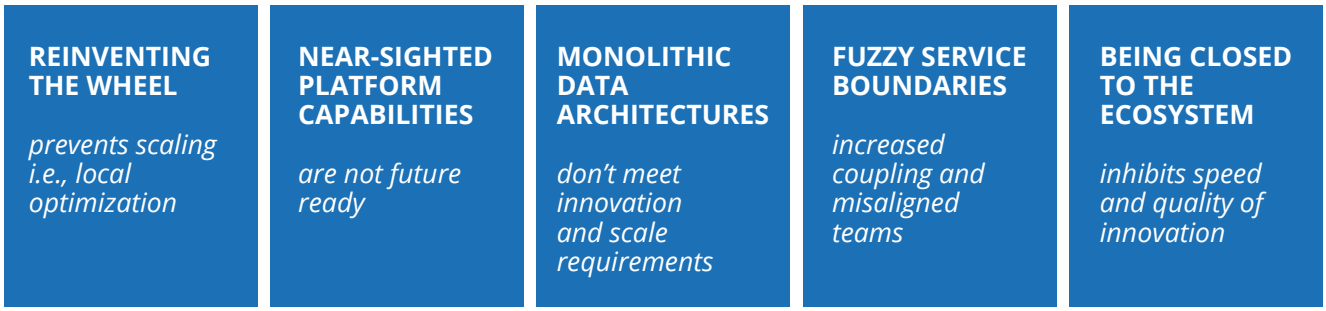

Fig 1: 5 Anti-patterns to avoid when going platform-first

Anti-pattern 1

Reinventing the wheel

prevents scaling i.e.local optimization

An enterprises' tech initiatives should instigate a network effect that lets individual teams focus on feature development. Leadership needs to prioritize the adoption of such initiatives alongside clear messaging on how the approach makes life easier for all the teams involved. Let's discuss this with a few examples. We're firm advocates of autonomous teams, where each cross functional team has the freedom and flexibility to build software suited to their specific needs. This encourages both a deep understanding of business problems and a highlevel of ownership to see them through. However, one side effect is each team tends to also create their own build and deploy pipelines along with monitoring infrastructure.

Also, multiple teams end up solving similar problems in multiple different ways using several tools. This results in an obvious duplication of effort that's narrow minded and focused on the immediate needs of that particular team rather than the comprehensive need of the enterprise. A lack of infrastructure standardization causes delays in new projects. Add to this, the constantly shifting regulatory landscape, the constant crop of new platforms and tools to explore and support - and we have ourselves an operations team that's pretty overwhelmed.

A recent AWS survey found that it takes more than a month to deliver new infrastructure for 33 percent of companies, and more than half had no access to self-service infrastructure. This is indicative of companies still struggling with their IT operations. Their systems engineers spend far too much time putting out fires and manually building, configuring and maintaining infrastructure. The result - early burnouts and frustration.

Our recommendation is to develop a technology platform that provides build, deploy and monitoring capabilities for the feature development teams. This requires standardization of the technology stack but relieves multiple teams from building the same deployment pipelines and monitoring stack.

Ownership is maintained with the feature teams as they have access to environments and the monitoring stack. Building a self service platform also reduces friction between the feature development team and the platform development team. This approach requires moving beyond the project-as-the-primary-mechanism for funding, staffing and delivery of technology.

Self-service infrastructure is a recommended approach because it lowers costs, removes silos, and increases productivity. It enables the effective use of technology investments, shortens release cycles and hikes employee satisfaction. Self-service infrastructure enables app teams to focus on business features while, build, deploy and observability is provided out-of-the-box to them. The app teams will continue to possess choice/control over what they need from the self-service infrastructure.

Here's another example. Often, multiple divisions within the enterprise end up building similar functionalities. A conscious effort to identify the functionality and standardize API contracts will save time and effort from being wasted in building similar APIs. Our suggestion is to encourage an enterprise-wide API repository that will greatly enhance discoverability and the reuse of existing APIs. This thinking should not be restricted to API platforms alone. One can build mobile platforms equipped with a framework to build Super Apps.

The framework will allow embedding independently developed mini apps while providing cross-functional features like context awareness to the apps. This makes it easier to develop and evolve functionalities much faster.

