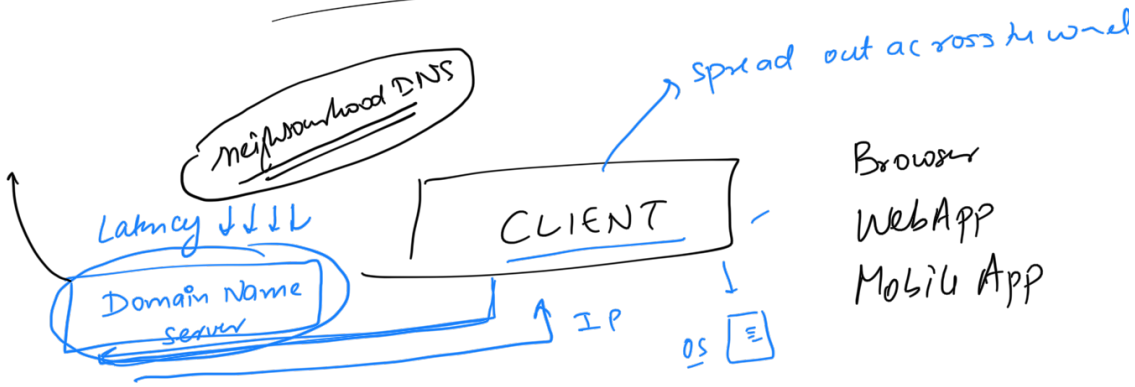
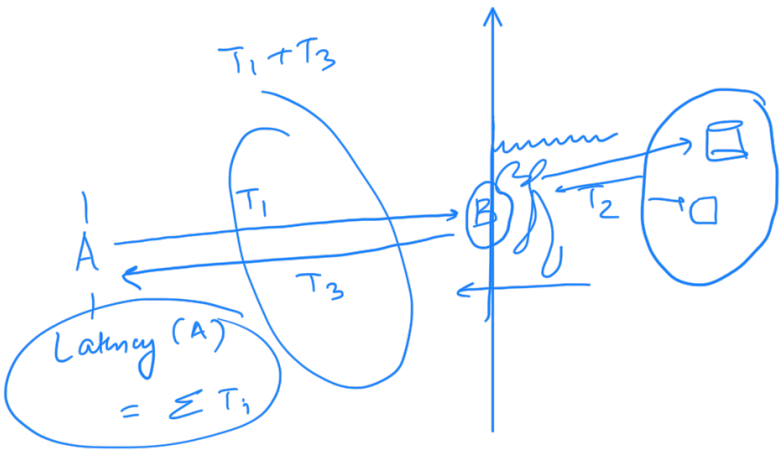
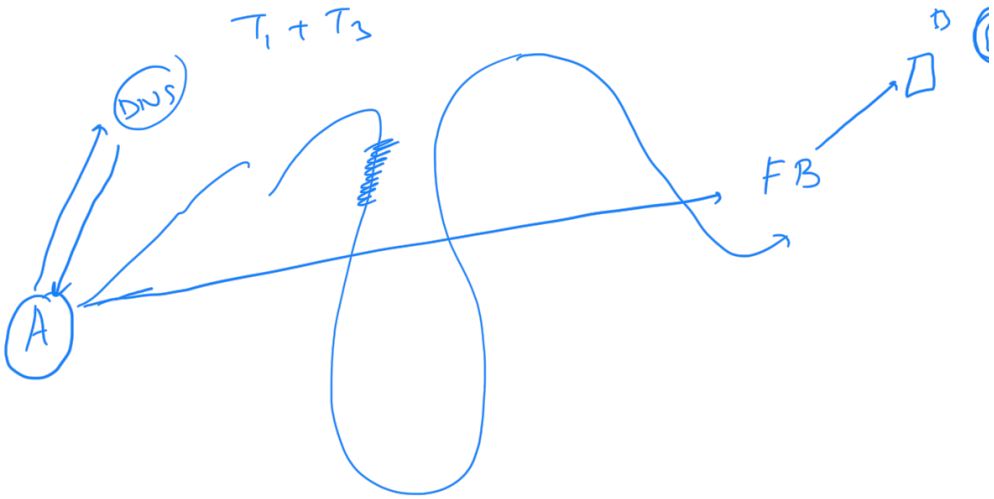
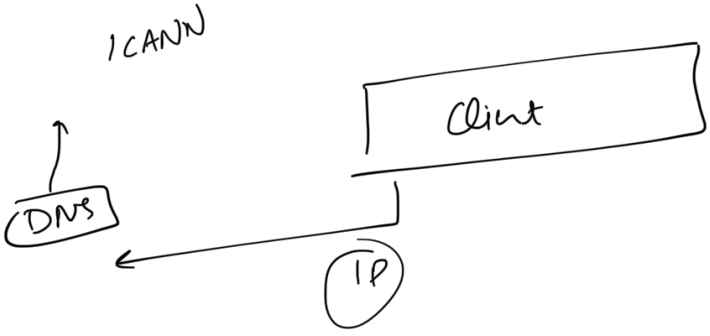
High Level Design

