

# Evolution of cloud

by Marc Maynard, Senior Director

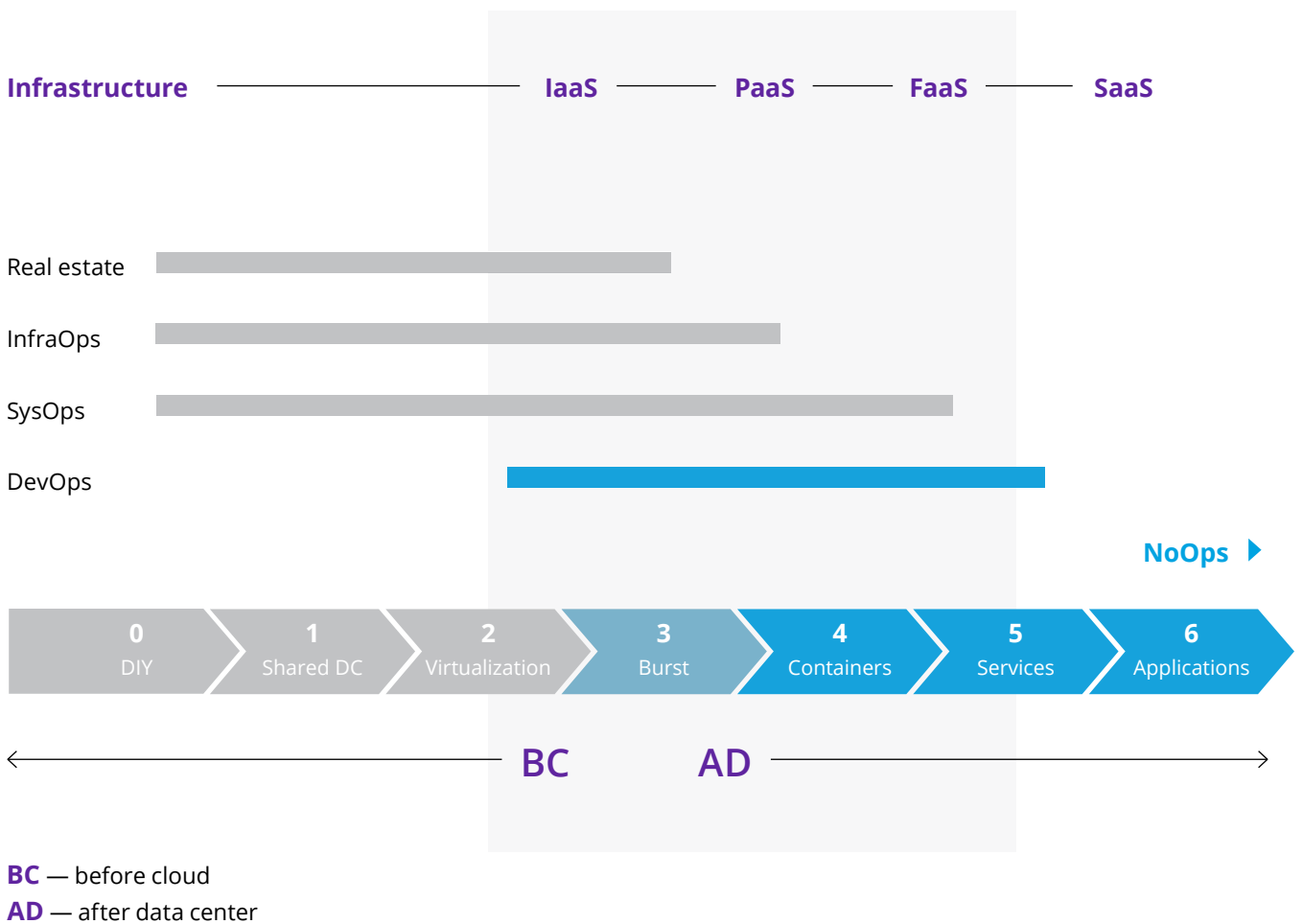
# Table of **contents**

<b><u>Introduction and executive summary</u></b>	<b>3</b>
<b><u>Infrastructure BC</u></b>	<b>4</b>
0. DIY data center and in-house platform services	4
1. Shared data center	4
2. Virtualization	4
<b><u>IaaS (Infrastructure as-a-Service)</u></b>	<b>5</b>
3. Dev, test and burst in the cloud	5
<b><u>PaaS (Platform as-a-Service)</u></b>	<b>6</b>
4. Containers and cloud deployment	6
5. Cloud platform services	7
Open-source software (OSS)	7
Serverless and FaaS	7
Platform services	8
Business services	9
<b><u>SaaS (Software as-a-Service)</u></b>	<b>9</b>
6. Applications	9

# Introduction and executive summary

This article explores the evolution of enterprise technology through seven stages. It highlights the impact on system development in capital markets and why organizations must adopt cloud-, modern- and service-orientated architecture.

**Diagram 1. Evolution of enterprise technology**



# Infrastructure BC

Before cloud (BC), there were three phases of infrastructure: DIY, shared data center (DC) and virtualization. Understanding these phases and how they were progressed by institutions' technology infrastructure departments helps us appreciate the origins, and some of the benefits, of cloud technology.

## 0. DIY data center and in-house platform services

The data center was situated on business premises with an array of staff to look after it, ranging from building services (security, building maintenance, utilities, racks), through server administrators and network engineers, to database administrators.

Rolling out a new system meant commissioning new, dedicated hardware. Ordering and installing could easily have a lead time of 3 months and, if there were any constraints such as data center space (during a DC migration perhaps) or a skill shortage, the lag time could hit 12 months.

The response of the bank's technology division to new market opportunities and regulatory requirements was very slow (typically planned in quarters and years). Also, disaster recovery (DR) was often behind production, infrequently tested and only just capable of keeping the lights on. No one could be confident that DR would work. Annual drills didn't really prove much.