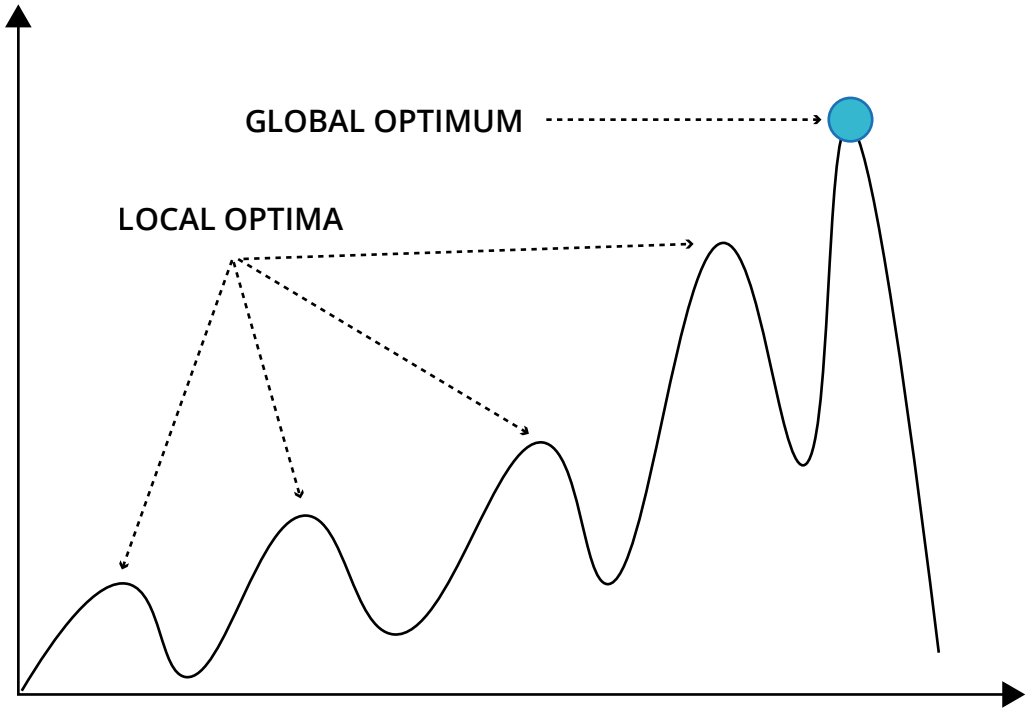

Fig 2: Global optimum and local optima

Anti-pattern 2

Near-sighted platform capabilities

are not future ready

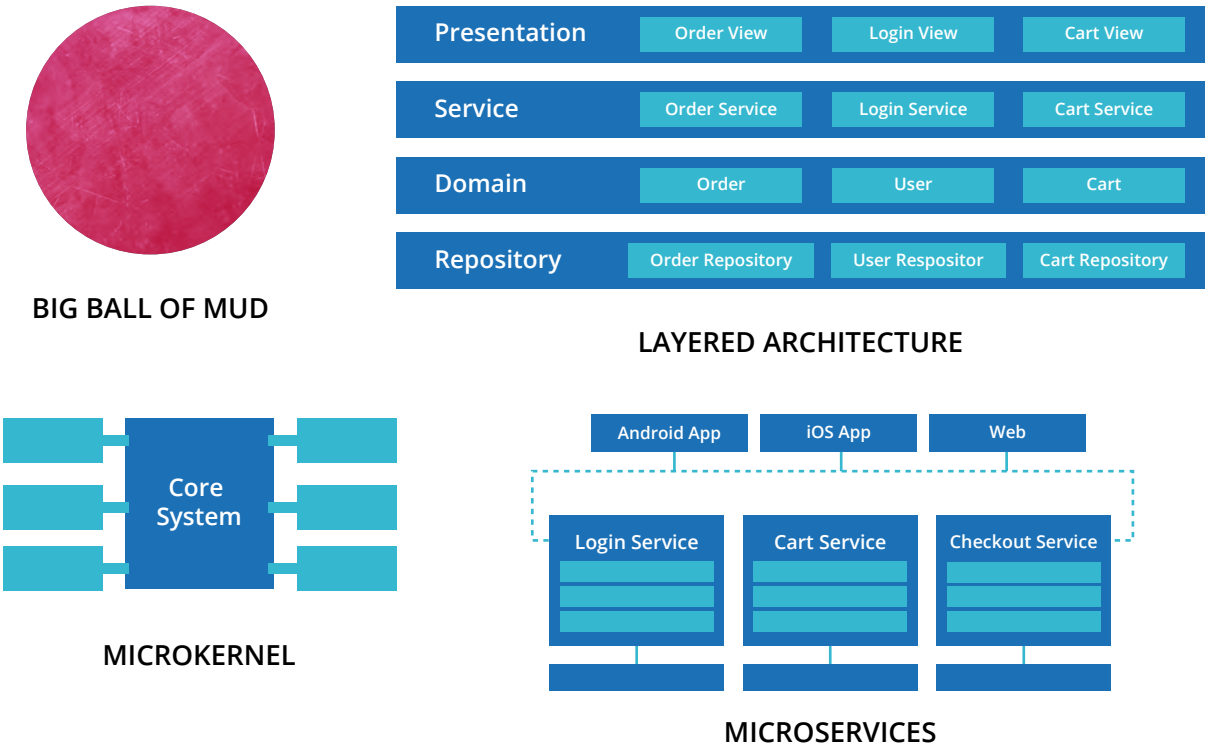

Fig 3: Architectural patterns

A significant hurdle when building platforms is the traditional enterprise's mindset to evaluate a platform on it's current requirements, and not on how it will evolve to meet the business' needs in the near and far future. Given that a business has to perpetually accelerate towards market relevance and customer acquisition, a single big bang creation followed by low to nil platform evolution will not work.

Our counsel is to design and manage enterprise platforms by architecting them to be modular and evolutionary, at the same time. This allows continual changes to both business capabilities and the technologies that power them.

Also utility components, that are not strategic to the business, need not be custom built but can be bought. For such components, 'buy and integrate' is a credible strategy to retain control over future evolution. It helps aligning with the long term tech roadmap and evolving future-perfect capabilities when needed.

One note of caution is that building for the future does not mean everything needs to be thought through now, but rather crafting an architecture equipped to adapt to perpetual evolution. This approach allows changing out elements of architecture, adding new parts in less time and incorporating the latest functionalities into the platform.

Anti-pattern 3

Monolithic data architectures don't meet innovation and scale requirements