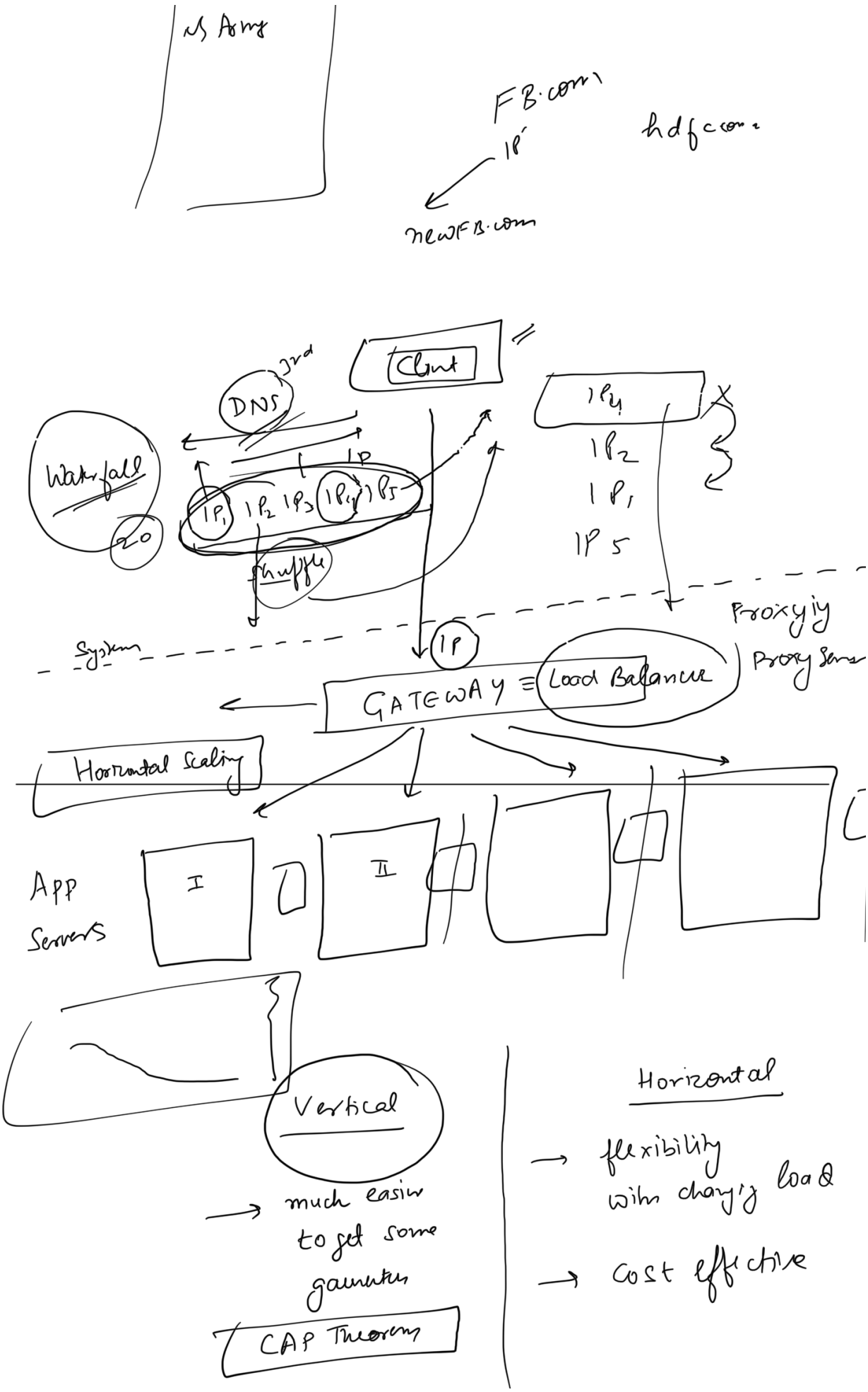


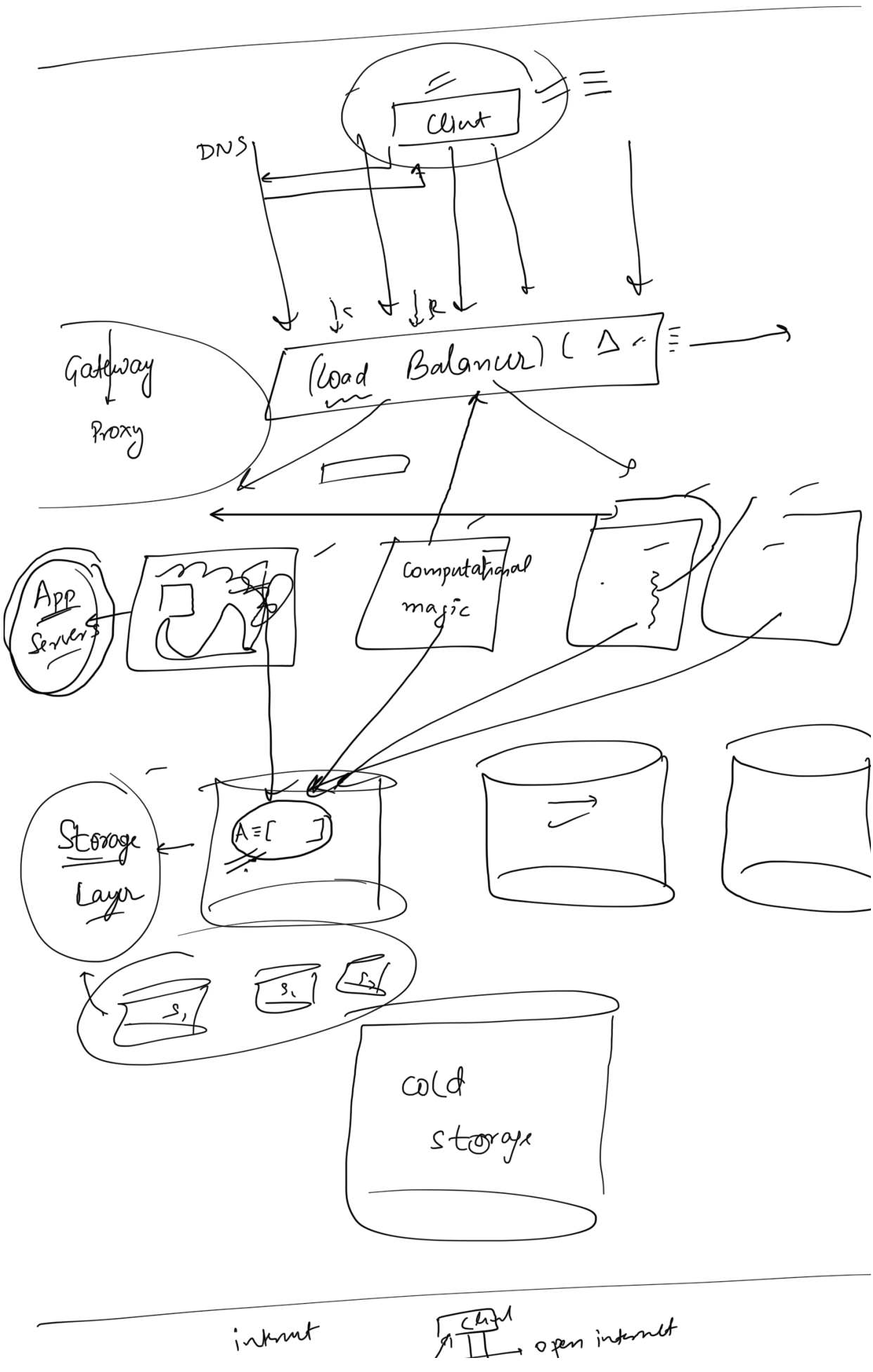


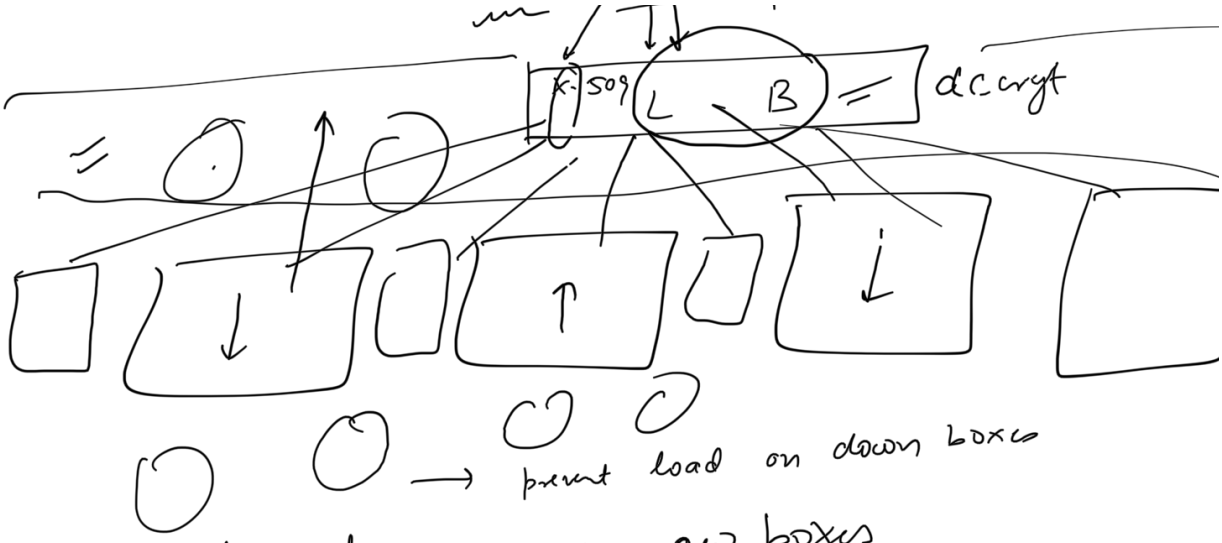




ISPs
Google
FB


MS Army
FB.com
IP
newFB.com
hdfc.com
Client
DNS
3rd
Waterfall
20
IP1 IP2 IP3 IP4 IP5
Shuffle
IP4
IP2
IP1
IP5
System
IP
Proxy
Proxy Server
GATEWAY = Load Balancer
Horizontal Scaling
App Servers
I
II
Vertical
→ much easier to get some guarantee
CAP Theorem
Horizontal
→ flexibility with changing load
→ cost effective


Client
DNS
Gateway
Proxy
Load Balancer
App Servers
Computational magic
Storage Layer
A=[ ]
S1
S1
cold storage
intranet
client
open internet



