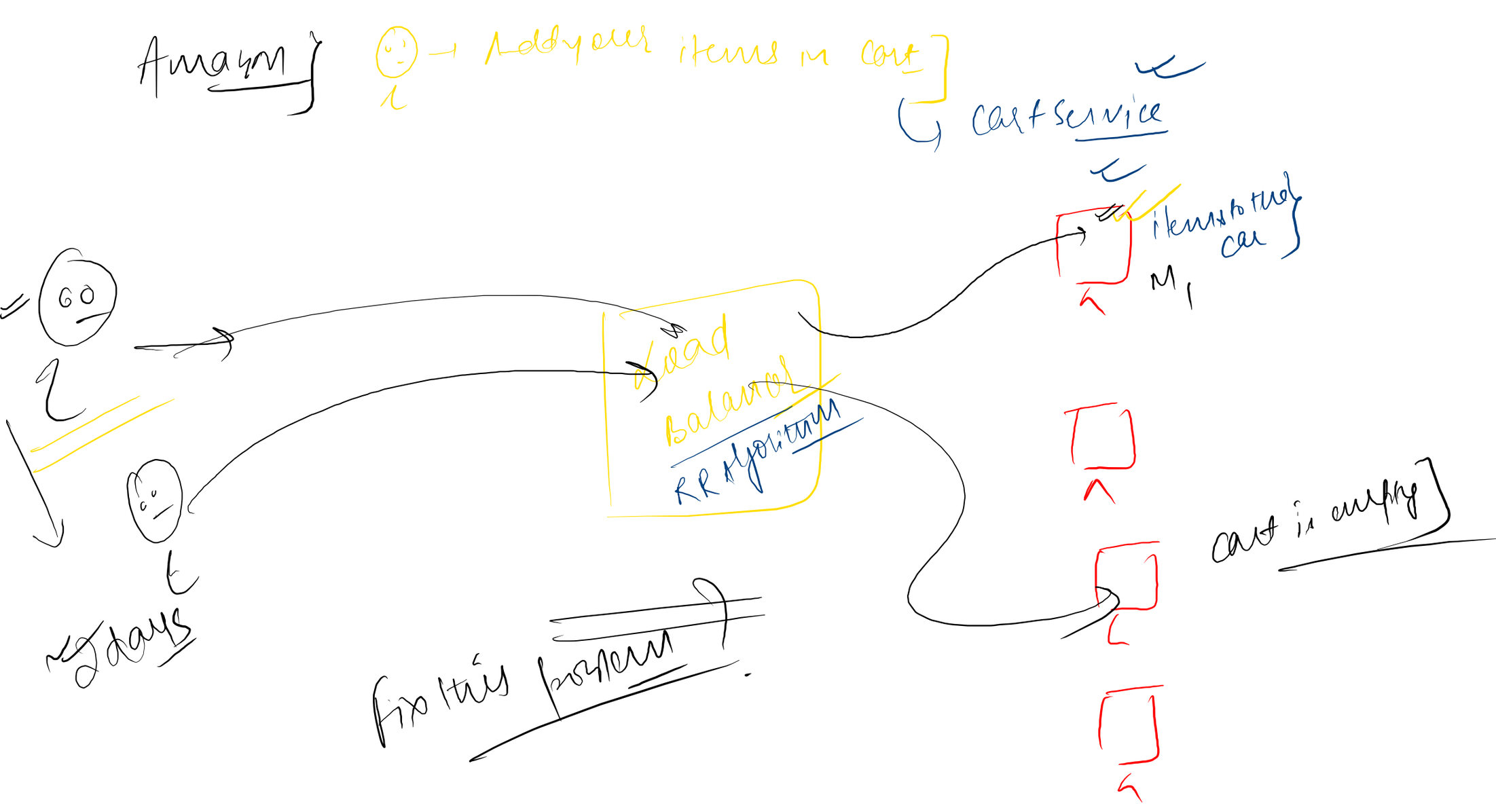