An article by Zhamak Dhegani, Principal Technology Consultant at ThoughtWorks says it best with these words, “Many enterprises are investing in their next generation data lake, with the hope of democratizing data at scale to provide business insights and ultimately make automated intelligent decisions. Data platforms based on the data lake architecture have common failure modes that lead to unfulfilled promises at scale. To address these failure modes we need to shift from the paradigm of a centralized data lake, or its predecessor data warehouse. We need to shift to a paradigm that draws from modern distributed architecture: considering domains as the first class concern, applying platform thinking to create self-serve data infrastructure, and treating data as a product.”

Our guidance is to, “be open to the possibility of moving beyond the monolithic and centralized data lakes to an intentionally distributed data mesh architecture; Embrace the reality of ever present, ubiquitous and distributed nature of data.” The shift, “requires a new set of governing principles accompanied with a new language:

- Serving over ingesting
- Discovering and using over extracting and loading
- Publishing events as streams over flowing data around via centralized pipelines
- Ecosystem of data products over centralized data platform

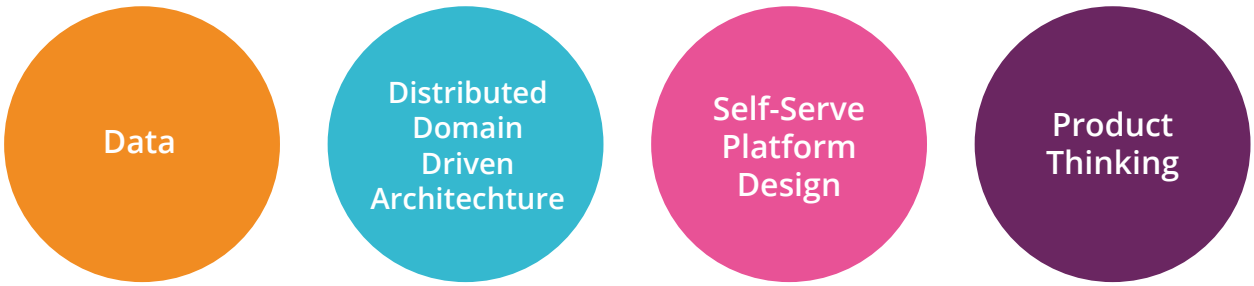

Fig 4: Convergence: the paradigm shift for building the next data platforms

Understandably, ensuring good data quality is difficult because a centralized team will not understand different types of data as well as the teams producing said data. Appropriate would be, to have the customer data product team manage customer data and ensure its quality. New data products being built on top of existing customer data products are assured of good data that's been provided by the customer data product team. This also guarantees a robust governance model of how to pool, use and distribute data that powers the platform andaugments its potential impact.

