Luxoft
A DXC Technology Company

Trends in technology modernization

The implications and advantages of moving to the cloud, and using open source in banking

by Mark Maynard
Technology Enablement Practice Lead
Luxoft

Today, all new development is cloud-native, cloud-first. It may be bold, but I suspect that by the end of 2022 the only things we see left on-premise are run-to-kill applications. With the move to cloud, we'll see a whole new ecosystem of aaS offerings. Those who don't make the shift, risk getting left behind and becoming uncompetitive and irrelevant.

Financial industry in the cloud

The existing mix of vendor and bespoke systems in the trading space is set to continue; the primary agenda is cloud option. Our industry has finally got the elasticity, the options and the advantages of the cloud, so the majority of banks are now rushing to make the move (to cloud). Consequently, organizations are looking at porting their existing stacks to a cloud provider — this might necessitate some platform changes, but the move to the cloud is the focus. This shift will precipitate more modular, dynamic and layered platforms and really start to facilitate aaS models. A fine level of granularity will emerge supporting more Agile and dynamic architectures that really focus on integration and fast change.

What's the focus for the different layers?

Let's simplify to platform services, internal (non-differentiating) business services and client facing services.

At the platform layer, the question nowadays is buy versus adopt open source (as opposed to build). For the business services layer the question is probably, or will be, build vs. buy vs. rent (aaS model). For business services where there is differentiation, bespoke build will prevail. Now there is also a subtext in IT versus user build specifically for internal systems; where enterprise IT will focus on building core trading frameworks and services that allow users to build their own low-code models and workflow. While externally facing systems (for the moment at least) will be bespoke built by enterprise IT.

For IT in most banks and trading houses at the moment, the ideal path is:

Rent for cloud elasticity and cost

Adopt for platform architecture (probably wrapped in a support model)

Buy for non-differentiating business services, perhaps moving to rent

Build for differentiation at the client touchpoints

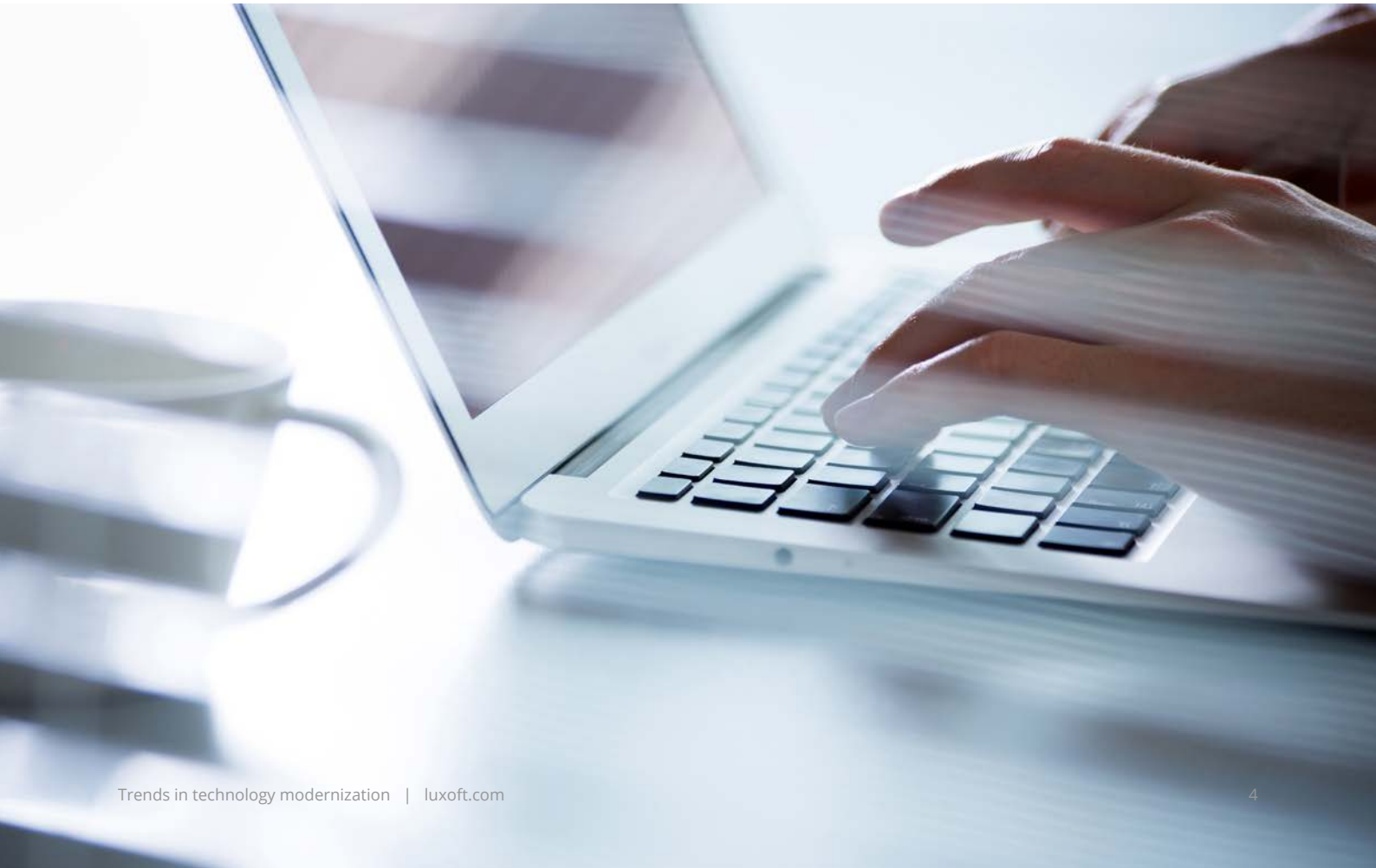
The key IT theme is to enable a dynamic plug-in environment, and a move toward becoming data-driven.

