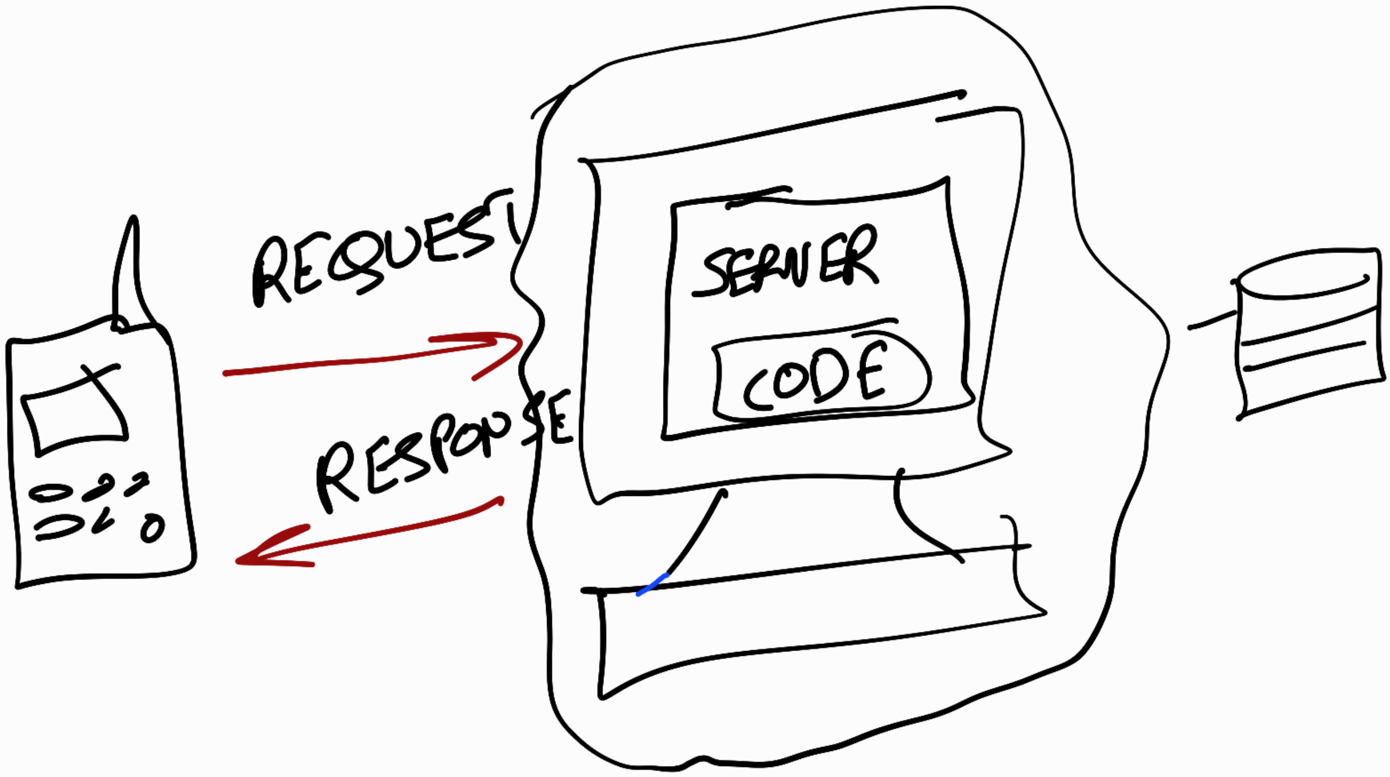
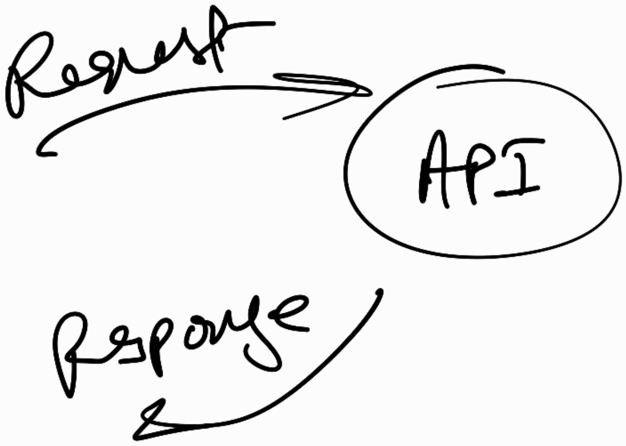
①. Horizontal vs vertical scaling:



1) Expose code as internet protocol i.e. API

<- if written code should be used by clients

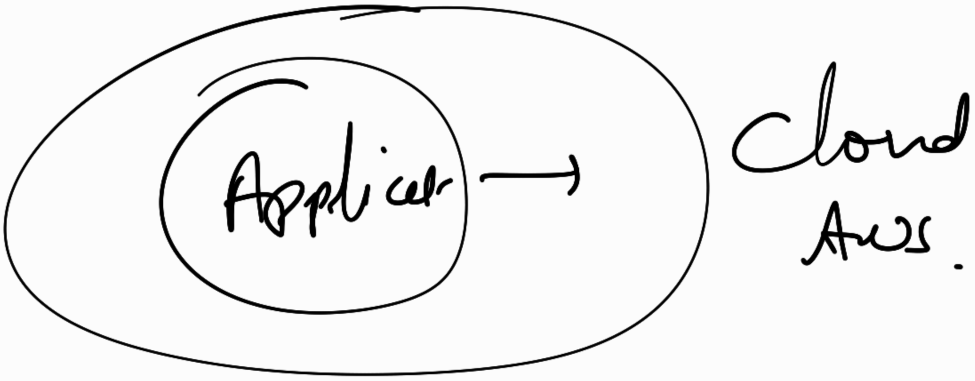


✓ each Req → Response.