Anti-pattern 4

Fuzzy service boundaries

increased coupling and misaligned teams

An organization cannot build a distributed platform armed with the mindset of building monoliths. When designed well, a distributed system ushers in the benefits to incrementally release and scale independent parts.

We've witnessed enterprises not applying/using Domain-Driven Design (DDD) principles leading to the lack of well-defined bounded contexts and service boundaries. This causes any change to spread across multiple services, increasing coordination and alignment efforts amongst multiple teams. Performance also takes a hit because of the increase in service calls.

Identifying bounded contexts leads to cohesive services, designed around business boundaries. Teams aligned to bounded contexts speak a ubiquitous language, ensuring complete alignment between business and tech teams. Understanding business context helps effectively drive the long term roadmap for the domain.

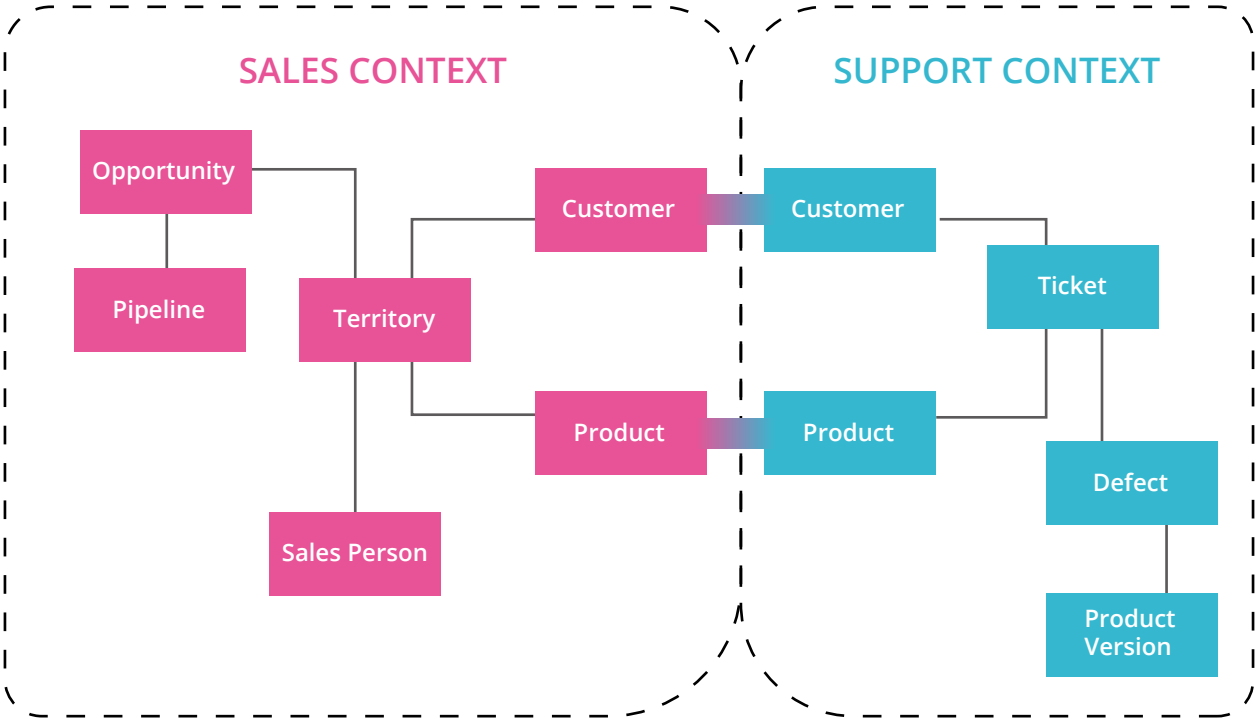

Fig 5: Bounded context

Anti-pattern 5

Being closed to the ecosystem inhibits speed and quality of innovation

A traditional approach followed by enterprises is to own (build or buy) any capability offered to their clients. It's important to nurture an ecosystem view of business, where we leverage others' capabilities as well as expose our own capability to others. This helps the enterprise sustain innovation.