What is driving the modernization of trading systems, apps and infrastructure in the banking industry?

Three themes. The first is democratization and the cloud. The pace of IT change is outstanding and the key driver here is democratization. Internet technology opened new markets and ways of engaging (first on personal desktops and then mobile devices); cloud technology is the next shift — it will open up a whole new ecosystem of platform services and allow banks to operate on a leaner estate. It will also lead to further disruption. With the cloud, elastic computing has unleashed even more possibilities in data processing and integrated management reporting, scenario modeling, and custom insights, to name a few. Organizations need to take advantage of these.

Second, for the last 20 years, it's been far too easy for organizations to be application-centric, rather than focus on business and user requirements. This has led to gaps and poor user experiences that need to be addressed. The issue has been exacerbated recently with budgets focused on mandatory requirements meaning pent-up demand.

Finally cultural change; the tech-savvy generations are now moving into leadership positions — bringing business users with the ideas and ability to innovate outside of IT. This is shifting what it means to be enterprise IT and how it operates.



What are the benefits of adopting open source technologies?

Code may have originated from an individual, academia or a corporation — what's key to making an open source project viable is that is has been actively adopted by the community and an effective community governance model is in force; perhaps we should call it 'community software'. For such software, we'd expect to see worldwide commits on GitHub every day and a massive community behind the code, providing unmatched scale — the key benefit.

There were security and longevity concerns regarding the use of open source, but as long as the system has scale, these concerns are nullified. In fact, because of its scale, community-developed software is the safest software available.

Historically, banks resisted using open source, but around the late 2000s, attitudes changed. This was largely thanks to corporations like Google publishing their systems as open source and seeing widespread adoption. Because the uptake was so prolific and the community surrounding open source became so enormous, it cannot be ignored.

There's a number of reasons why open source became so popular and why it's become so important:

- It's independent of any country, organization or author (so there's no company, country or key-man risk)
- Open source is generally easier to install (there's no negotiation or lead time), often being the de facto standard in cloud services
- Open source, by definition, is usually also an open standard. Software should be insulated throughinterfaces to limit lock-in. Open source will have open standards in place (or evolving), which will support portability
- With the community, the talent pool on these platforms is simply bigger
- The platform itself will evolve, both functionally and in terms of the number of options, and get bug fixes at rates which can't be matched privately

Let's categorize the open source world into three parts:

- Tools for bespoke development — that's design time activity: Code, libraries, and design time tools
- Platform services: Data stores, event systems and messaging etc.
- Business services (e.g., a risk framework)

For bespoke development, the question is a moot point because every developer in every organization will be doing this through Maven or similar; using UI frameworks; using tools like Jenkins. This has all become mainstream.

For platform services, because the communities are cross-industry and so large, and as above they are often standard services across cloud hyperscalers, there is little alternative but to adopt.

Finally for Business Platforms, the issue here may be scale. Without a large community backing an open source platform, it isn't viable. There are some solutions of course, but is the scale really there in financial services?

Is there a future for open source in the banking industry?

Open source is typically successful because it was developed by a player. Banks can be considered as software houses. We do see banks publishing PoCs and ancillary parts of their system, but nothing meaningful so far. Will a bank that has invested in a bespoke platform publish core source code? This has to happen eventually — as the first bank to do so will gain the advantage. This will be the game changer, so that's a space to watch.

About **the author**

Marc Maynard
Technology Enablement Practice Lead,
Luxoft

Marc Maynard has been delivering technology, mainly in capital markets and banking, for 30 years. He's performed bespoke builds from front office to back; predominately trading, P&L and risk platforms. Now he's responsible for technology enablement capability at Luxoft — helping tech departments within financial services work better: Helping clients go faster, better and cheaper.

About Luxoft

Luxoft is the design, data and development arm of DXC Technology, providing bespoke, end-to-end technology solutions for mission-critical systems, products and services. We help create data-fueled organizations, solving complex operational, technological and strategic challenges. Our passion is building resilient businesses, while generating new business channels and revenue streams, exceptional user experiences and modernized operations at scale.

luxoft.com