→ prevent load on down boxes

intranet → rgistr new boxes

→ balance out

— X.509 —

Layer 4 ≡ Transport Layer

addition security

~~security~~ Source IP

Source Port

Dest IP Dest Port

fast

simple

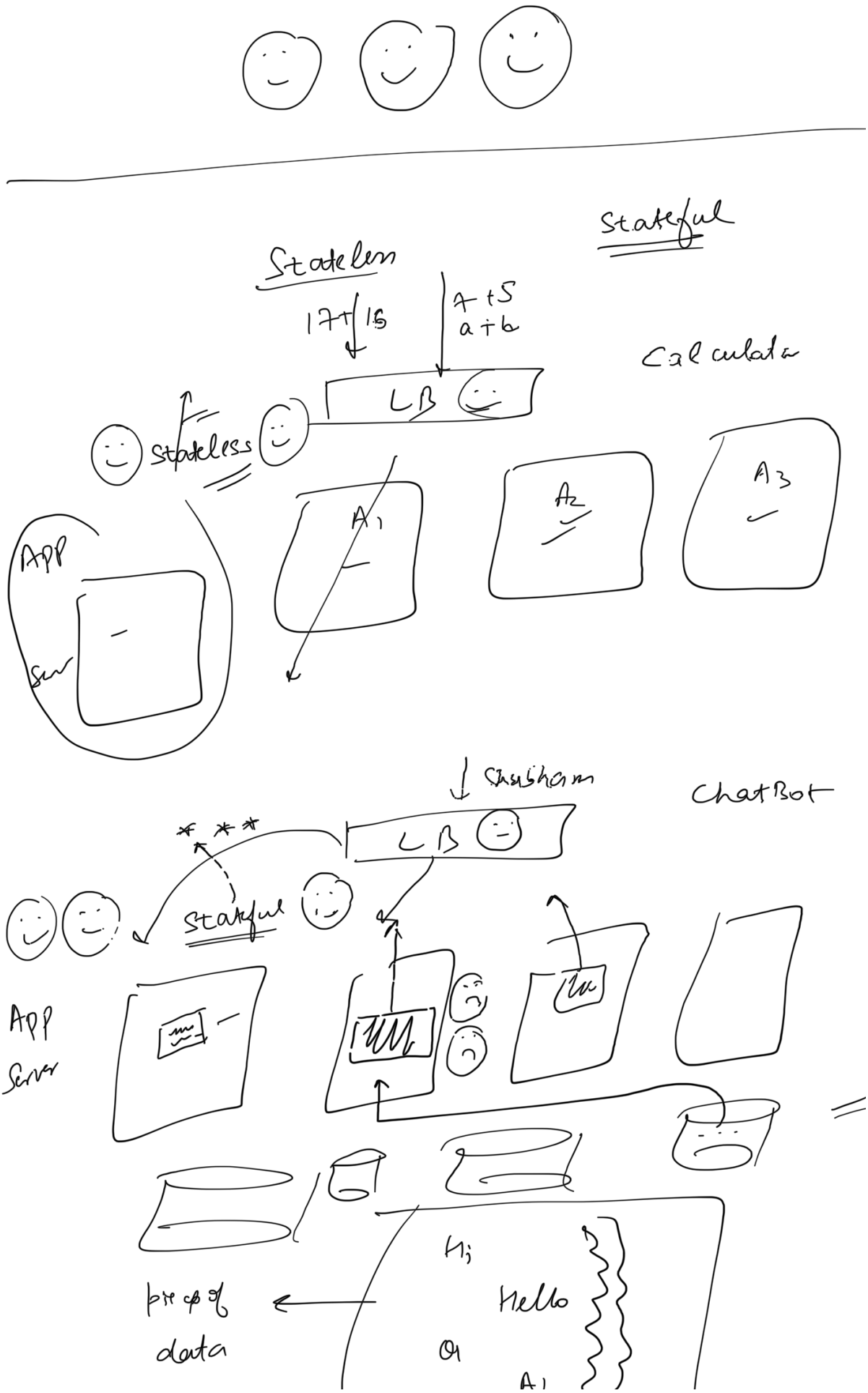
Layer 7 LB ≡ Application