Core capabilities that differentiate the enterprise should be built into the platform. For non-core or very specialized capabilities, the organization should leverage innovations that are current to the market. The platform should orchestrate ecosystem-wide capability integrations that will result in accelerated time to market, with market leading solutions.

For instance, India's Aadhaar platform allows 3 forms of biometric authentication; thumbs, fingers and iris scan. The biometric device's captures come from third parties who follow the Unique Identification Authority of India's (UIDAI) prescribed method for device integration. Here, UIDAI's core team is not building the biometric system themselves, as it's not their core capability. It's being built by partners within the ecosystem. When the platform is modular, like in this instance, such integration and collaboration is uninterrupted.

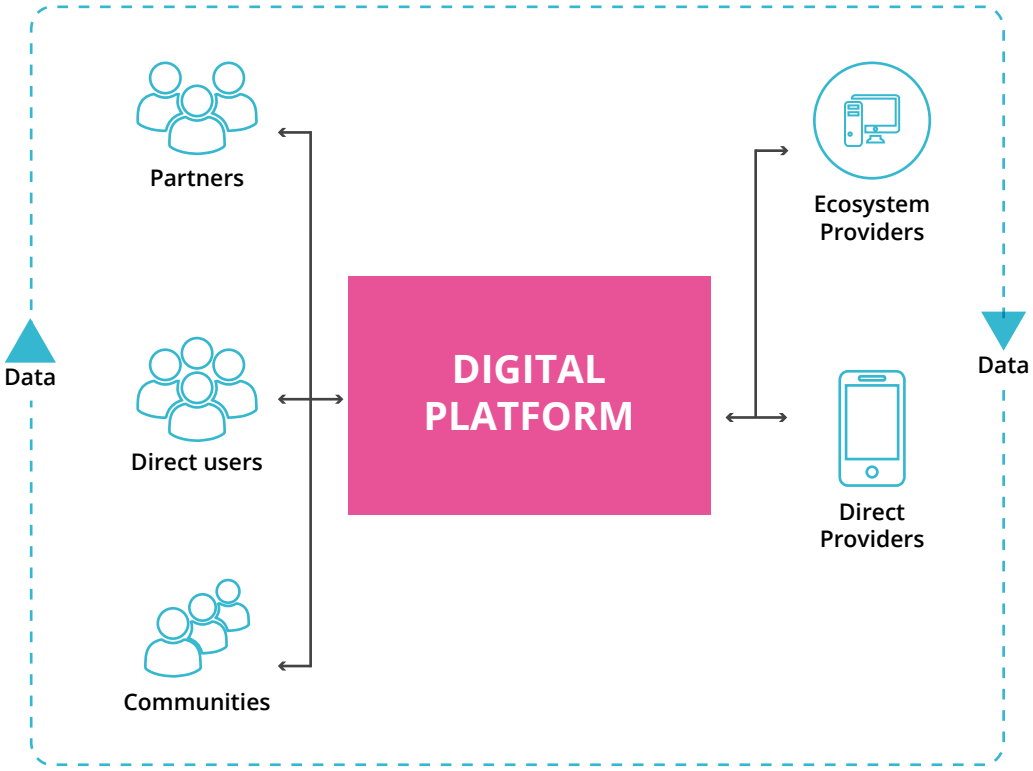

Fig 6: Digital platform ecosystem

Banking as a Service is another great example of how the headless platform exposes its capabilities. A bank is initiated into the customer's context of shopping where the latter can avail a service through a digital company's engagement layer. To illustrate this further, let's look at the act of purchasing a car through a dealer. All activities r aight from the Know Your Customer (KYC) process, to establishing credit worthiness, to selecting a loan product, to loan approval, to opening a loan account and more can be carried out within the context of that car purchase.

These applications can extend to other sectors like retail where the shopping experience is augmented via text, voice or video based channels like WhatsApp, Twitter, SMS etc. in addition to conventional web based channels.

While staying competitive in a world that prioritizes superior customer experience and market relevance, businesses are seeing the clear link between continuous innovation of products or services, and platform thinking.

We, at ThoughtWorks are seeing a record number of businesses embrace the proven platform-first approach to unlock transformational results.

About ThoughtWorks

We are a software consultancy and community of passionate purpose-led individuals, 7000+ people strong across 43 offices in 14 countries. Over our 25+ year history, we have helped our clients solve complex business problems where technology is the differentiator. When the only constant is change, we prepare you for the unpredictable.

More here

ThoughtWorks®

thoughtworks.com