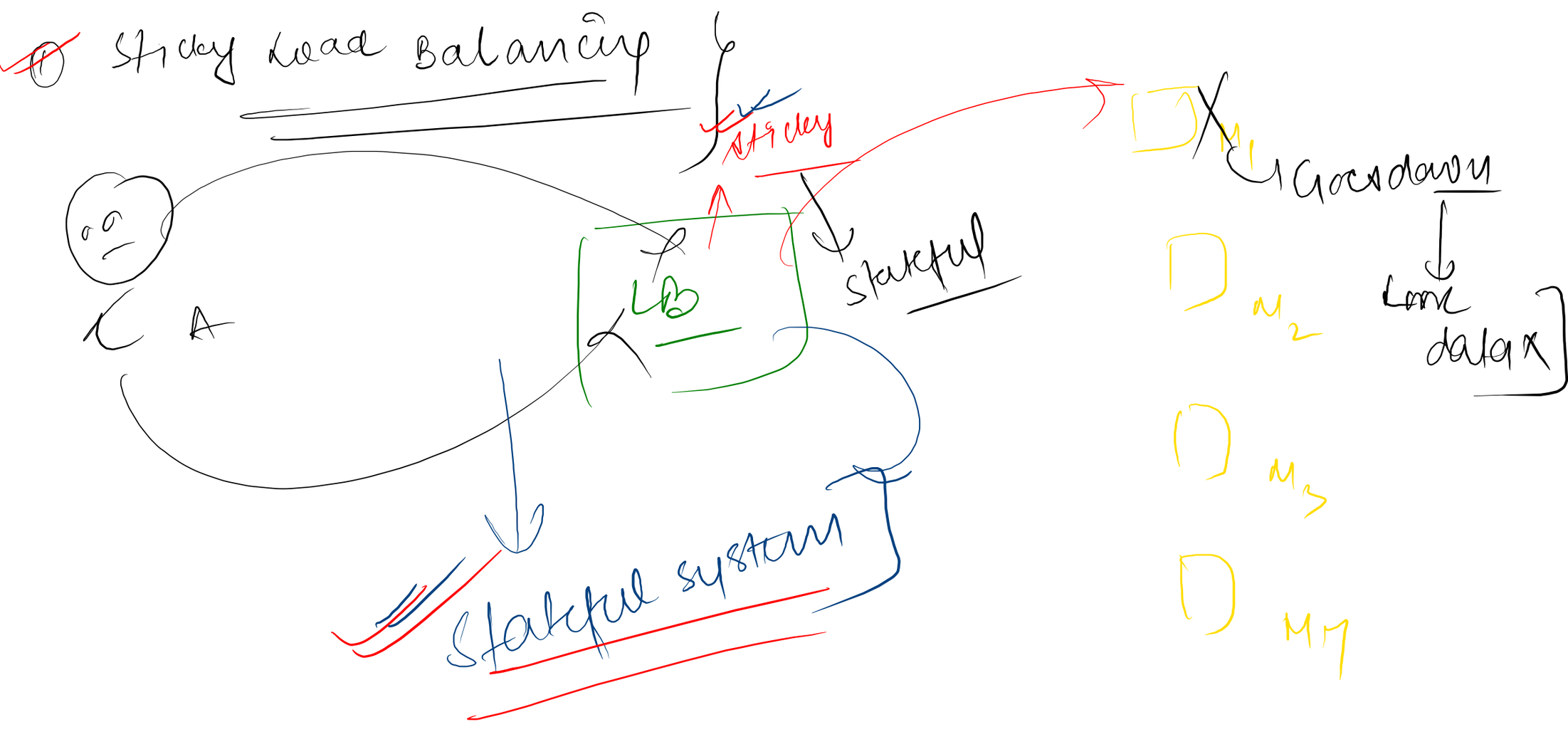
