CIBC INNOVATION

Building the Optimal Architecture for Open Banking

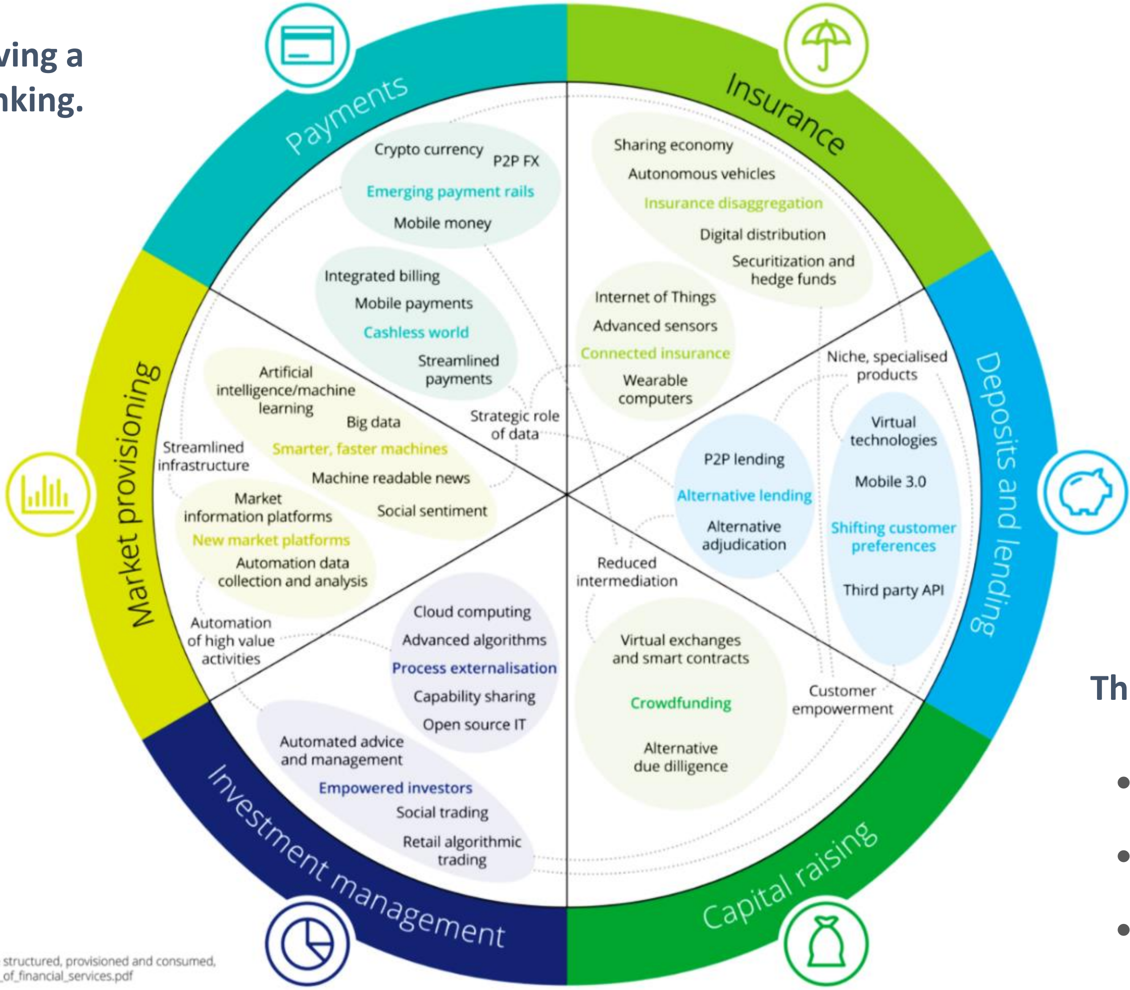
Presented @ DRIE Symposium

Toronto, Ontario, Canada / June 18th, 2019

Banking that fits your life. CIBC

Banking Evolved | Banks are quickly becoming technology companies.

Advances in new technologies are having a dramatic impact on every facet of banking.



Source: The Future of Financial Services – How disruptive innovations are reshaping the way financial services are structured, provisioned and consumed, World Economic Forum and Deloitte, June 2015, p.12. See also: http://www3.weforum.org/docs/WEF_The_future__of_financial_services.pdf

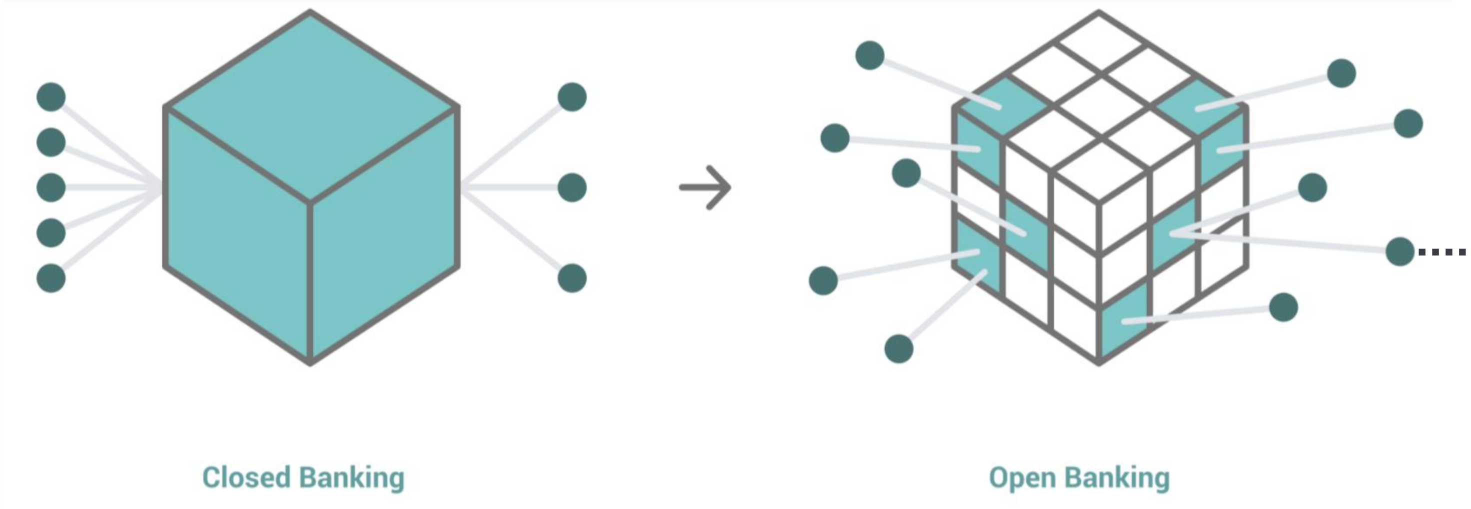
There are three main pressures driving this:

