Microservices

Improve time to market with microservices

By Alberto Valverde Escribano
Senior R&D Engineer

“Clients are trying to modernize their applications to keep up with the rate of change… in the past, operations dictated how applications were written… people spent years writing monolithic applications in which a lot of application function was packaged inside the app… now they are seeing the difficulty developers are having in adding new features to respond to changing marketing demand.”

What are microservices?

Microservices is an architecture style which prescribes building large complex software applications using many small microservices. These microservices are narrowly focused, independently deployable, loosely coupled, language agnostic services fulfilling a business capability. These multiple microservices communicate with each other using language-agnostic APIs such as REST.

These microservices are applications in themselves and are often owned by small teams. Unlike the normal practice, the team which coded the microservices is also responsible for its support.

Why microservices?

Business drivers like agility, better reliability, improved scalability, security has forced architects to consider new paradigms like cloud computing and microservices. The traditional monolithic applications suffer from challenges like

a. Difficult to develop: The more is the code base, the more difficult is it to understand and maintain.
b. Difficult to test: One single change can affect multiple sub units forcing much larger testing effort. The testers should also be aware of the various code inter dependencies.
c. Slower to adapt: Changing even a single aspect of the application requires the entire code base to be affected, thus making it much slower to change and error prone.

In contrast, microservices, by definition itself are smaller manageable units of business capabilities which are easy to develop, test and maintain and also faster to adapt.

MICROSERVICES ARCHITECTURE

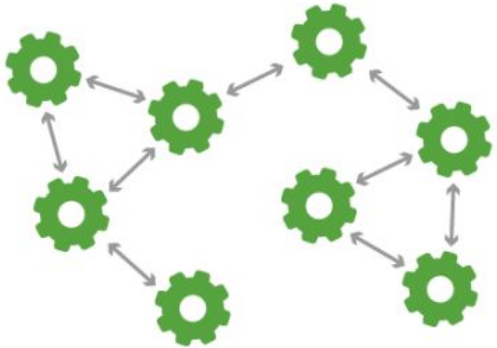
2000's

SERVICE ORIENTED ARCHITECTURE

SOA based applications are compromised of more loosely coupled components that use an Enterprise Services Bus messaging protocol to communicate between themselves.

2010's

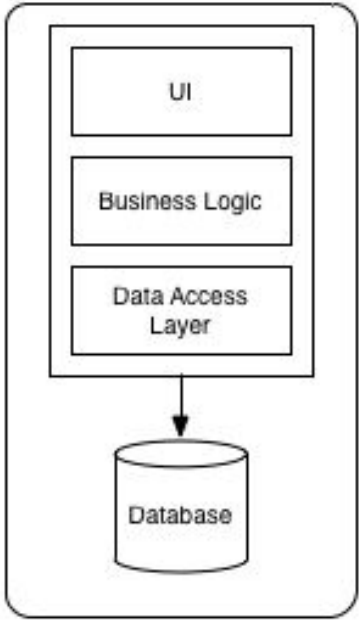
MICROSERVICES ARCHITECTURE



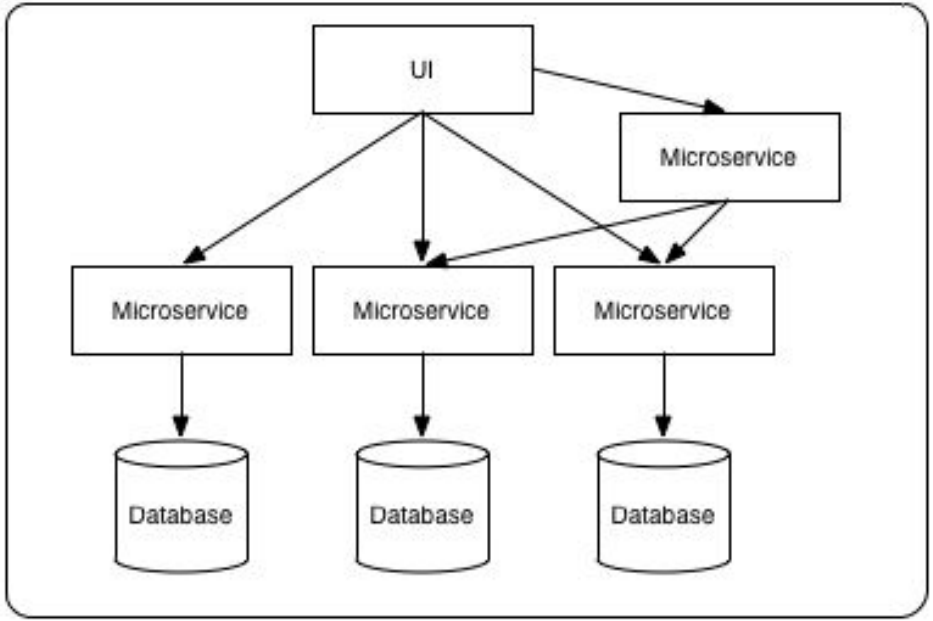
Microservices are a number of independent application services delivering one single functionality in a loosely connected and self-contained fashion, communicating through light-weight messaging protocols such as HTTP, REST or Thrift API.