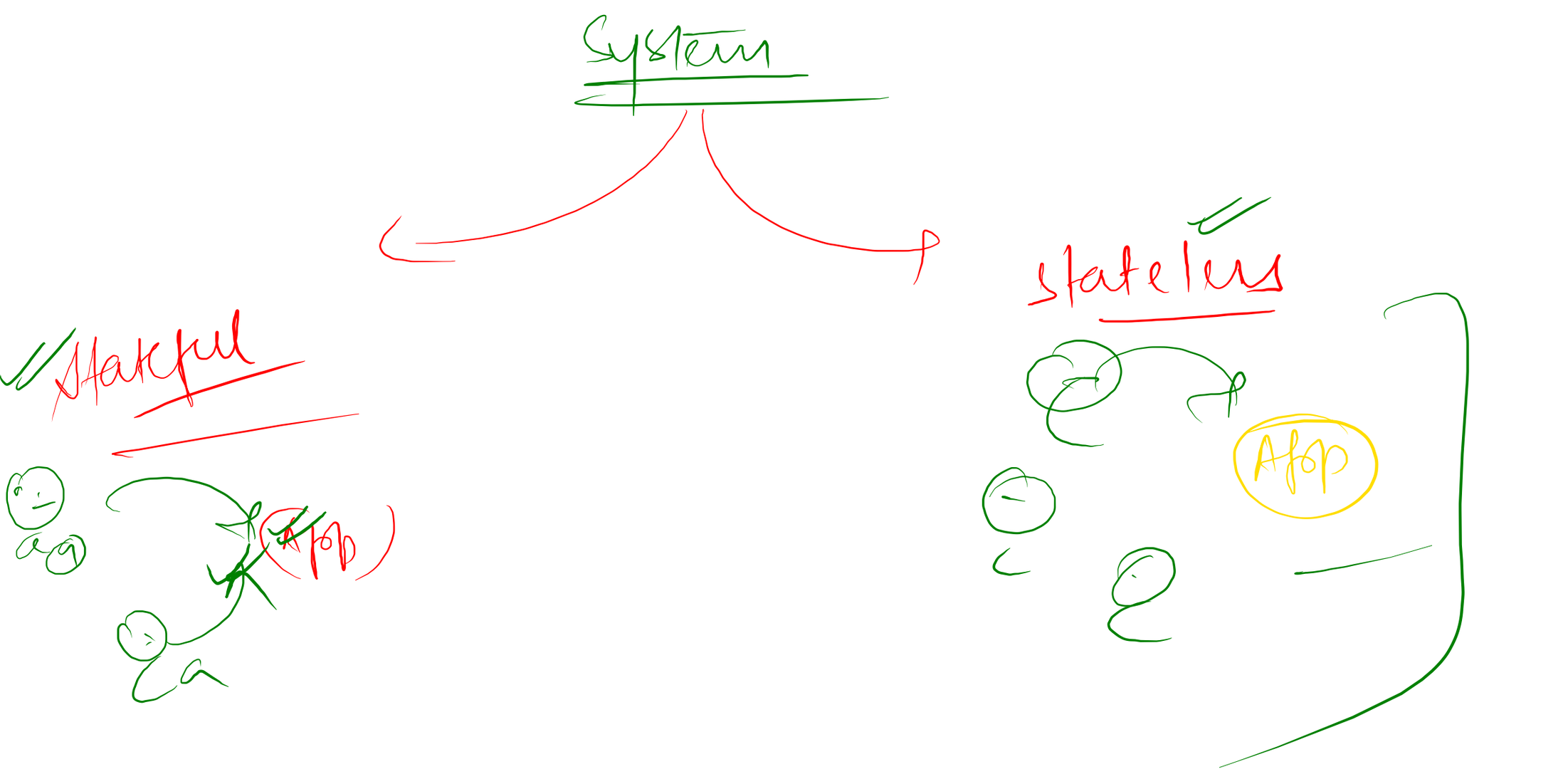

