Organization of dis sis = distinction between logical organization and phys realization

Logic organ: collection of sw comp that constitute the sys => sw architecture

Phys real: instaniate and place sw comp on real machine -> sys arch

Arch styles:

Layered arch

Object based arch

Data centered arch

Event based arch

Microservice

How to distinguish diff arch style? 3 elements

What are the components?

How the comps are connected(connector) (mechanism to exchange data between comp, procedure call)

What type is the exchange data between the component?

Layered arch

Divide sys into diff layer

Each layer has its own task

To give the trans to each layer => less affection when perform update (layer1 update no affect layer2)

Req flow and response flow: layer n send res to layer n+1, all the layer are trans to each other

Applied in networking community (OSI model), or general dis sys arch with middleware

Widely used today is 3 layer arch

User interface layer

Processing layer

Data layer

Object based arch

Comp: obj (oop)

Connector: procedure call (method call)

Obj can call a method that located in other obj (obj client and obj server)

Obj server is the obj that implement the calling method

Eg: cobra system (Common Object Request Broker Architecture) , client can call a method that in the server, all the module play the role of middle ware of this arch, that will transform the calling method in the server, after that It will send back the result to the client

Remote method invocation (call a method from an obj from diff process) and local method invocation (same process)

Middle ware: obj req broker

Event based arch

Commu thru the propagation of events (optionally carry data)

Publish subcribe sys: consist of subcriber and publisher, sub to the system which kind of event it can supply, publisher will publish the event to the sys, publisher doesn’t care/know about subcriber, after that the sys will check the sub event list to know which what subcirber can support this event, use multicast method to send the event to the subcriber

Event is not data type to send and receiver, just an infomation

Process are lossely coupled: all comps are indiependent, no constraint between publisher and subcriber

This arch is mostly applied in iot sys

Data centered arch

Commu thru a common repository (sender must send data to a share data space, then the receiver comp will pull the data from the data space)

Eg: email system, sender write email and send, email send to the email server, receiver pull all email from the mail server (pop3, imap)

All 4 above are monolithic arch, that combine diff module in the same layer

All ict sys nowadays move to cloud style, monolithic is not appropriate anymore

Microservices

Divide the whole system into small services, each service is developed by a small team, there are no constrain between services,

New arch style, very appropriate for cloud sys developing

Benefit: simple to deplo y

Eg: