**What are Micros Services**

In software development domain, microservices is a new architectural approach in which a single application is composed of many small independent modules or services modeled around a business domain. In this approach each independent module or services is self-contained and implements a single business capability. It makes application easier to build, much easier to expand and scale.

**Monoliths : A Traditional Approach**

Monolith approach is a traditional way of building enterprise applications which has become more complex as application gets expand. A monolithic application is built as a single unit contains all the code for all business activities an application performed. Enterprise Applications are built in three parts: a database (consisting of many tables usually in a relational database management system), a client-side user interface (consisting of HTML pages and/or JavaScript running in a browser), and a server-side application. This server-side application will handle HTTP requests, execute some domain-specific logic, retrieve and update data from the database, and populate the HTML views to be sent to the browser. It is a monolith – a single logical executable. In a monolith, server side application logic, front end , client side logic, background jobs, etc, are all defined in the same massive code base. To make any modifications to the system developer will build and deploy the entire stack all at once.

**Differences Between Traditional Architecture and Microservices Architecture**

In monolithic approach enterprise application is build as a single unit that contains all code. While in microservice architecture , applications are build into multiple independent modules that works together seamlessly. These modules are individually responsible for performing precisely defined, standalone tasks and communicate with each other through simple, universally accessible application programming interfaces (APIs). In order to understand the difference between monolithic and microservices approach, consider an E-commerce application as shown in figure below.

A screenshot of a social media post

Description automatically generated

Figure 1 Monolithic Architecture vs. Microservices Architecture

It is observed from the above diagram that in monolithic approach all features were under a single instance of sharing a single database. But with microservices approach each feature was allotted a different microservice, handling their own data, and performing different functionalities.

**Key Advantages of Micro Services**

1. *Independent Development* – All microservices can be easily developed based on their individual functionality
2. *Independent Deployment* – Based on their services, they can be individually deployed in any application
3. *Fault Isolation* – Even if one service of the application does not work, the system still continues to function
4. *Mixed Technology Stack* – Different languages and technologies can be used to build different services of the same application
5. *Granular Scaling* –  Individual components can scale as per need, there is no need to scale all components together
6. *Increased developer productivity -* New developers can get up to speed rapidly, since it’s easier to understand a small, isolated piece of functionality than an entire monolithic application.
7. *Better alignment of developers with business users.* Since microservice architectures are organized around business capabilities, developers can more easily understand the user perspective and create microservices that are better aligned with the business.
8. *Future-proofed applications-* When innovations happen and new or updated technology disrupt your software development process, microservice architectures makes it easier to respond by replacing or upgrading the individual services affected without impacting the whole application.

**Example: Shopping Cart Application**

When you open a shopping cart application, all you see is just a website. But, behind the scenes, the shopping cart application has a service for accepting payments, a service for customer services and so on.  Suppose that developers of this application have created it in a monolithic framework as given in figure below.

A screenshot of a computer

Description automatically generated

Figure 2 Monolithic Architecture

So, all the features are put together in a single code base and are under a single underlying database. Now, let’s suppose that there is a new brand coming up in the market and developers want to put all the details of the upcoming brand in this application. Then, they not only have to rework on the service for new labels, but they also have to reframe the complete system and deploy it accordingly. To avoid such challenges developers of this application decided to shift their application from a monolithic architecture to microservices. Microservices architecture of shopping cart application is depicted in the diagram below.

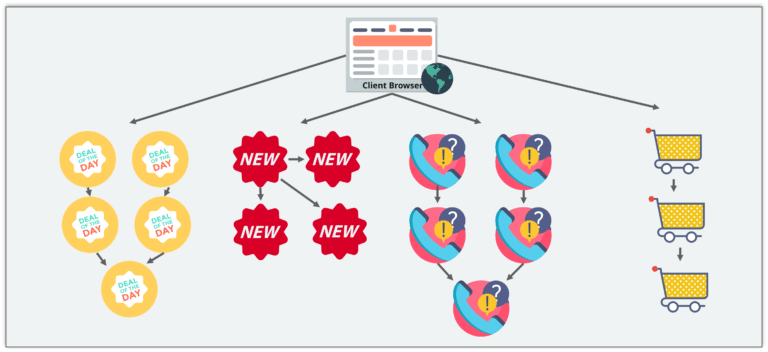


Figure 3 Microservices Architecture

This means that developers don’t create a web microservice, a logic microservice, or a database microservice. Instead, they create separate microservices for search, recommendations, customer services and so on.

This type of architecture for the application not only helps the developers to overcome all the challenges faced with the previous architecture but also helps the shopping cart application to be built, deployed, and scale up easily.

## ****Companies using Microservices****

Name of few companies that are using Microservices to build applications, are as under:

1. Amazon
2. Netflix
3. GILT
4. UBER
5. Twitter
6. Theguardian
7. Paypal
8. Ebay

## ****Key Enabling Technologies****

### Any modern tool or language can be used in a microservices architecture, there are a handful of core tools that have become essential and borderline definitional to microservices:

### Containers, Docker, and Kubernetes

**Reference**

<https://www.edureka.co/blog/what-is-microservices>

<https://www.ibm.com/cloud/learn/microservices#toc-what-are-m-4NBNe-ID>

<https://www.redhat.com/en/topics/microservices/what-are-microservices>

<https://martinfowler.com/articles/microservices.html>

\*\*\*\*\*