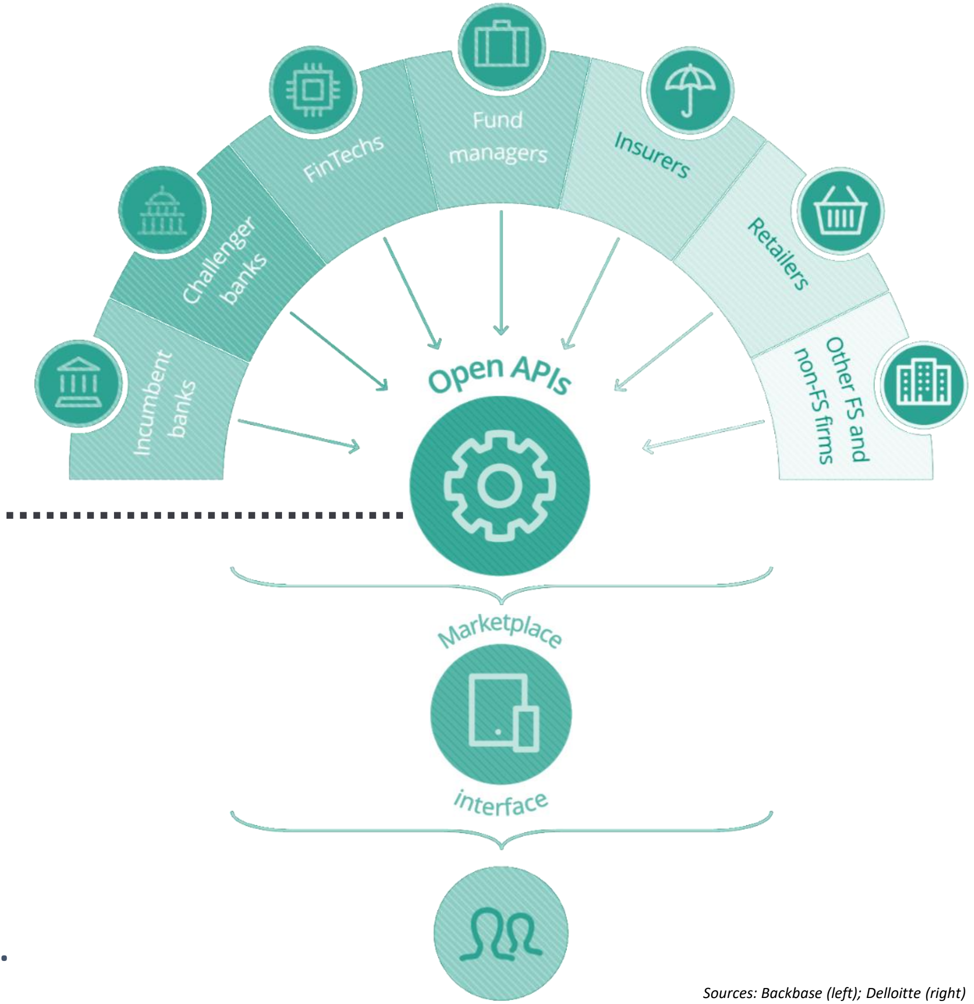
- **Rising Customer Expectations**
- **Competition from FinTechs**
- **Regulatory Environment**

To drive *innovation*, banks have to learn all about Agile, Dev-Ops, Cloud & (of course) APIs.

The Rise of the API | APIs are the pipelines that power Open Banking

In response to the three main pressures , banks are rapidly adopting APIs, in an attempt to decompose their business into re-usable chunks.





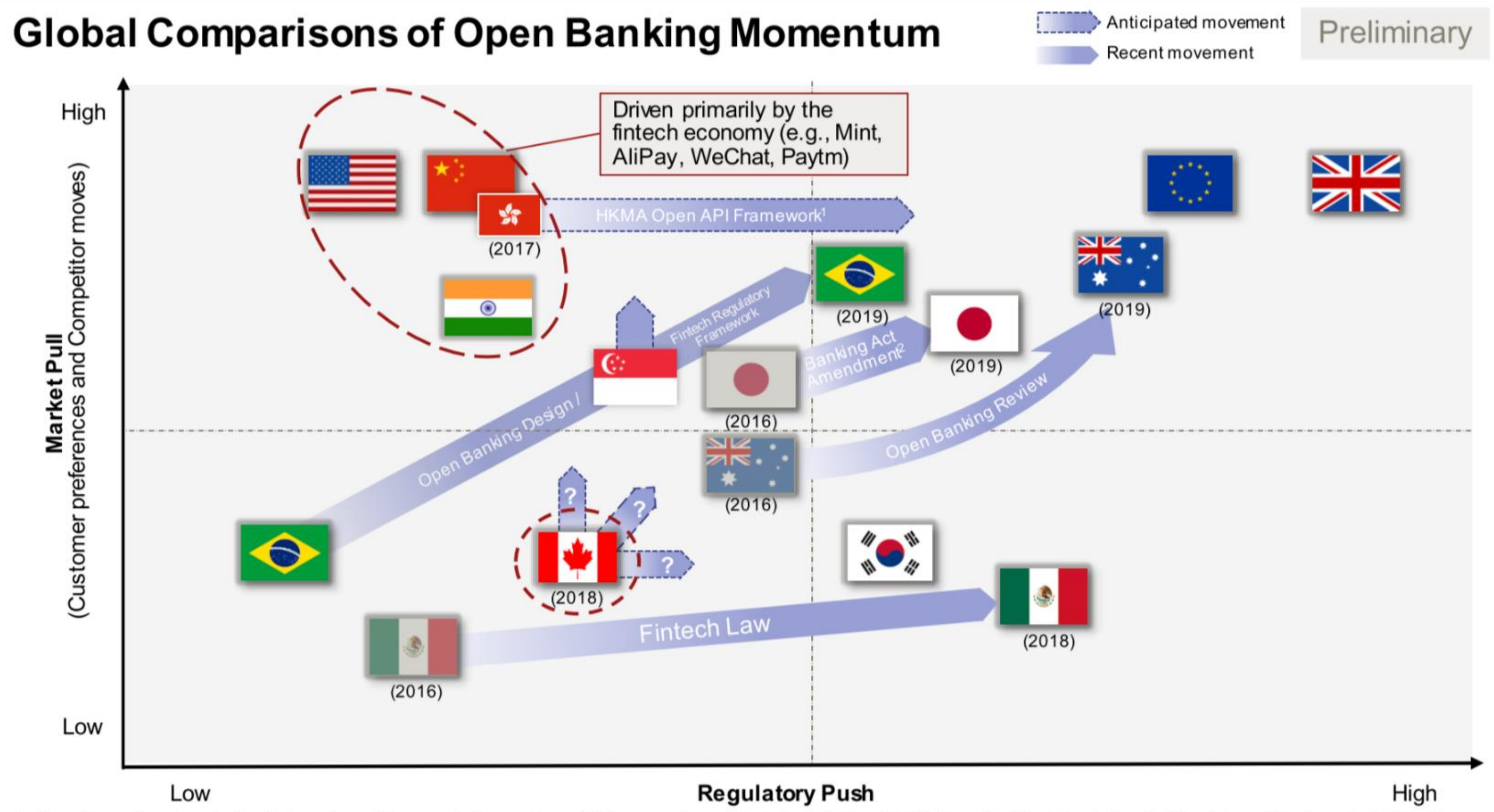
These chunks of functionality can be shared inside the bank, but can also be shared beyond the bank, which is what has come to be called Open Banking.