Amazon
Add your items in Cart
CartService
items to that cart
M1
Load balancer
RR Algorithm
cart is empty
My days
fix this problem


Sticky Load Balancing
Sticky
Stateful
LB
A
Stateful system
M1
Cookie
Local data
M2
M3
M4


System
Stateful
App
Stateless
App

Stateful → Preserve the information about the user/client

(Challenge)

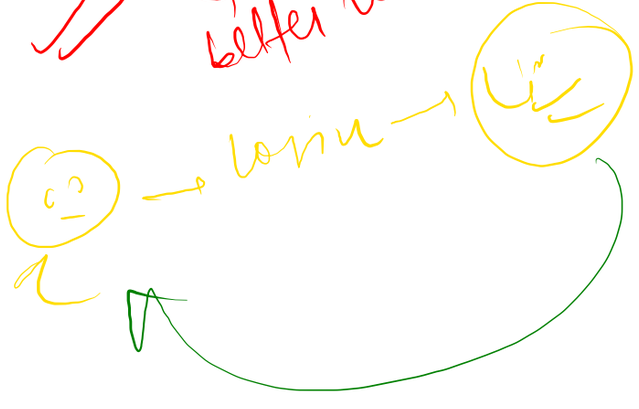
Advantage

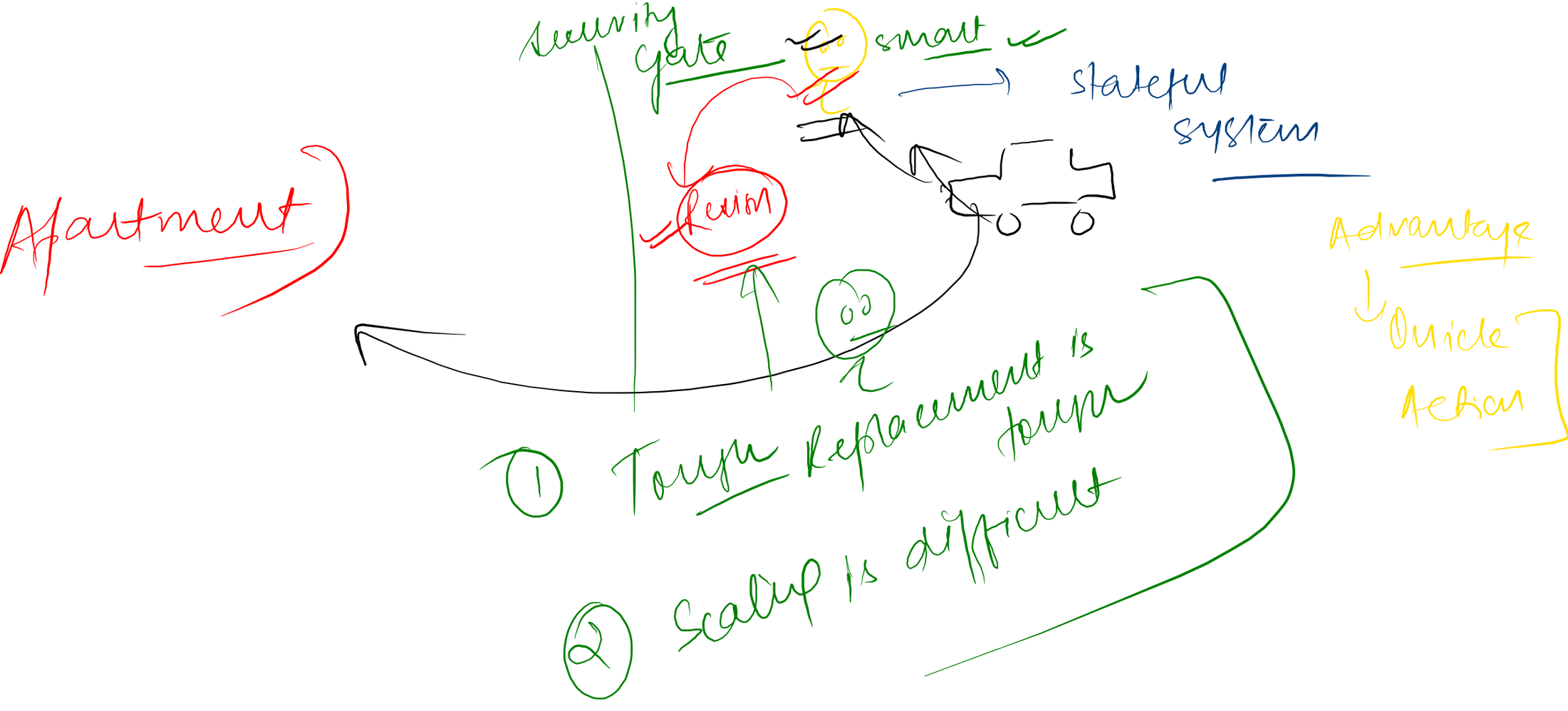
① customized flow → better user experience

