

Microservice Application

Advantages:

* Technology independent
* Engineering focus
* Scaling
* Not Overloaded VM/Containers
* Quick CI/CD Test, Build
* Better understanding due to small chucks
* loosely coupled

Disadvantage:

* Inter-process communication
* Distributed transaction
* More resources

Functional Decomposition: Divide large application into smaller chunks based on their functionality.

**API Gateway:**

Client/UI invokes API Gateway in which we call multiple microservices depending upon the requirement. These calls may be sequential or parallel.

BFF (Backend for frontend): we can have channel(web/mobile) specific API Gateway.

Advantages:

* Authentication: this can be done before invoking any of the microservice so that each of microservice will not need to do authentication separately.
* SLL Termination: HTTPS call mainly required to call API Gateway and microservices can be invoked from API Gateway with any protocol like http, https, web socket, RPC (Remote Procedure Call)
* Load Balancing:
* Insulation: Insulate microservices to get exposed to directly to external world.
* Further decomposition/combination of microservice will not impact the client code.

Disadvantage:

* More latency
* More management

**Service Registry**

Place which provides network address of the microservice.

* Self-registry: Microservices register their address in service registry. To keep their addresses updated in service registry, microservices keeps on refreshing this data at a certain interval.
* Third Party Registry: Service registry asks for the address to microservice. To keep their addresses updated in service registry, service registry keeps on refreshing this data at a certain interval.

Service Discovery: API Gateway ask for the network address of a microservice to service registry. Service registry provides all the network address of a asked service and then API Gateway does the load balancing on the network it received.

**Inter Communication**

Synchronous Call:

Advantage:

* Real Time
* Easy

Disadvantage:

* Service unavailability
* High Response time

Async Call:

Queue can be used to make async call. If API Gateway invokes MS1 and MS1 has dependency on MS2 then MS1 will push the message to queue and MS2 will subscribe the same queue. Once MS1 is done with the push, it can respond to client.

Advantage:

* Faster APIs
* Decoupled services
* Works when services are down (queue takes care of messages)

Disadvantage:

* Complex Design
* Process latency (overloaded queue)
* Monitoring Cost

**Circuit Breaking**

* Cached Response
* Fallback Service
* Configure healing time

**Service Mesh**

Service mesh will be there in all the microservices which works as proxy whenever a MS tries to invoke other service. It does follow activities:

* Load balancing
* Service discovery
* Metrics
* Circuit breaking
* Timeout