Sources: Backbase (left); Deloitte (right)

However, Open Banking means different things to different people in different places.

Global Landscape| Some regions regulate and some are market-driven.

Global Comparisons of Open Banking Momentum

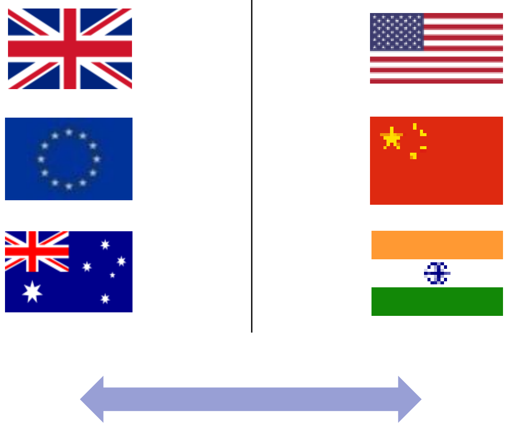


1. Hong Kong Monetary Authority issued a public consultation on Open API framework modeled after PSD2 and CMA Open Banking Regulation. It will only target the largest retail banks
2. The 2017 amendments to the Banking Act in Japan will require financial institutions to make efforts to enable Open API within two years from the date of enforcement; the amendments are to come into effect on June 2017

Source: AT Kearney

Regulation-Driven
(Push)

Market-Driven
(Pull)



While different in key ways, note that both approaches have the <u>same goals</u>:

- **Competition**
- **Innovation**
- **Transparency**

These two approaches end up greatly affecting the focus of the solution architecture.

Architecture Focus | Regulations drive APIs, but markets drive speed.

Regulatory mandates force banks to publish Open APIs and support an ecosystem of TPPs, demanding strong capabilities around API management…

… However, nothing in the regulations forces them to modernize their legacy systems, drastically slowing down delivery.

By contrast, market-driven solutions, unencumbered by regulation, have focused more on accelerating delivery speed.

Regulation-Driven
Market-Driven

DEVELOPER EXPERIENCE
Presentation Layer

BUSINESS CAPABILITIES
Logic Layer

OBJECTS & MODELS
Data Layer

ENVIRONMENTS
Infrastructure Layer

Architectural Concerns
(Full Stack)

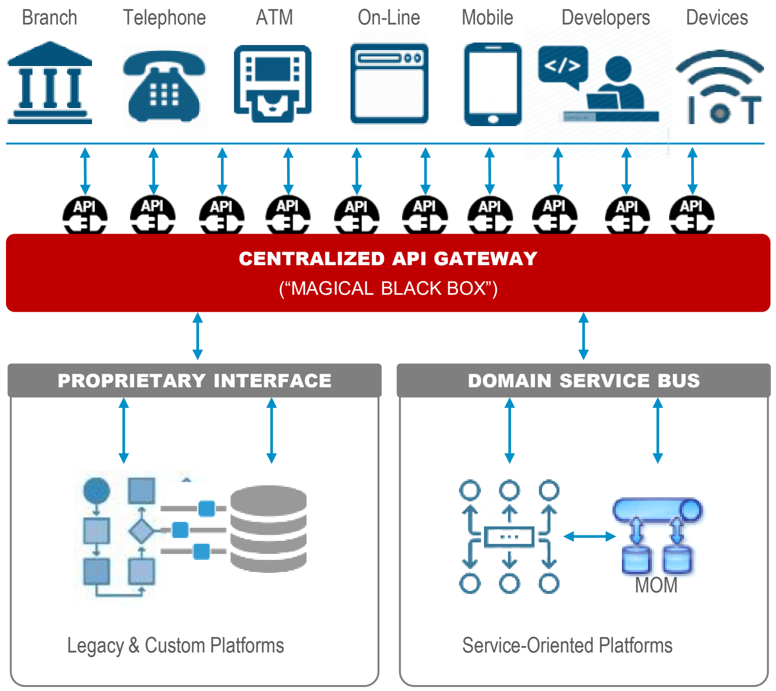
- **Open APIs & Open Data**
- **Developer Experience**
- **Common Standards**
- **Agile Requirements Definition**
- **Consumer-Driven Contracts**
- **Precise scaling and enhancement**
- **Distributed Data using Events**
- **Unstructured "Big" Data Lakes**
- **Automated DevOps deployment**
- **Public cloud native (containers)**
- **Continuous Improvement (CI/CD)**

To truly support Open Banking, a bank needs both API management *and* delivery speed.