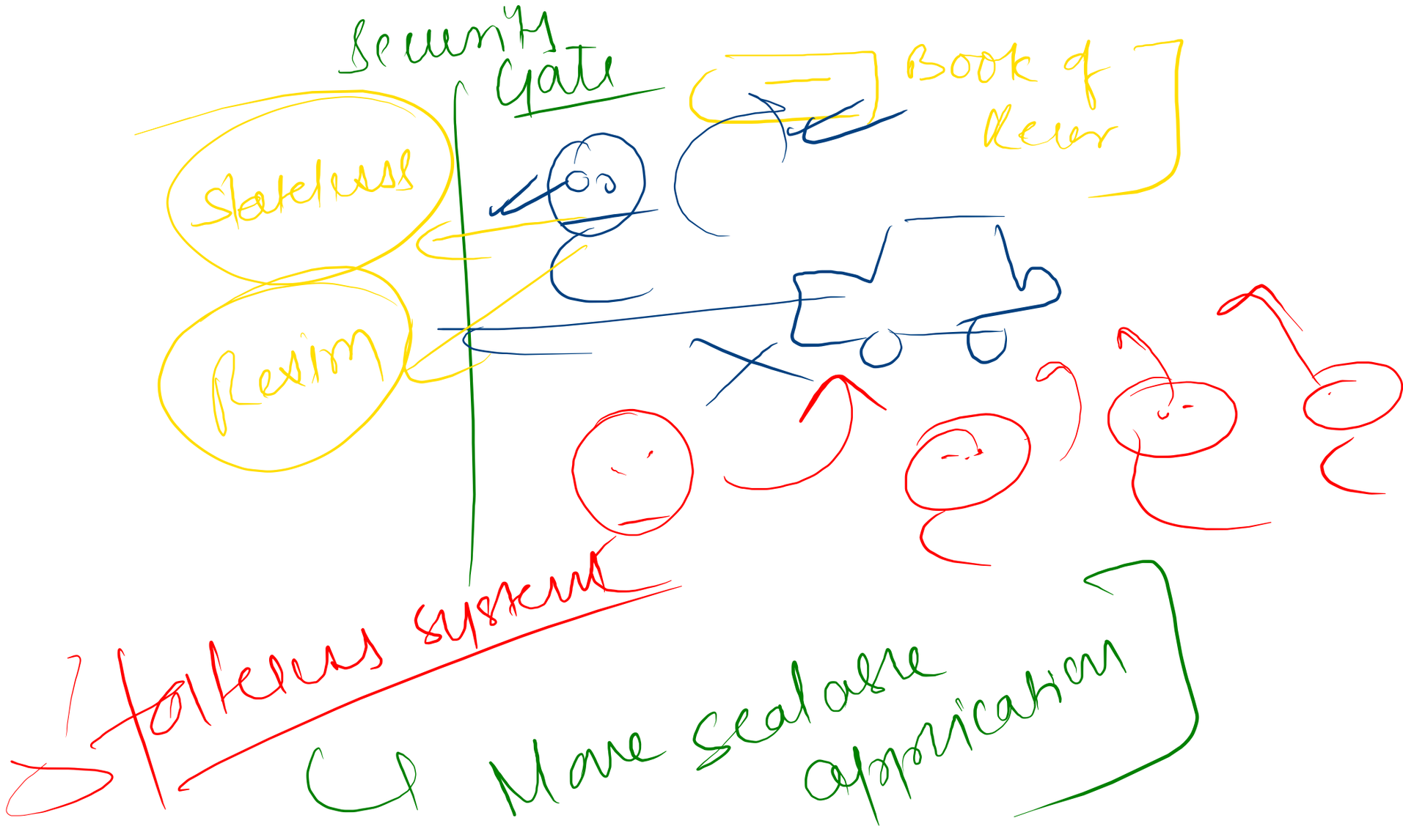
fascinating way

② failure recovery ↑ expensive operation

③ scaling becomes tough

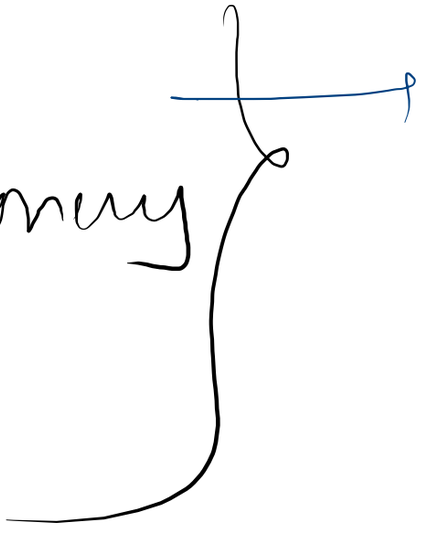



Security gate
smart
stateful system
Apartment
Region
Advantage
Quick Action
1 Token Replacement is tough
2 Scaling is difficult