## 1. Shared data center

Sharing data centers meant organizations could forget about the overheads of managing DC premises, while benefiting from space flexibility and a cost-sharing model. Most organizations would now have moved to several live data centers — sharing their regional or system load. DR could be addressed more readily. Infrastructure management was incentivized to adopt this model and own the migration.

However, there was still a heavy reliance on physical hardware and, in banking and capital markets, a strong desire to isolate systems to run on their own boxes. Hence, a reluctance to introduce new systems as that still meant new hardware. This often led to unwieldy system extensions and compromised architecture (the legacy of which can still be seen today).

It was common to see EOD reporting, analytical services, computational services and transactions all mixed in the same systems, which stemmed from a lack of infrastructure agility. It was also common to see data replicated everywhere, so departments could limit dependency on infrequent releases and manage their own change.

Administrators would build a cottage industry around protecting production — seemingly designed to prevent any change at all — so securing sign-off for new releases could also take months. This meant limiting the frequency of change, bundling new features for testing and release and, subsequently, very simple but big-impact changes could be delayed.

## 2. Virtualization

As systems ran on their own dedicated hardware, this meant that:

- Each system would have limited spare capacity for peak demand and future requirements
- Much of the infrastructure would be idle

Scalability issues were hard to address cost effectively.

Virtualization was the next step, separating virtual servers from physical hardware. This enabled multiple virtual servers to run independently on the same physical server, as if they were on their own box. Again driven by infrastructure departments, primarily, to optimize hardware use and lower operational cost, but it had another benefit. Virtual servers could be spun-up in minutes for new environments and ventures.