The Role of API Gateways | Beware the "Magical Black Box" approach.

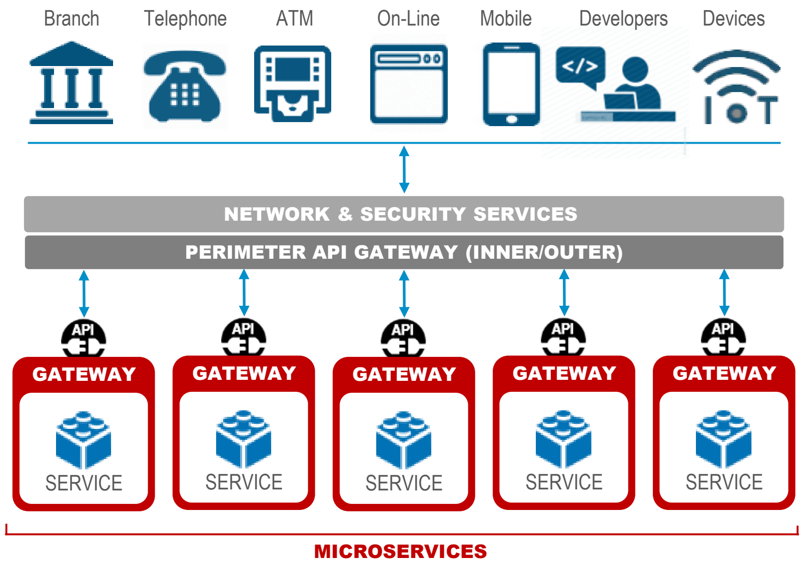
Centralized Gateway

All cross-cutting concerns and isolation are in a single, "smart" layer.



Distributed Gateway

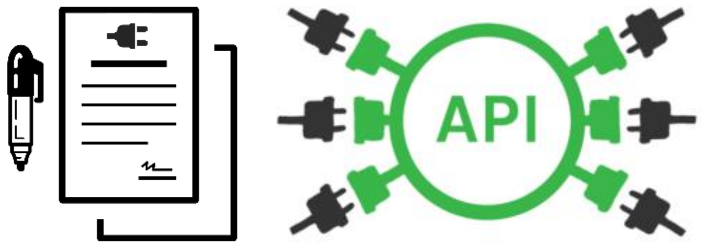
Cross-cutting concerns are distributed to the individual end-points.



A distributed approach pushes all gateway functions down to each individual *microservice*.

Microservices Explained | APIs & microservices are *not* the same thing.

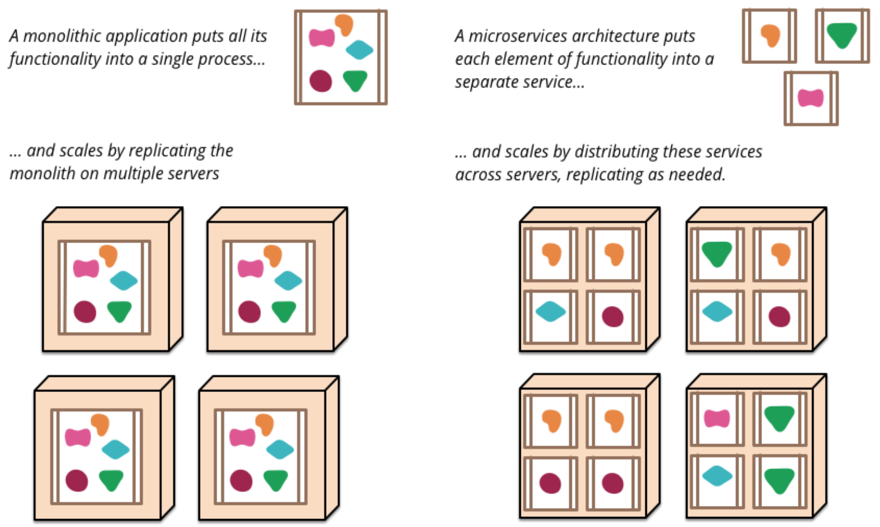
APIs are the contract



Good contracts based on modern standards (i.e. REST, JSON, etc.) make integration easier & drastically improve the Developer experience on the Consumer side…

… However, the same API could be implemented on *any* back-end, and the wrong one (i.e. legacy systems) will greatly hamper agility & speed, and therefore innovation.

Microservices are the implementation



A monolithic application puts all its functionality into a single process…

… and scales by replicating the monolith on multiple servers

A microservices architecture puts each element of functionality into a separate service…

… and scales by distributing these services across servers, replicating as needed.