✓ DB Required for Tx

✓ cloud → set of computers that somebody provides you.

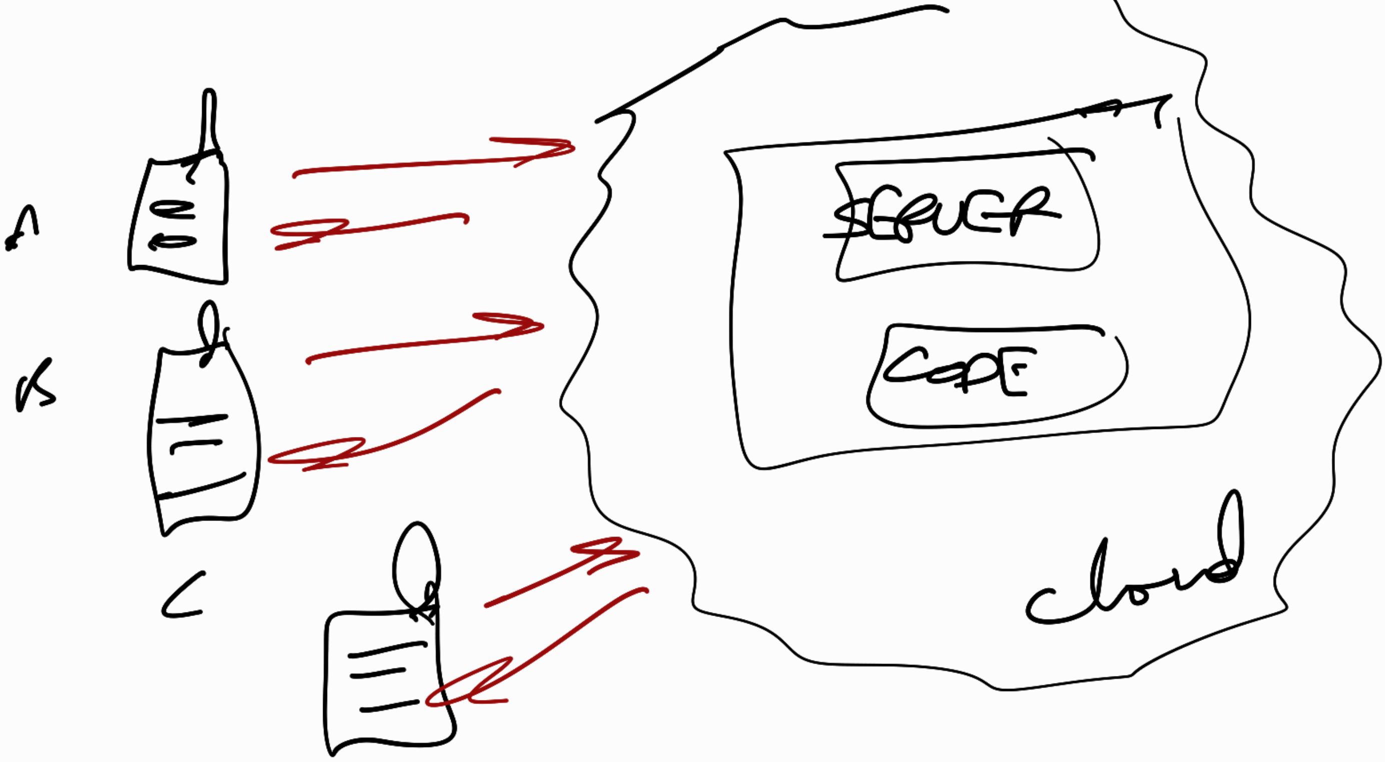
ex: AWS.



✓ set of computers
Remote login

ex: Now Server hosted on cloud.

suppose if client Req are
more



if code in machine not able of handle all Req's → A,B,C.

Sol^n 1 Buy Bigger machine

sol 2 Buy more machines

The Ability to handle more Requests by sol 1 or sol 2 is called scalability.

we ca handle more Rq's if we throw more money at problem.

① BUY BIGGER MACHINE — VERTICAL SCALING.

② BUY MOR MACHING — HORIZONTAL Scaling.

Horizontal	Vertical
1 2 3 4	HUGE BOX
① Load Balancing Required	① N/A
② ✓ RESILIENT if one fail → Req to another, retr	② single point failure
③ N/w calls slow [b/w conn's] → RPC (Remote procedure calls)	✓ ③ Inter process Comm. (fast)

(4) DATA INConsistency	(4) ✓ Data Consistency
(5) Scales well ✓ if users increase.	(5) Hardware limit Can't make system bigger/bigger if users increase?
2,5 - Good points	3,4 - Good points

Money wise ✓ Vertical scaling.

is it scalable ?
is it Resilient ?
is it Consistent ?

These are the major Considerations while designing a system we design a System which is going to meet the Reqirements.