# IaaS (Infrastructure as-a-Service)



## Simple definition of cloud

Made possible by virtualization: **Self-service** and **on-demand** provision of **pay-as-you-go-metered** computing services, accessed over the **internet** from (massive) **resource pools** that provide **rapid elasticity**

## 3. Dev, test and burst in the cloud

Constraints on the provision of environments in on-prem data centers can really slow down model and systems development, testing and releases. Cloud could provide virtual networked compute and storage over the internet. It proved popular for development, testing and burst, and was often initiated by business users (citizen developers) and radical dev teams — especially in risk systems where there are inherent spikes in compute demand across production, test and development environments. The early adoption of cloud in these spaces advanced the DevOps agenda. It enabled developers to undertake tasks traditionally managed by infrastructure operations, ushering in Infrastructure as Code (IaC) with tools such as Terraform.

Initially, cloud adoption for production deployment in capital markets was slow, but now it's accelerating at pace. Many senior decision-makers used to feel that cloud just wasn't right for their organization and the slow uptake was compounded by infrastructure protectionism, misguided security and regulatory concerns, and no clear sponsorship and ownership of the agenda. Divisions between infrastructure and development, as well as run the bank and change the bank, meant the transformation agenda and its benefits were not aligned with departmental objectives, so organizations were not set for cloud transformation. This procrastination and stalled progress lasted almost a decade while change slowly precipitated from the banks' grass roots.

As we move into cloud-based virtual infrastructure, distinctions between engineers and administrators are disappearing. Organizations need different types of IT teams and help with the transformation.

Moving to cloud infrastructure services is straightforward from a technology perspective. After all, the hyperscalers want your business and will go the extra mile to make a lift-and-shift as easy as possible. If an organization is immature with IaC and deployment, there will be an opportunity to progress this, but the biggest challenges are in integrating new business processes into the organization. All hyperscalers provide a cloud adoption framework (CAF) to guide an organization on its journey to cloud.

With it being so easy to spin-up new infrastructure in the cloud, it's equally easy to accumulate, and lose track of, spiralling costs. To control and optimize costs, an organization should have good cloud governance and an efficient FinOps model that allocates costs, and balances self-policing with organizational oversight. The FinOps agenda will form an important part of any CAF, and it makes sense to leverage the experience of a hyperscaler's partner on how to set this up correctly.

Cloud might eventually see the end of the in-house data center and, while we're not at that point yet, we're already moving into the after data center (AD) era; beyond simple virtual compute, networks and storage, to Platform as-a-Service.



# PaaS

## (Platform as-a-Service)

A second major selling point of cloud is its platform services. These comprise managed infrastructure, databases, messaging systems and the like. As organizations move into these managed cloud platform services, the distinctions between system administrators and developers starts to blur if not disappear altogether. In some ways, we're going full circle and returning to the DIY days of old. The next generation of developers are expected to know these platform services, and microservices (container, API and mesh technologies). The DevOps agenda features tasks that were traditionally managed by system operations.

### 4. Containers and cloud deployment

Containers build on server virtualization. They isolate an application environment but utilize a shared operating system to enable more efficient use of the underlying hardware and much faster spin-up. Containers are set up using technology like Docker and Kubernetes can be used to manage container deployment and scalability. The efficiency and solid predictability of containers (for production, test and development environments) means organizations and developers naturally gravitate to the technology (the days of "but it worked in test" are gone). Across the whole of the capital markets technology space, institutions are now developing in containers — usually as part of a DevOps agenda — and actively moving legacy applications into a container stack. The payoff is such that most organizations want to get this done quickly and need additional developer bandwidth to do so. Cloud adoption accelerates the container agenda as, together, they're a perfect fit.



## 5. Cloud platform services

### Open-source software (OSS)

There's a strong relationship between cloud services and OSS, probably because:

- Hyperscalers want to make it easy to use their services, and OSS can offer standards, familiarity and portability
- Given OSS solutions already exist, it's a fast way to expand service offerings

While there might be altruistic reasons too, the relationship between cloud and OSS is self-reinforcing. As one becomes more popular, so does the other. Capital markets technology cannot escape this and while there are widespread benefits of adopting OSS across the technology stack, in the platform space, OSS has been a forced adoption. This has changed the perception of OSS. So, OSS is set to move up the stack from platform to widespread application services, all available in the cloud.