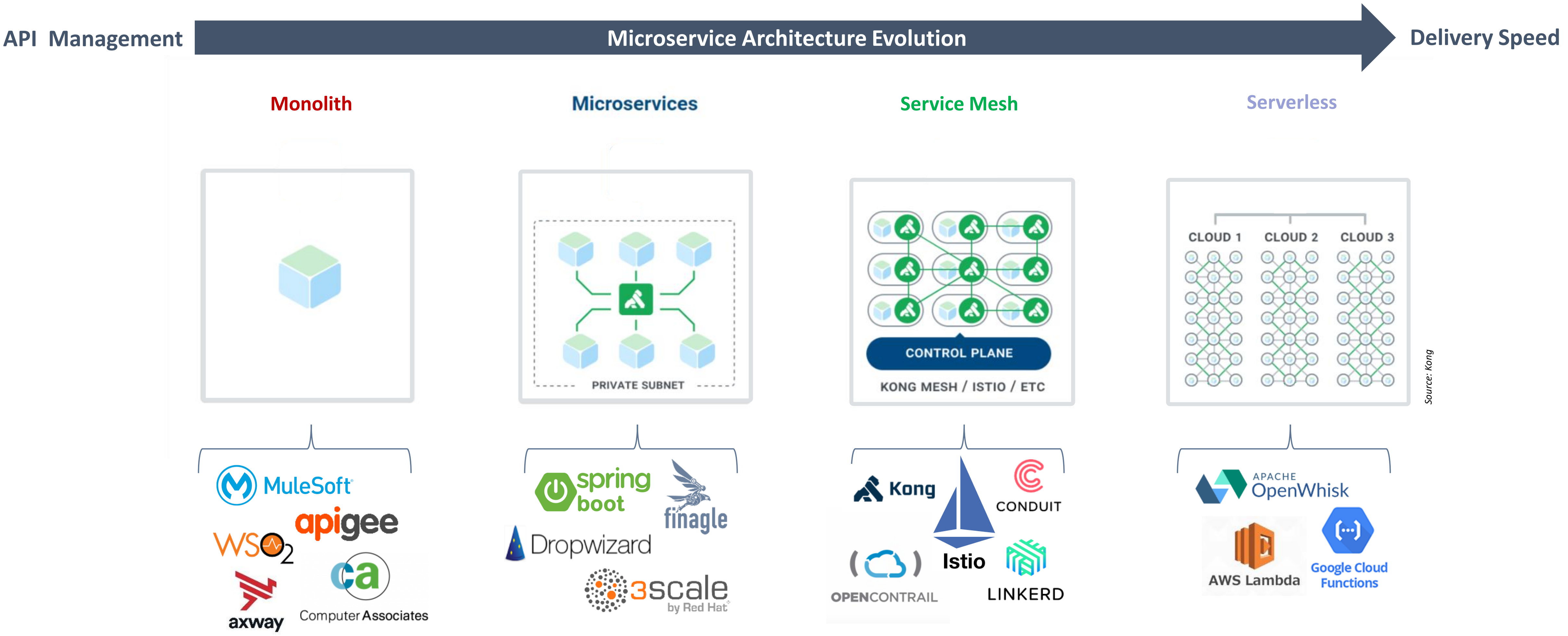
Source: martinfowler.com

Microservices allow you to decompose monolithic applications and enable delivery speed because they are…

- Independently Upgradeable
- Precisely Scalable
- Completely Portable

APIs make integration easier, but *only* microservices lead to high-speed release cycles.

MSA Roadmap| Microservices Architecture (MSA) is evolving rapidly.



API Management

Microservice Architecture Evolution

Delivery Speed

Monolith

Microservices

Service Mesh

Serverless

Source: Kong

The rise of *service mesh* will increase market volatility, and early adopters will gain a lead.

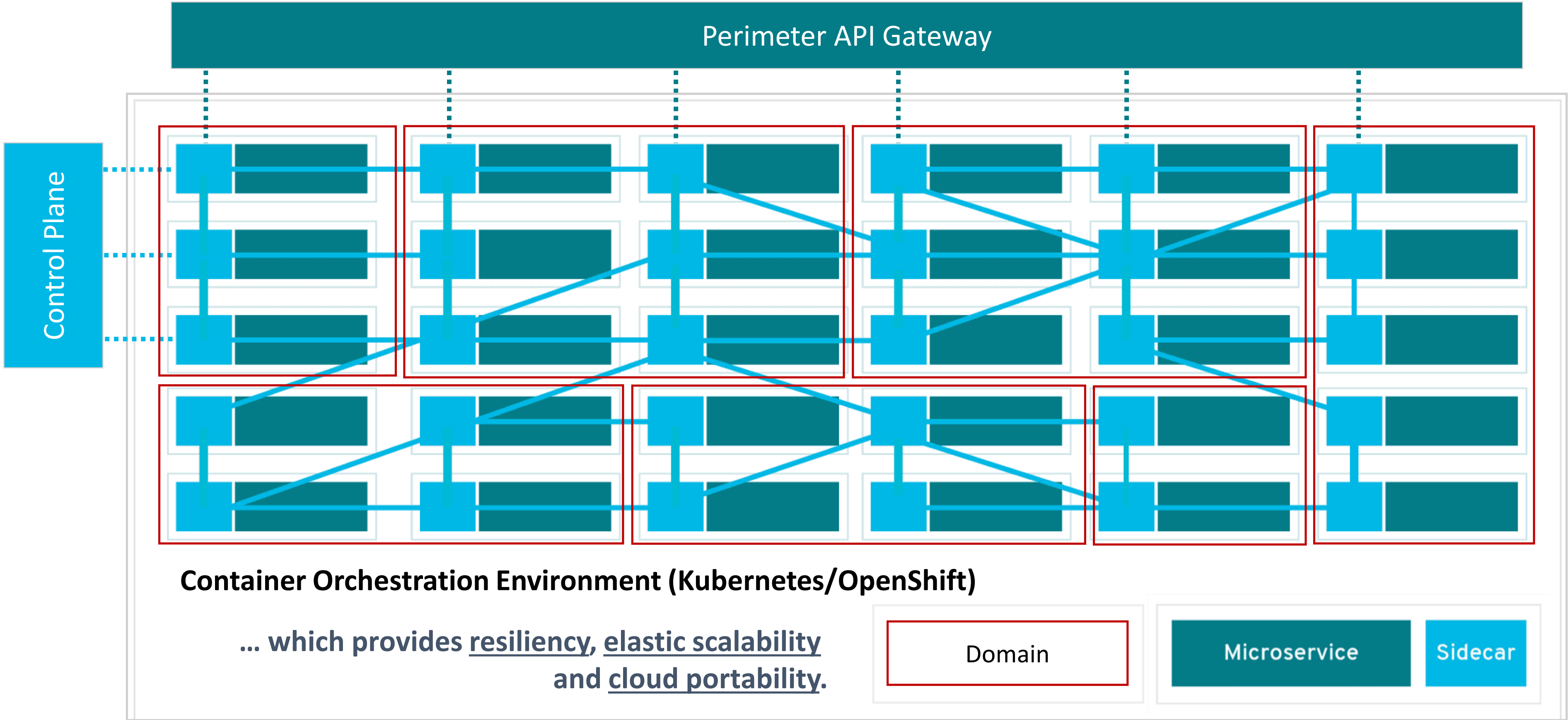
Optimal Architecture | The service mesh is perfect for Open Banking.

The service mesh acts as a uniform infrastructure for direct service-to-service communication (via APIs).

It utilizes lightweight proxies deployed side-by-side or together with the services known as sidecars.

