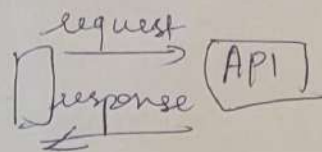
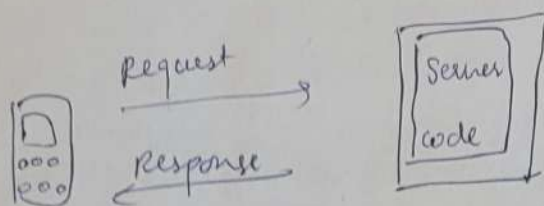


VID 2 - BASICS OF SYS DESIGN



Suppose we have computer stuff where we have written some code → i.e. normal fun, takes some inp & gives op.

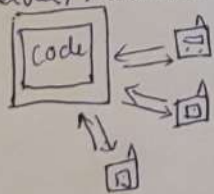
- Now people think it might be useful & they want to use that code
 - but we cannot give our computer to everyone.
 - So we will expose the code on the internet using API
- API → Applⁿ programming interface

Things require to setup this computer

- Server, computer, plugs.
- Now people are using our code and pay us for code.
- But in case server goes down, what will happen?
- AWS → use this, it is set of desktops/computers where we can run our code.

SCALABILITY → **Vertical** & **Horizontal**

server/machine



- Now suppose our code is in cloud.
- No. of users who are requesting to use this code increases and our normal machine not able to handle those requests.

Solⁿ 1 → Vertical Scaling → ~~increase~~ increase the size of machine so it can handle more request

Solⁿ 2 → Horizontal Scaling → increase the no. of machines so requests would be distributed among the machines

HORIZONTAL SCALING

M1 M2 M3 M4

① Load Balancer required to route the requests to diff servers/machines

② RESILIENT

if one server breaks, the req can be redirected to other machine/serv.

③ all communication happens within network \rightarrow so Inp/Op is slow
req/response is slow

④ Remote procedure call

④ Data inconsistency
Suppose there is a case where we want to send data from M4 to M3 and from M3 to M2, then in that case we might need to lock the PCs

⑤ Scales well

we can inc the no of machines to any limit we want

What is used in real world

Ans \rightarrow Both

from Horizon scaling \rightarrow use pt ② & ⑤

from Vertical scaling \rightarrow use pt ③ & ④

Ideally suggested to use Horizontal scaling where we use big machines \rightarrow ie combination of both.

VERTICAL SCALING

M' Big machine

① x/A.

② SINGLE pt of Failure
if the machine breaks, the code cannot be retrieved.

⑤ Inter process communication
all commⁿ happens directly with server.

④ Data consistency

③ Hardware limit

\rightarrow at one pt we can big the max big machine and req is more, we cannot make it big