Application architecture patterns

Vikash Verma

Agenda

Microservices vs monolithic vs SOA.

Illustrate use case scenarios for each.

Microservices vs monolithic vs SOA

SOA

Application components provide services to other components via a communications protocol over a network.

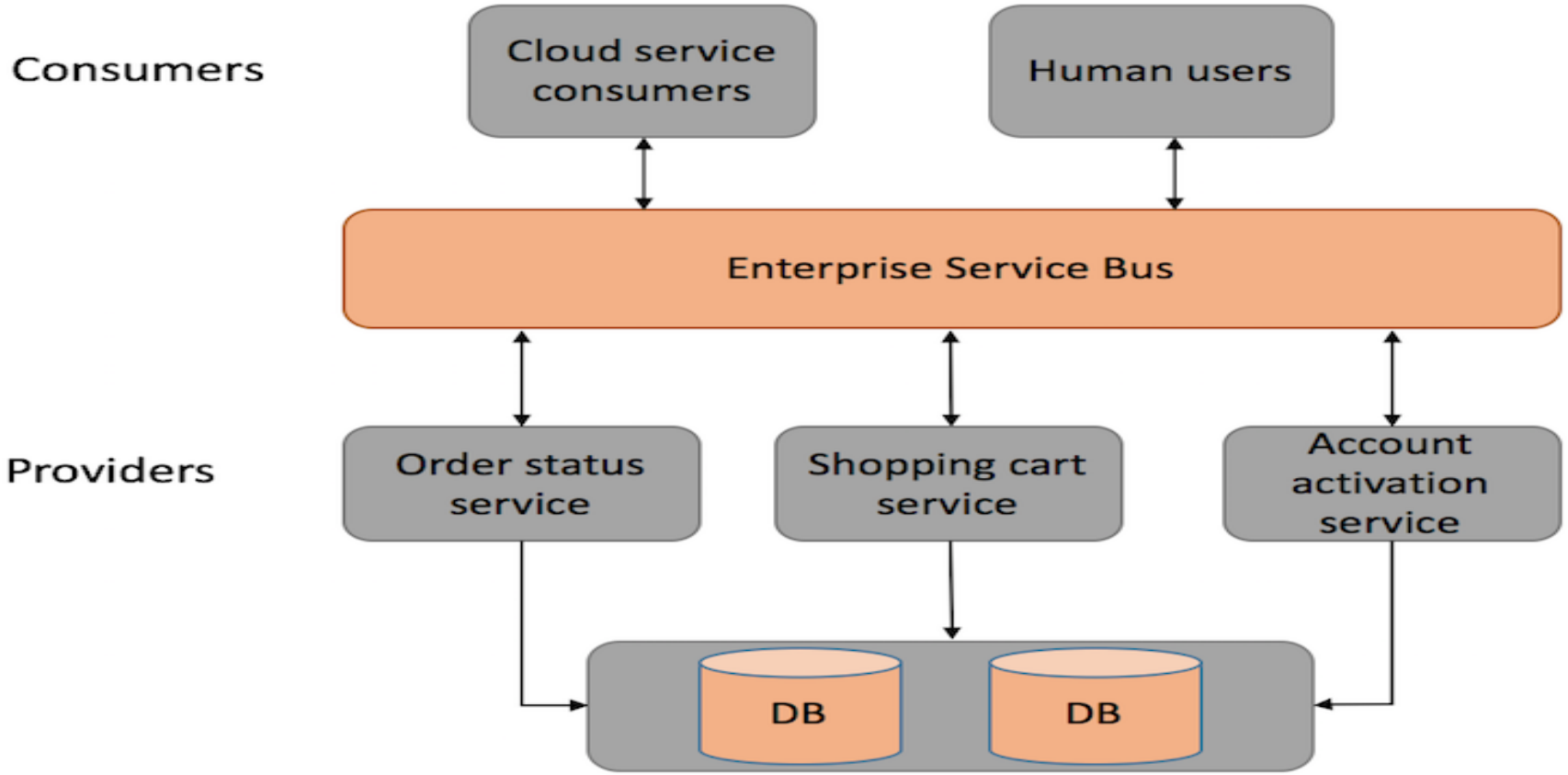
The communication can involve either simple data passing or it could involve two or more services coordinating connecting services to each other.