It ensures consistent handling of cross-cutting concerns in a fully distributed manner, using a control plane:

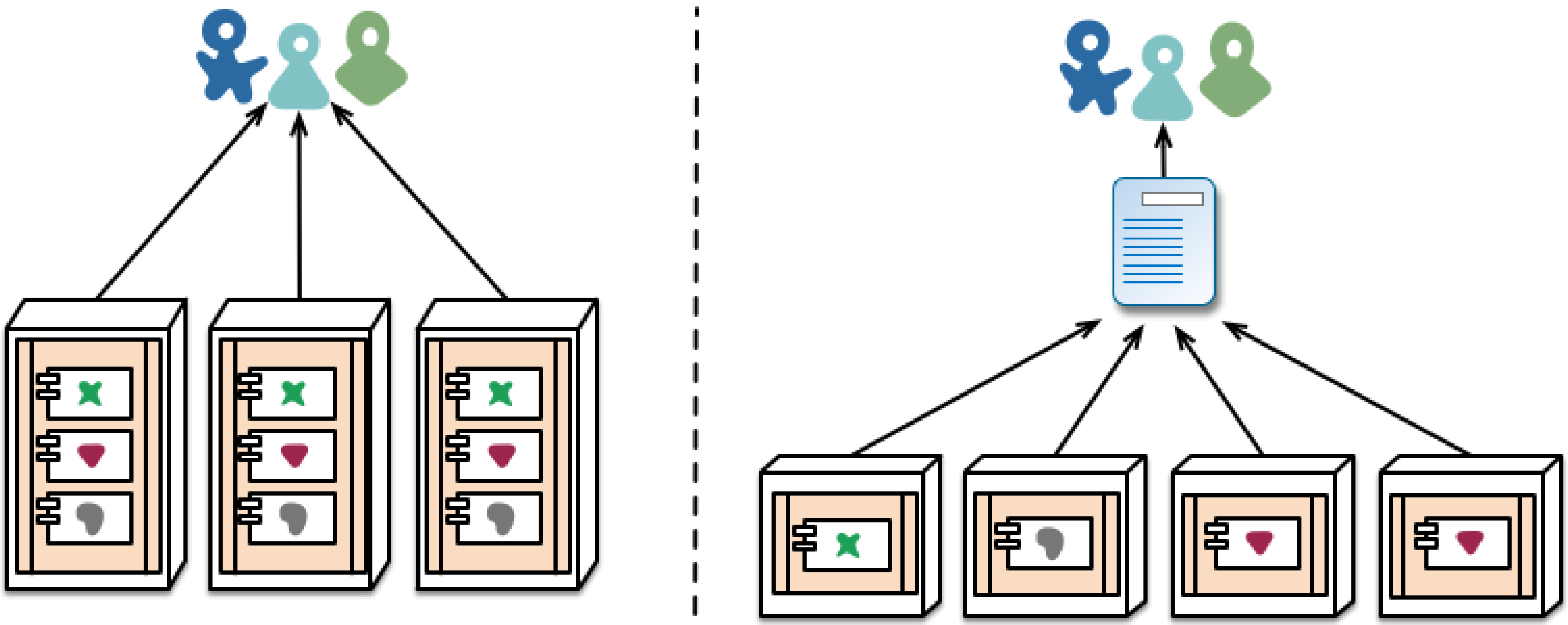
- Consistent Routing
- Security
- Logging
- Monitoring



… which provides resiliency, elastic scalability and cloud portability.

The service mesh offers precise control & visibility, while supporting speed & innovation.

Ideal for Disaster Recovery | Microservices are built for resiliency.



monolith - multiple modules in the same process

microservices - modules running in different processes

From a DR/BCM perspective, microservices running on a Container Orchestrator like Kubernetes/OpenShift are a vast improvement over J2EE.

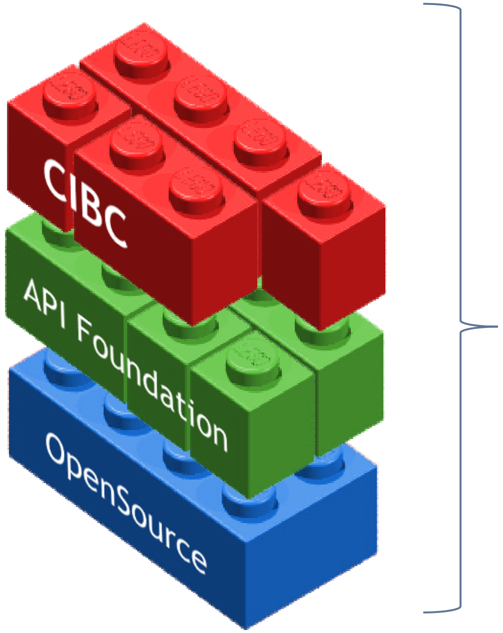
J2EE/legacy applications are monolithic, so they are scaled and managed by replicating copies on multiple servers.

Microservices break down the monolith and distribute the functions across servers, scaling replicas up and down dynamically based on load. They are so resilient that some actively attack their own Production environment (Netflix's Simian Army).

Microservices are self-contained, auto-managed & cloud-native, greatly easing DR/BCM.

CIBC Case Study | Building a service mesh API platform for a bank.

Instead of gambling on one of many vendor platforms in a highly volatile market, CIBC built our in-house platform, the API Foundation on <u>open-source</u>, <u>cloud native</u> technology.



API FOUNDATION

We then subjected our API platform to intense, independent 3rd party evaluation. So far the evaluations have been very positive.

DXC CAPCO redhat ca technologies FORRESTER

Then we took our framework to Europe…

NORDIC APIS

“**Great framework** - thoughtful, holistic and **quite advanced**”

“Uses **open source** intelligently”

“**Security** model is built-in and **highly robust**”

“Building custom as a **hedge against a volatile market** makes sense right now”