MSA VS SOA

SERVICE ORIENTED ARCHITECTURE	MICROSERVICES ARCHITECTURE
Maximizes application service reusability	Focused on decoupling
A systematic change requires modifying the monolith	A systematic change is to create a new service
DevOps and Continuous Delivery are becoming popular, but are not mainstream	Strong focus on DevOps and Continuous Delivery
Focused on business functionality reuse	More importance on the concept of "bounded context"
For communication it uses Enterprise Service Bus (ESB)	For communication uses less elaborate and simple messaging systems
Supports multiple message protocols	Uses lightweight protocols such as HTTP, REST or Thrift APIs
Use of a common platform for all services deployed to it	Application Servers are not really used, it's common to use cloud platforms
Use of containers (such as Docker) is less popular	Use of containers (such as Docker) is less popular
SOA services share the data storage	Each microservice can have an independent data storage
Common governance and standards	Relaxed governance, with greater focus on teams collaboration and freedom of choice



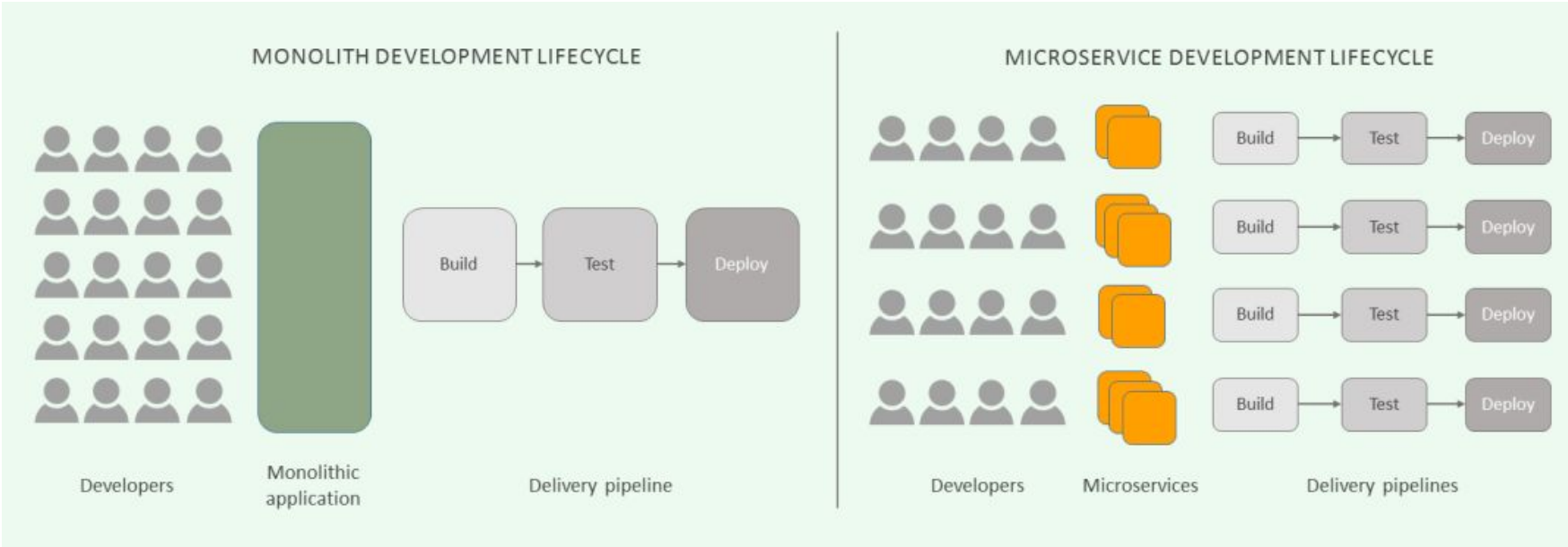
Monolithic Architecture

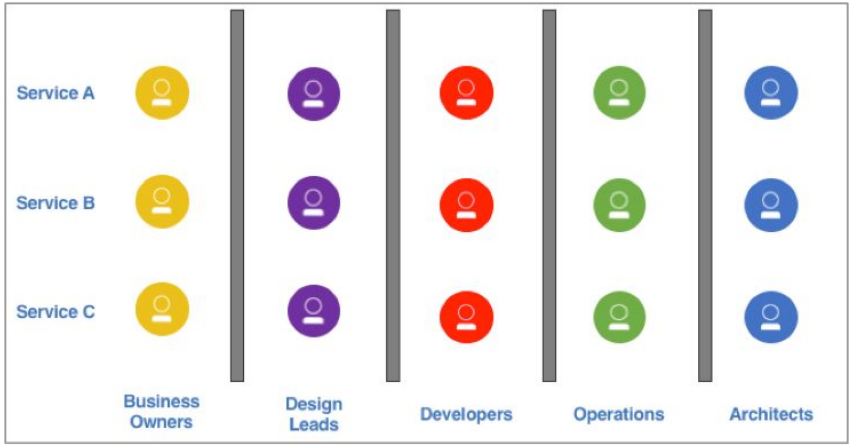


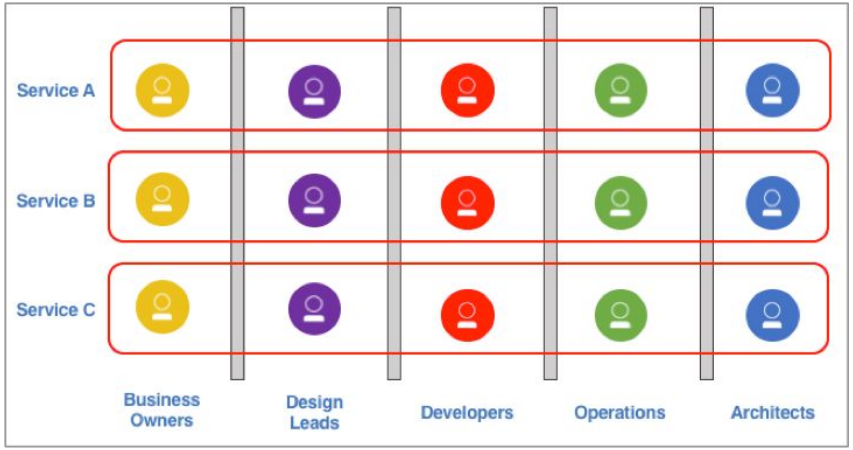
Microservices Architecture

Abandon the old departmental structure and focus teams around certain microservices.

This means that each of these teams will consist of various members with different skill sets such as system analysts, UX/UI designers, backend & frontend developers, etc. This way, the teams are responsible for their project (microservice) from end to end – from development and deployment to operations, monitoring, and management. This, in turn, will increase their motivation to create the product they feel their own.







Understand your culture and skill set

Microservices architectures can only succeed when teams have the power to own the complete software development and operations lifecycle. Building cross-functional teams representing all roles and responsibilities is fundamental in implementing microservices-based architectures. Everyone—from design to development to operations to business owner—works closely together and is often collocated.

Since every stakeholder is represented in the team by design, development, and operations, work can move forward more quickly, efficiently, and with clear focus on improving user experience to achieve business goals.

Thanks you for attending!

Contact

Phone: +34 608 837 254

Email: valveraa@essilor.es

Online: https://sites.google.com/essilor.com/creativity/