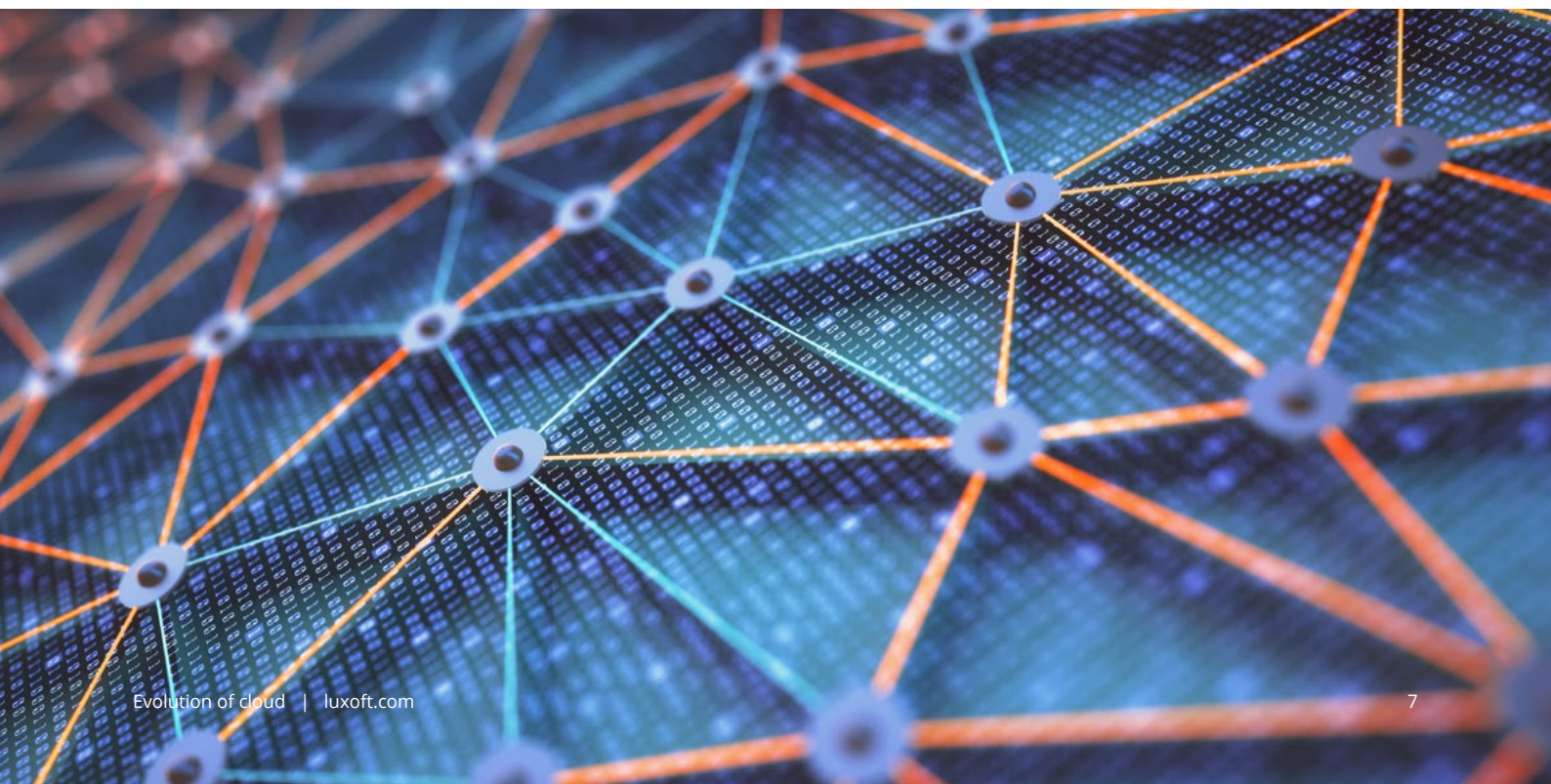
The availability of these services will drive players toward an API, event-driven and serverless architecture. Currently, the perceived performance of cloud services

is holding back progress in the low-latency, front-office space. But in the middle and back office, we're likely to see rapid evolution. iPaaS (Integration Platform as-a-Service) has emerged to tie business workflows together in this space.

