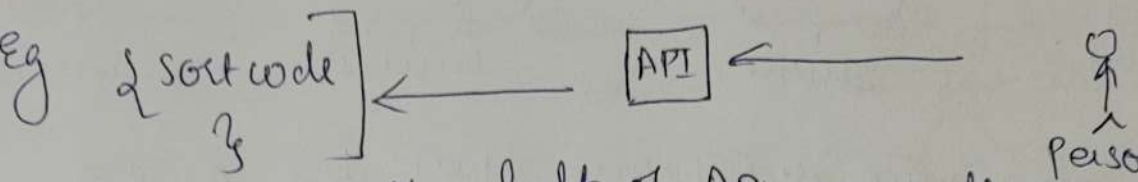
VID14 API

API - is a documented way in which external consumers can understand how they can interact with our code



→ Person A with the help of API can interact with our code

def1 → It is set of instructions that allows 2 softwares to communicate

def2 → Mechanism that allows 2 softwares to communicate with each other using set of instructions & protocols.

Eg Function → API for getAdmins (String grpID)
{
}
error1 grp doesnot exists
error2 grp is deleted

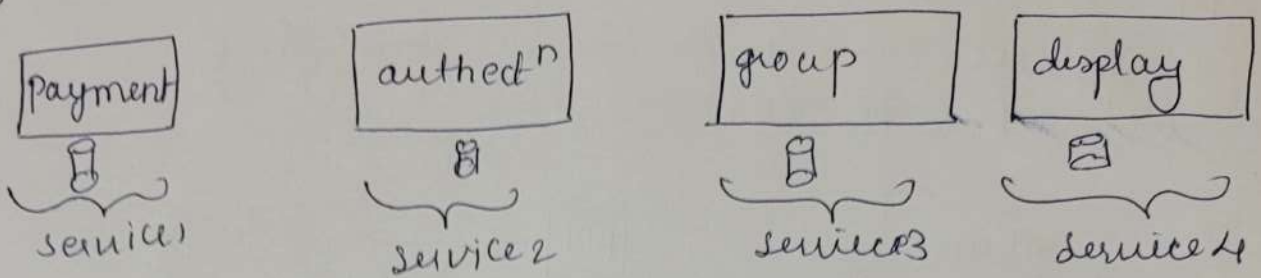
Response → list<Admins> ---→ response.

Suppose we want to call list of admins of a watsapp grp ie we will call function / API that will interact with our watsapp software calls the func and will give the response
→ If things go well we will get list of admins.
there are 2 possibilities of error.

Its just a contract → ie we are just concerned of how to call fun/API, not concerned about the what the function does.

where the function or API be exposed ?

Eg we have a microservice - ie airbnb applⁿ / watsapp



In our microservice architeture there are no of services. so our get admin fun / API should be placed at service 3: group service.

```
request ← getAdmin (String groupID)
₹          {
           }
```

response1 ← List <Admin> admins (if everything is good)
response2 ← error group doesnot exists
response3 ← error grp ID ~~has~~ ~~to~~ is deleted

⚡ Things to keep in mind while genrating API / calling API

① → Naming is imp.

② → giving our API more info so we don't have to call other API → use when we want our API to be fast.
So addition of paramatus are useful

Eg → getAdmin (String grpid, String grpname)

③ → Response should ~~e~~ contains only defined results
Eg we call admin
in response we get admin.
grp created date } → not needed
No of peoples in grp } → not needed

→ Sometimes, in response of api call the developers include additional information unnecessarily which is confusing.

→ Defining correct errors.

Eg getAdmins (String grpID)

we are grpid as string so we will just define in parameter if user enters id as int then the error would be thrown and query would be ~~[illegible]~~ failed.

So this is not expected error.

invalid errors — our grpid string है par user int दे रहा है तो error ede do 404

valid error → if no grpid exist नहीं करती फिर error dena की " grpid doesnot exists"

→ What IS API

→ Eg & use case of API

→ Types of API

→ API Standards

API – APPL^n Programming Interface

APPL^n → when 2 appl^ns wants to interact with each other, it does through APIs

Programming → Because one piece of code can interact with other piece of code which can be running on same machine or different machine, it can be in same or another lang but one piece of code would interact with another piece of code programatically (we can send, read or fetch data)

Interface → abstraction – [Code1] → [code2] c1 will call c2 through API
c1 knows ~~what~~ that he needs to call c2., ~~but~~ he doesnot knows how c2 is doing

Interface → one code knows which code to call, & what C2 does

C1 → C2 C1 knows that he needs to call C2, but he doesn't know how the C2 works

Eg we hit post button on ins

post → API → postphotocode

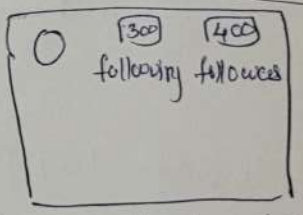
when user hits post button, the API will call post photo code that will post the photo

→ post button only knows that he needs to call post photo code Api, but he doesn't knows how the postphotocode API code works

API → we have an Interface with we interact with another code. We know only what the other code does. But doesn't know how the implementation of other code is happening behind.

Adv of API

① communication ② Abstraction



② Eg Today we have 200 followers
we click follower button → it will call get follower API and we get 200

→ now some insertion happens as no of followers inc

→ Tommorow next day, we dont have to do any operation in code, we just have to click followers button → that call getfollowers api and we will get result

③ → Platform agnostic

If one code is written in java & other in python with help of API we can directly call other code

---

TYPES OF API

→ Private API → transaction API in gpay
→ Public API → google map API, weather API
→ Web API → apps running on cloud, Eg get followers, Superset of ① & ②
→ SDK / library API → lock, release, fork, join

FACTORS FOR BUILDING API - API FACTORS

① → API contracts → it is a contract or list that the api developer gets before developing any API
→ API contract is imp because it ensures that the ~~get developed~~ the Api that is going to be develop would & meet the expectations of both provider and consumer.
→ API contract has set of → rules → specifications
→ functionality → expectation for data exchange & commun
→ what all data should be shared.

② Documentation → Rules how API should be called

③ Data format → data format ie in ~~which~~ ~~can~~ what would be the data format of our API
Eg JSON, XMAL, XML file format

④ API security → there should be security to ensure that only valid or true users / code should access our API
As API is a interface that provides access to code
if no security then an illegal user can break our code or could misuse it

---

STANDARDS OF API

→ RPC
→ SOAP
→ REST