There are three main reasons we took a <u>build posture:</u>

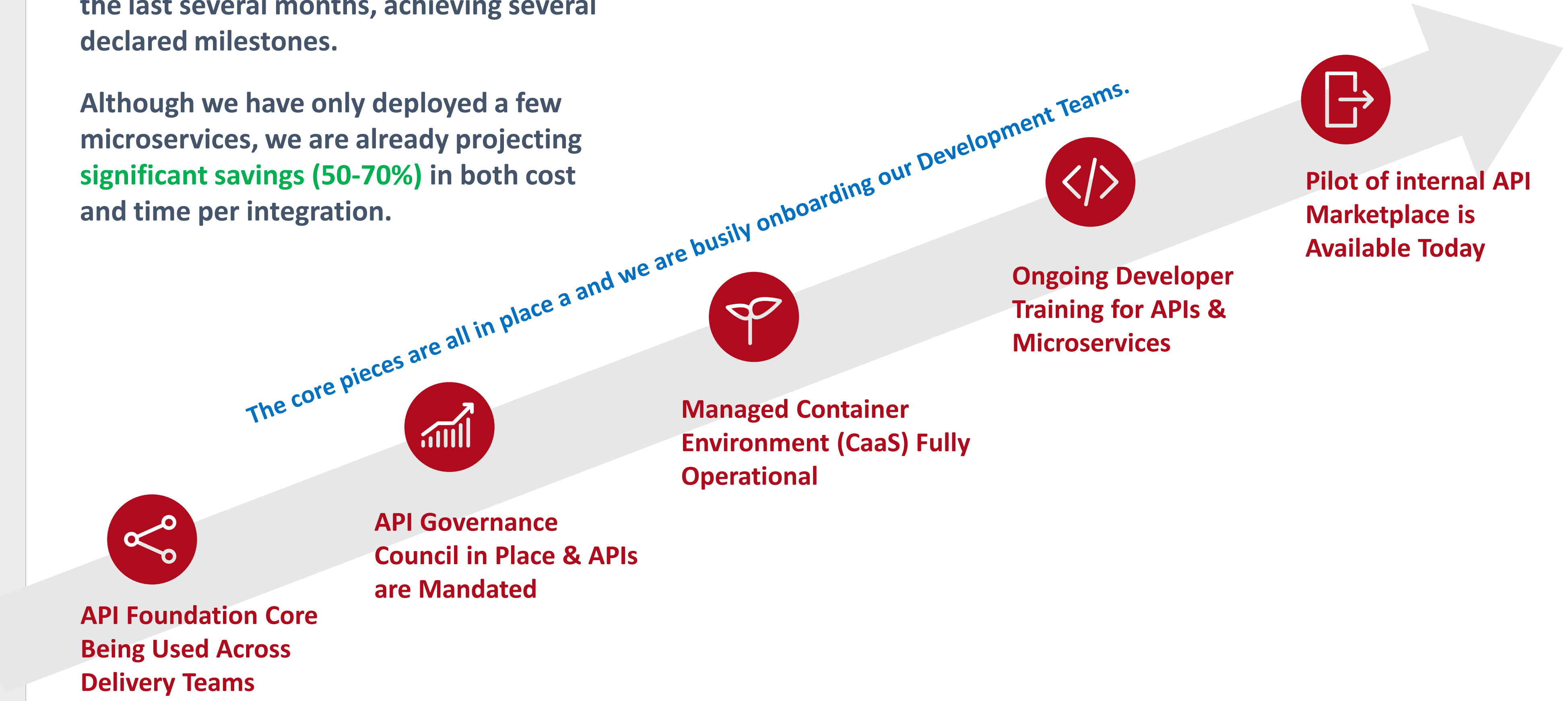
- **Hedge against a volatile market**
- **Steer the technology to suite our needs**
- **Develop critical internal skills**

The award-winning API Foundation is the first service mesh built specifically for banks.

Real Results | The API Foundation is now in full swing across the bank.

We have taken great strides forward over the last several months, achieving several declared milestones.

Although we have only deployed a few microservices, we are already projecting significant savings (50-70%) in both cost and time per integration.



The core pieces are all in place a and we are busily onboarding our Development Teams.

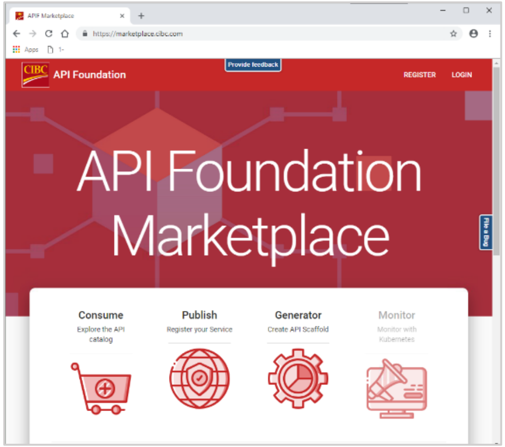
API Foundation Core Being Used Across Delivery Teams