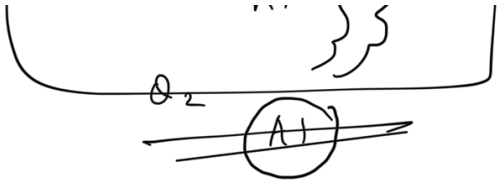
user id ≡ 707

uid = 107

much bttr flexibity

Slowest

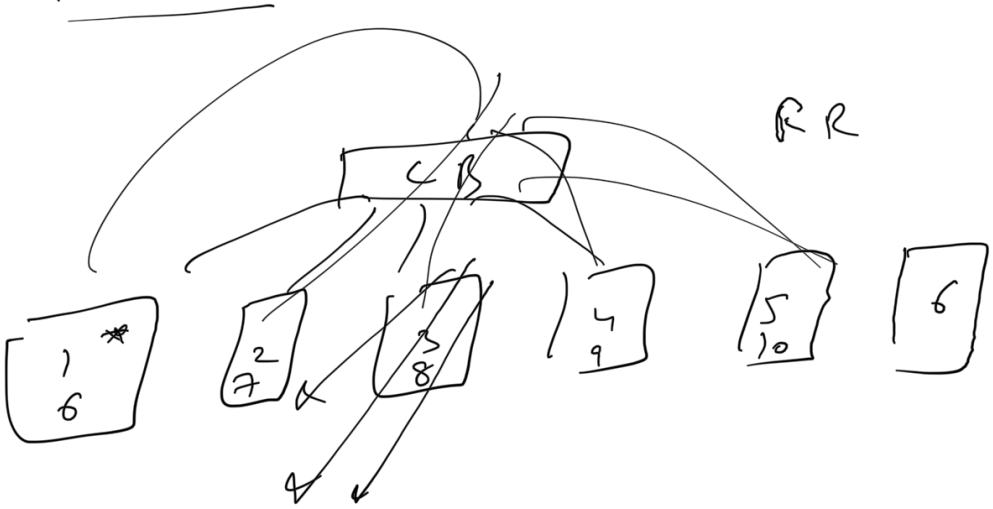

Stateful
Stateless
17+15
4+5
a+b
Calculate
LB
stateless
App
Server
A1
A2
A3
Chatbot
LB
stateful
App
Server
Hi;
Hello
data
A1