Services (such as RESTful Web services) carry out some small functions, such as validating an order, activating account, or providing shopping cart services.

There are 2 main roles in SOA, a service provider and a service consumer. A software agent may play both roles.

The Consumer Layer is the point where consumers (human users, other services or third parties) interact with the SOA and Provider Layer consists of all the services defined within the SOA.

Microservices vs monolithic vs SOA.



Microservices vs monolithic vs SOA.

Microservices Architecture

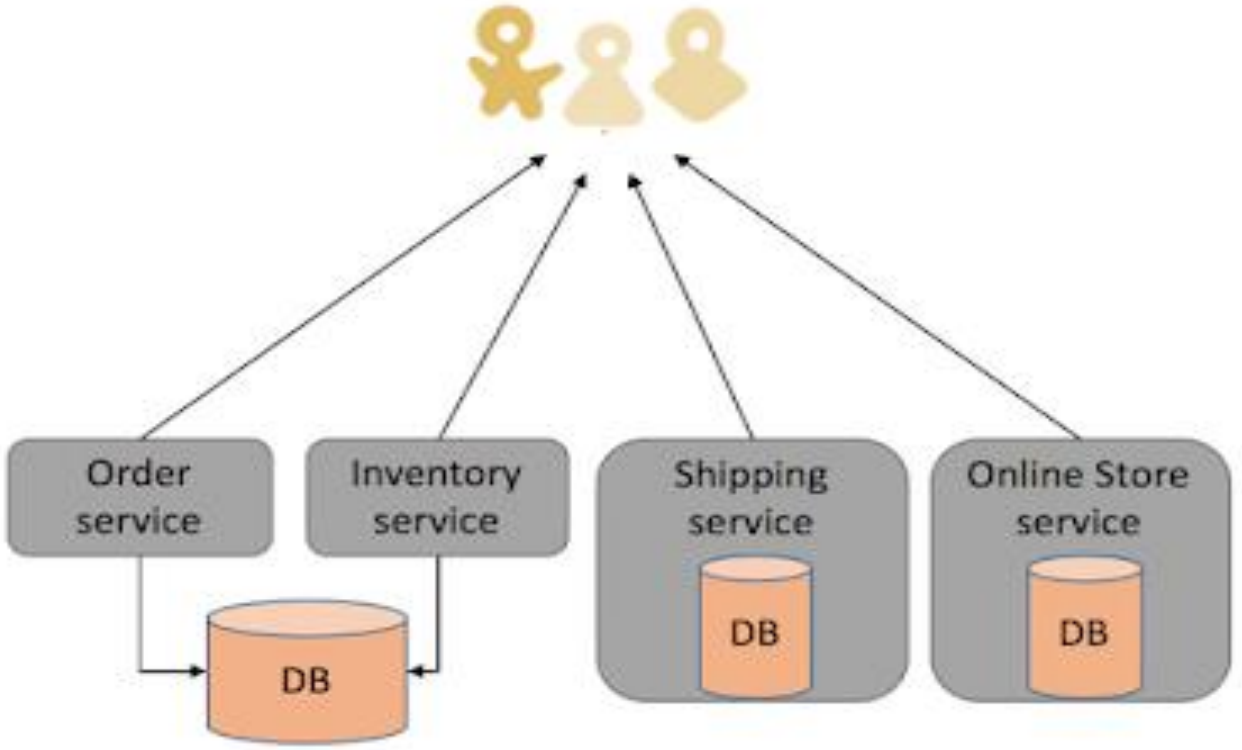
Microservices is a software architecture pattern in which complex applications are composed of small, independent processes communicating with each other using language-agnostic APIs.

Microservices must be a real need in the system architecture as it could be designed wrongly.

It means a service should be independently deployable, or be able to shut-down a service when is not required in the system and that should not have any impact on other services.

Microservices vs monolithic vs SOA.

Microservices Architecture



Pros of SOA

Reusability of services

Better maintainability

Higher reliability

Parallel development

Cons Of SOA

Complex management

High investment costs

Extra overload

Thank You