API Governance Council in Place & APIs are Mandated

Managed Container Environment (CaaS) Fully Operational

Ongoing Developer Training for APIs & Microservices

Pilot of internal API Marketplace is Available Today



API Marketplace

https://marketplace.cibc.com

CIBC API Foundation

Provide feedback

REGISTER LOGIN

API Foundation Marketplace

Consume
Explore the API catalog

Publish
Register your Service

Generator
Create API Scaffold

Monitor
Monitor with Kubernetes

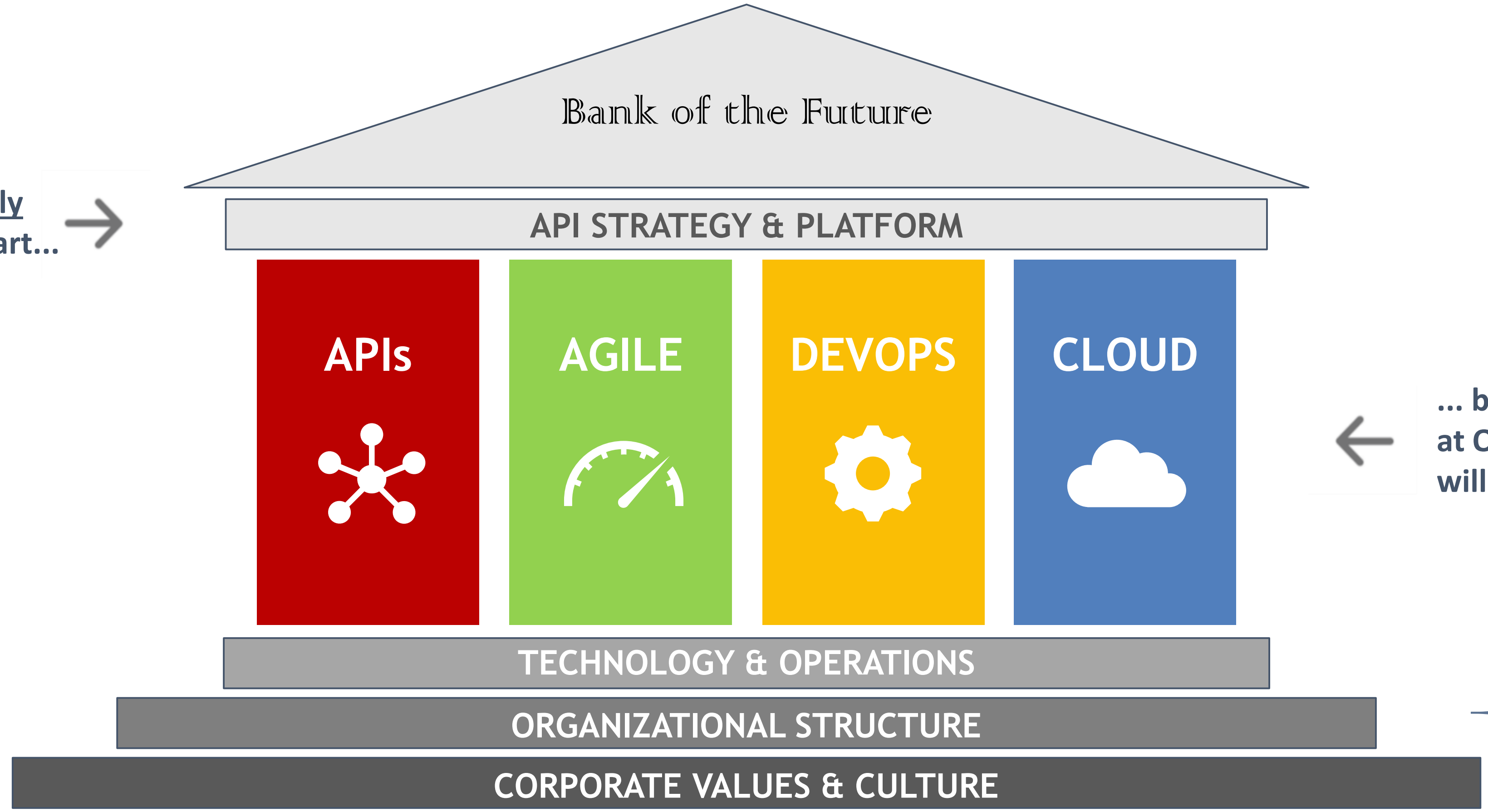
Did the APIF Foundation really win awards? Yes!

- Recognized by Nordic APIs as a leader in MSA
- 2018 EA Award from Forrester & InfoWorld
- 2019 Celent Model Bank Award for Integration

While we have made progress towards a clear vision, there is still a long way to go.

Bank of the Future | Success demands a technological balancing act.

Open Banking only focuses on this part...



Bank of the Future

API STRATEGY & PLATFORM

APIs

AGILE

DEVOPS

CLOUD

TECHNOLOGY & OPERATIONS

ORGANIZATIONAL STRUCTURE

CORPORATE VALUES & CULTURE

... but to be successful at Open Banking, you will need all of these...

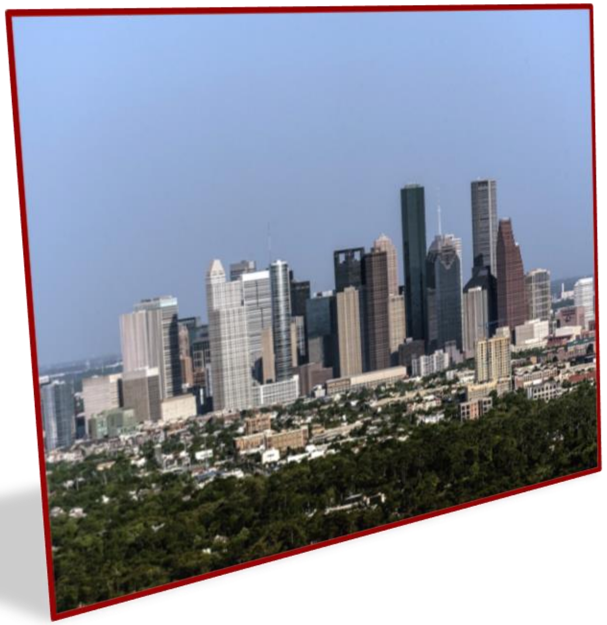
... and, crucially, you will have to tackle structural & cultural challenges as well.

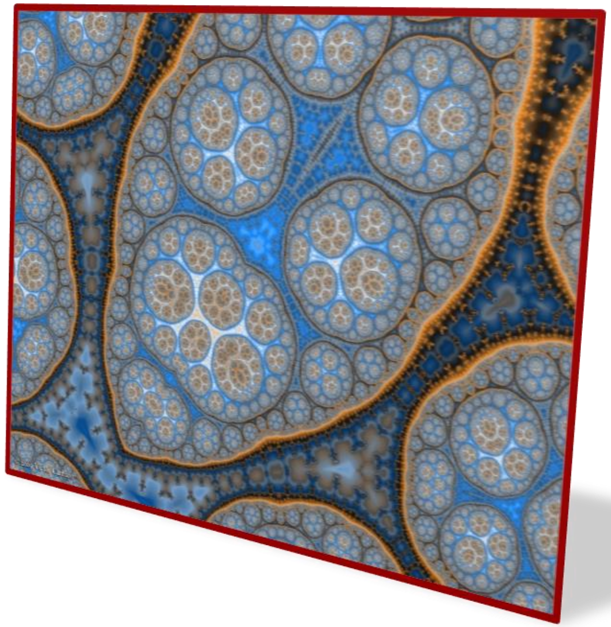
The pillars give you speed, but only a culture that *embraces change* drives true innovation.

Build for Change| The metaphors used for IT systems are changing.

Industrial Metaphors

- Bridges
- Buildings
- Cities





Biological Metaphors

- Cells
- Molecules
- Organisms

> *"There is no design at the beginning. You begin by coding a small amount of functionality, adding more functionality, and letting the design shift & shape."*

Martin Fowler, Software Design Guru, Thoughtworks

Systems that can *evolve* quickly are critical to the delivery of Open Banking technology.

CIBC INNOVATION

Thank you.

QUESTON & ANSWER

Eyal Sivan, Senior Director, Enterprise Architecture

Eyal.Sivan@CIBC.com

Banking that fits your life. CIBC