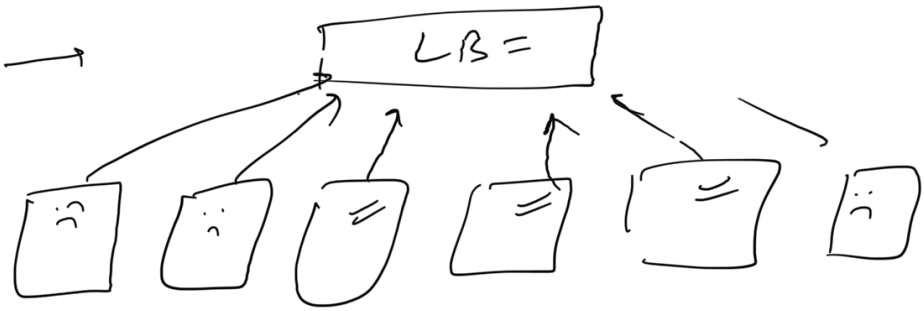


Load Balancing

Stateless

→ Round Robin

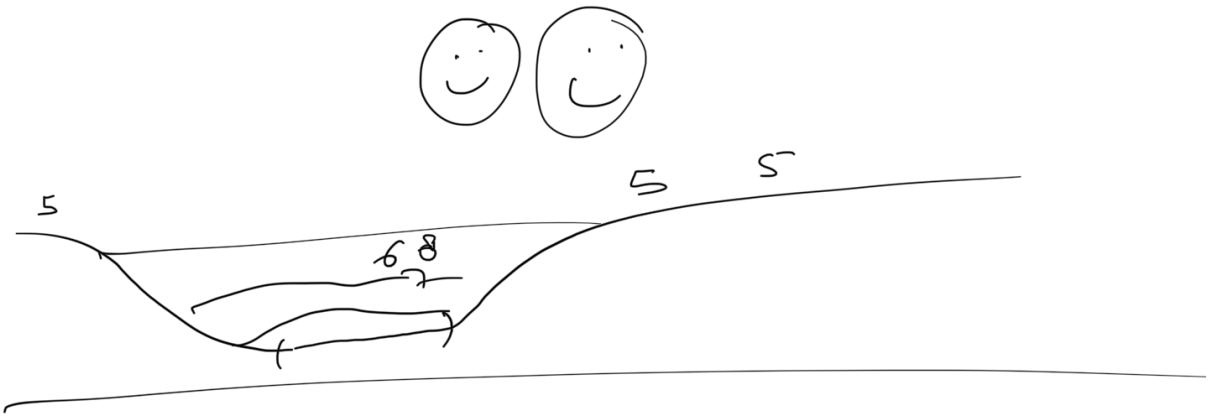




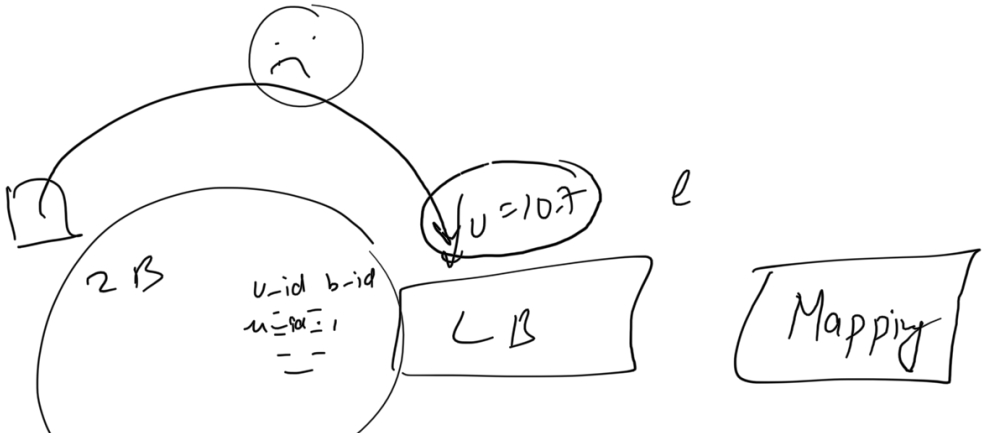
WRR

→ (Least Connection first)

→ WRR ARR



Load Balancing in Statful

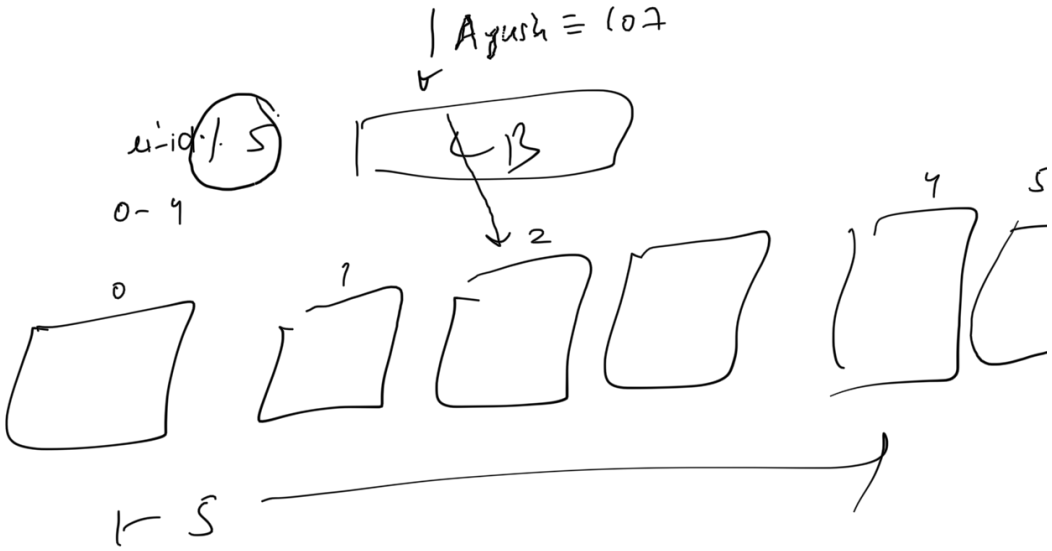


Sol (1) Maintain Map BAD

Sol (2) Hash

10 boxes ≡ 10 App Server

uid % 10 = 0-9



→ aaurat of stickiness

→ LB ☺

Do consider
On server inc/dec
you will have %6
to move stats of all boxes
internally