### Serverless and FaaS

Whether running or not, containers still need virtual infrastructure. The next evolution in cloud platform services will be of those that only charge for runtime, being completely divorced from any user knowledge of the virtual hardware on which they're running (now tagged "serverless"). Functions as-a-Service (FaaS) is still a relatively new paradigm for the capital markets space, as technology requirements are generally quite heavy and these services are still focused on lightweight requirements. However, they lend themselves to event management and process integration, and are likely to be used for such things as customer onboarding, relationship management, trade lifecycle and other process-driven requirements.

Further, serverless functions are evolving into mini lightweight app environments.



## Platform services

Platform services are core services that a developer would typically integrate into an application; services like databases and messaging. In the cloud, platform

services don't require operational assistance, however, popular OSS implementations will be available as well as hyperscaler proprietary solutions.



### IDE

Integrated development environments and DevOps tools.



### IAM

Services for managing in-house and customer access to your application stacks. Including key management.



### Databases and analytics

Provision of SQL and noSQL datastores as services. Services evolve to provide scalability, search, analytics and data-science tools as natural extensions. Analytical and AI services are often key draws to the cloud, and hyperscalers invest in these services to boost differentiation.



### Messaging

Broadcast and point-to-point messaging services.



### API

Cloud is essentially a set of APIs and encourages developers to build applications the same way. Services are there for managing APIs (e.g., versioning and throttling). Mesh services are evolving.



### Monitoring and logging

These will be designed to integrate into a single pane and often developer services too, to enable inline debugging.

The choice can be overwhelming at first but, generally, there's an obvious fit for each organization. Cloud architects can socialize the right stack.



## Business services

Common business services such as email, are widely used in the cloud already. This will go much further.

As applications are built around services and APIs, the opportunity to integrate third-party business services becomes obvious and straightforward. If a third party offers the same functionality for a cheaper cost that's easily measurable or has features an organization needs, simply integrate it through an easy OSS API swap-out or hook.

The need to utilize services across hyperscalers and optimize costs will emerge. To date, most banks have partnered with a single hyperscaler, but this will change and organizations will need to embrace a multi-cloud operational environment, and a whole new set of governance standards and tooling.

# SaaS (Software as-a-Service)

## 6. Applications

Perhaps the journey ends at SaaS. Here, an organization can devolve the entire application stack either to a third party in a hyperscaler marketplace to run a standard service, or for a third party to manage and advance a customized solution in the cloud.

Luxoft has been leading the way through the whole journey, from the early days of distributed grid computing to burst to cloud, in DevOps and cloud-first bespoke builds. We are now perfectly positioned to provide a fully managed SaaS across capital markets (plus banking and insurance). In capital markets this includes:

- Client Lifecycle Management as-a-Service
- Trading Systems as-a-Service (TSaaS)
- Treasury Systems as-a-Service
- Investment Systems as-a-Service

Cloud began life as a provider of infrastructure compute and storage. It evolved to embrace platform, development and data services, and is now capable of delivering a fully managed solution for business services.



## About **the author**



### **Marc Maynard**

Senior Director

Marc Maynard has been delivering technology, mainly in capital markets and banking, for 30 years. He's performed bespoke builds and package implementations from front office to back; predominantly trading, P&L and risk platforms. Now at Luxoft he advises on architecture and shapes client engagements.

If you'd like to find out more visit [luxoft.com/capital-markets](https://luxoft.com/capital-markets) or consult one of our experts at [financialservices@luxoft.com](mailto:financialservices@luxoft.com)

### **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](https://luxoft.com)