Advantage of Microservices

1. Faster release cycle
2. Easy to test
3. Follows single responsibility principle.
4. Upgrading each service individualy rather than the entire application

Disadvantage :

1. There is higher change of failure during communication between services.
2. Difficult manage large number of services.
3. Need to maintain network latency and load blalancing

<https://www.infoq.com/articles/microservices-design-ideals/>

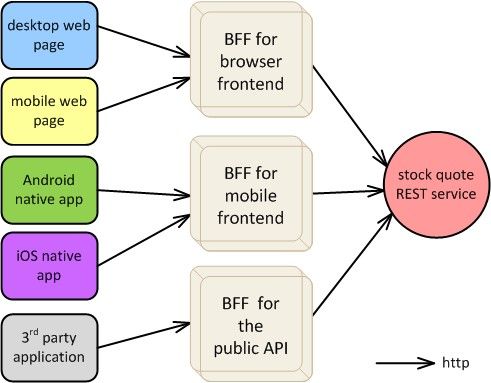
https://microservices.io/patterns/microservices.html

Scalability is the property of a system to handle a growing amount of work by adding resources to the system.

In general terms, throughput is the rate of production or the rate at which something is processed

* **I**nterface segregation
* **D**eployability (is on you)
* **E**vent-driven
* **A**vailability over consistency
* **L**oose coupling
* **S**ingle responsibility

Interface



Deployment

* + - Containerization and container orchestration
    - Service Mesh : Tool used for traffic monitoring ,policy enforcements,autntication,RBAC,routing , circuitbreaker,message transformation . Popular service mesh also includes Istio,Linkerd and consul connect.
    - API Gateway.- Amazon API Gateway
    - Severless Architecture – FaaS paradigm . AWS Lambda,Azure Functions, GCP functions.
    - Monitoring tools: New Relic ,Cloud Watch , Datadog, Prometheus, Grafana.
    - Logs consolidateion tolls : ELK stack , Splunk
    - Tracing tools : AWS – X-ray,Zipkin and Jaeger.
    - Deveops – CI / CD pipelining
    - Iaas – build and deploy cycle – reduces human intraction.
    - Externalizing the configuratrion.

Event – Driven : (Scalability and Throughput)

* + REST service
  + PRC calls with gRPC or GaphQL
  + Asynchronous call that goes through queue in the message broker.(Apache Kafka, RabbitMQ, Amazon SNS)
* SAGA design patterns – correction events mechanism of undergoing data changes
* Keep the end -user informed on the progress and mishaps

Avaialbility :

Service Data Replication Pattern :

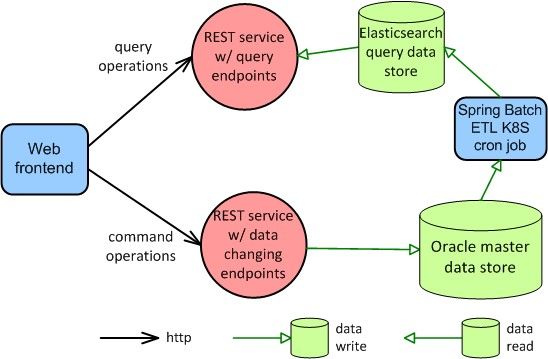
* + Used when miroservice want to use data from other application.
  + Pub-sub messaging instead of API call.(data -synchronisation mechanisam)

CQRS : Command Query Responsibility segregation

* + Improved performance and autonamy

Event Sourcin g - Prvide query view of data instead of strogin the state of object in data base .

Spring batch kubernetes cron job executed periodically to update the elastic search base store based on thechanges happended in the oracle DB



Loosely coupling :

API with Backend for Frontend

Database per Microservice pattern

Single deisgn responsibility :

Having multiple responsibility in one class naturally leads to tight coupling becomes fragile

* + Industries to follow Domain-driven design.
  + **Microservice can hava scope of DDD bounded context, configuration and dependencies.**

DDD is not a technology, rather it introduces terms, practices and principles to support decision making in complex business domains. If the complexity of the domain is not addressed early and properly within the application design, the best technology won’t help to create a system which is maintainable, extendable, etc.

Important terms in DDD

Domain Logic – business logic

Domain Model – includes ideas , knowledge , data a, metrics and goals.

sub domain

Design pattern – reuse code

bounded context -contains the complexity of the application.

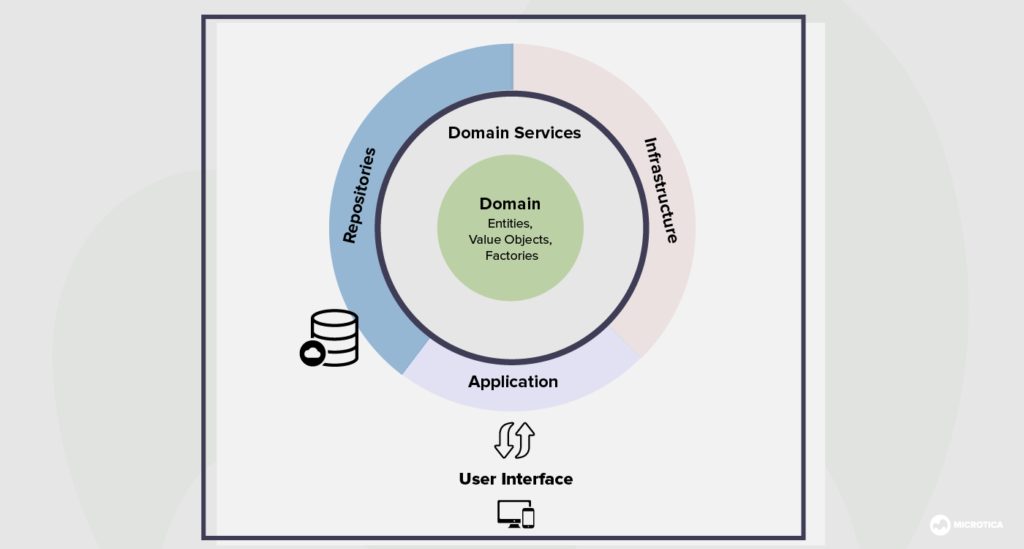
Ubiquition Language.

Entities 🡪 data + behaviour

Value Objects and aggregates

Domain Service

Repository – collection of business entities that simplifies data infrastructure.



Advantage :

1. Simple communication
2. More flexibility
3. Domain id more important than the UI/UX

Disadvantage :

1. Deep domain knowledge is needed.
2. Contains repetitive practices
3. Not good with high – technical projects