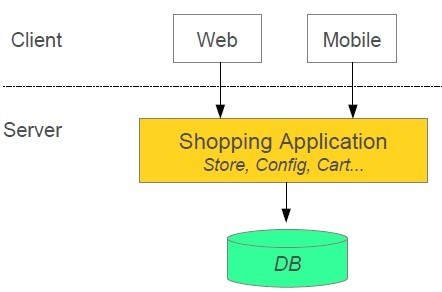
What are Microservices?

**Microservices are an architectural style that develops a single application as a set of small services. Each service runs in its own process. The services communicate with clients, and often each other, using lightweight protocols, often over messaging or HTTP.**

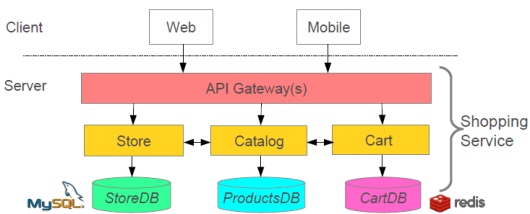
### Shopping system without Microservices (Monolith architecture)

In this architecture we are using Monolith architecture i.e. all collaborating components combine all in one application.



### Shopping system with Microservices

In this architecture style, the main application divided into a set of sub-applications called microservices. One large Application divided into multiple collaborating processes as below.

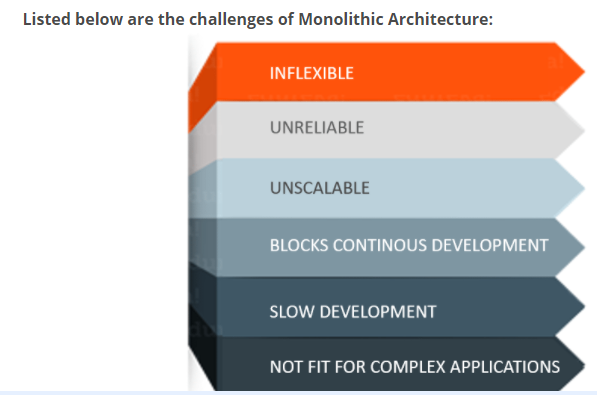


## **Why are Microservices used?**

**Microservices were used to overcome the challenges of monolithic architecture that prevailed initially in the market and also enables you to deploy independent services.**

**How Does Monolith Architecture Looks Like?**

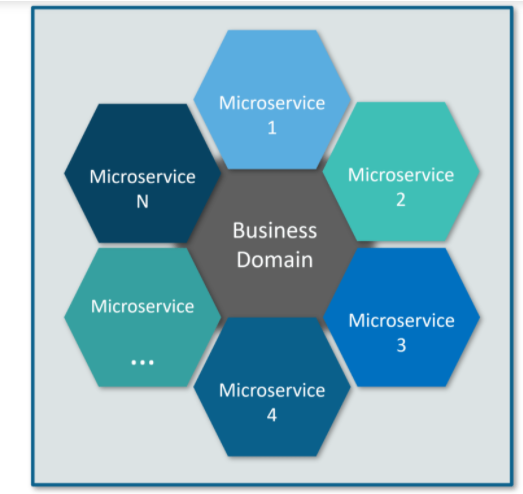
In layman terms, you can say that its similar to a big container wherein all the software components of an application are assembled together and tightly packaged.



**Figure 1:** Challenges of Monolithic Architecture

* **Inflexible –** Monolithic applications cannot be built using different technologies
* **Unreliable –** Even if one feature of the system does not work, then the entire system does not work
* **Unscalable –** Applications cannot be scaled easily since each time the application needs to be updated, the complete system has to be rebuilt
* **Blocks Continuous Development –** Many features of the applications cannot be built and deployed at the same time
* **Slow Development –** Development in monolithic applications take lot of time to be built since each and every feature has to be built one after the other
* **Not Fit for Complex Applications –**Features of complex applications have tightly coupled dependencies

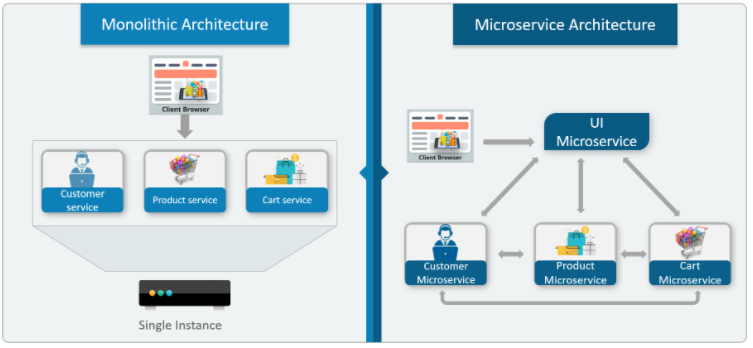
## **What Is Microservices?**

**Microservices** is an architectural style that structures an application as a collection of small autonomous services, modeled around a **business domain.**

**Figure 2**: Microservices Representation

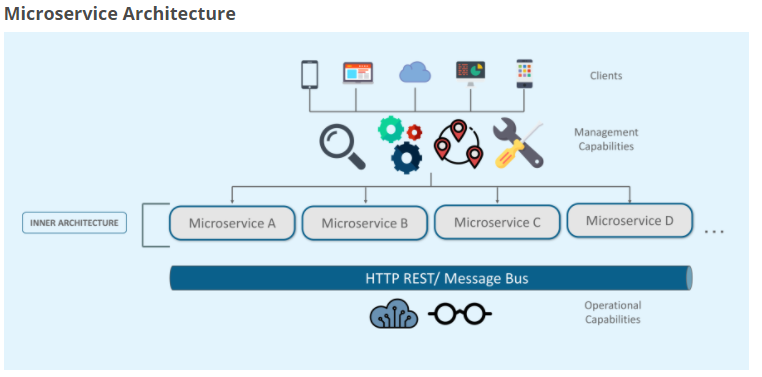
 each service is **self-contained** and implements a **single business capability.** Take a look at <https://www.gartner.com/en/research/methodologies/gartner-hype-cycle>.

## **Differences Between Traditional Architecture and Microservices**

Consider an E-commerce application as a use-case to understand the difference between both of them.

**Figure 3:** Differences Between Monolithic Architecture and Microservices

all the features initially were under a single instance sharing a single database. But then, with microservices, each feature was allotted a different microservice, handling their own data, and performing different functionalities.

**Figure 4:**  Architecture