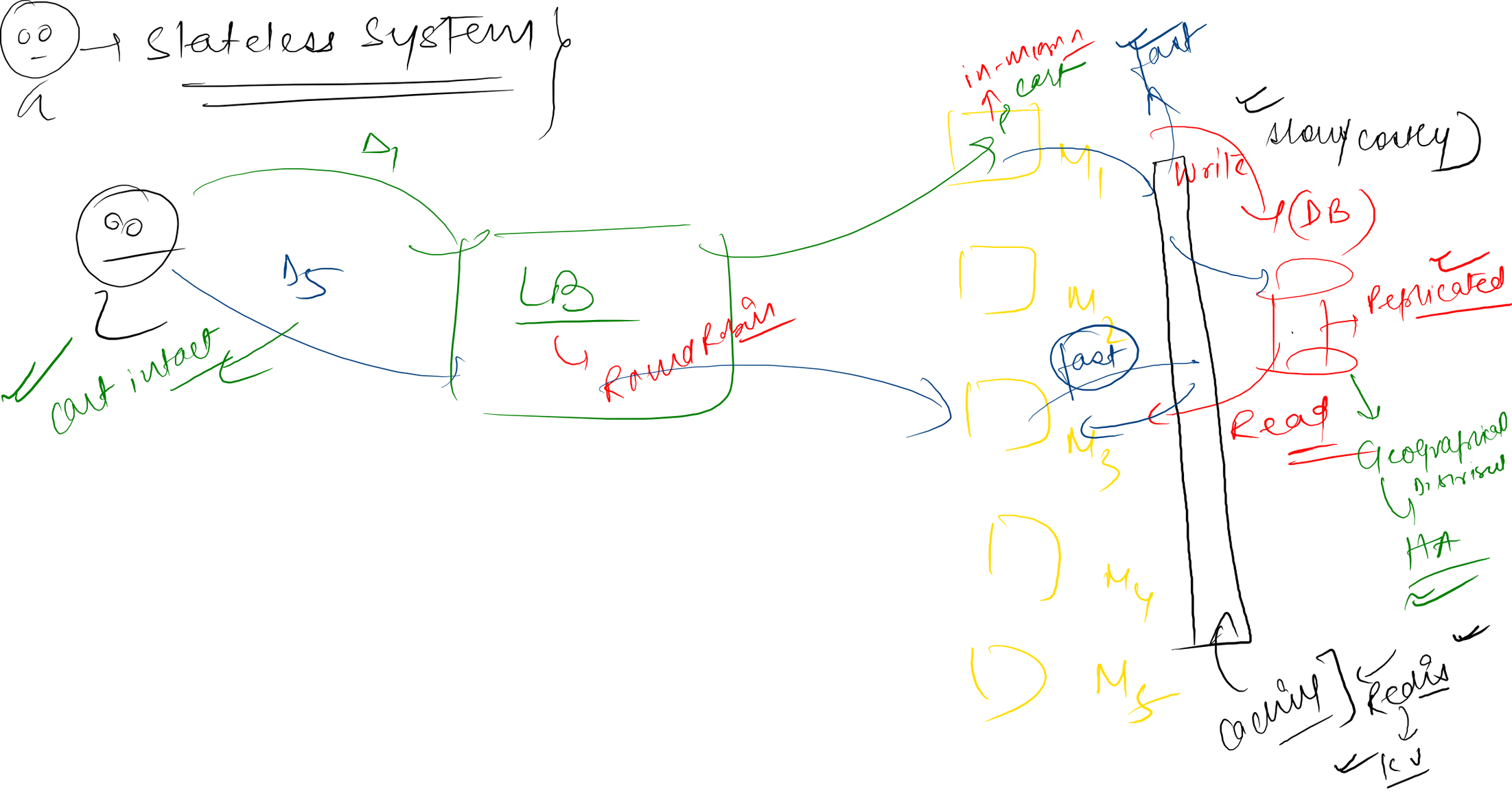

Security Gate
Book of Deer
Stateless
Resim
Stateless system
Move scalable application

① scalable system

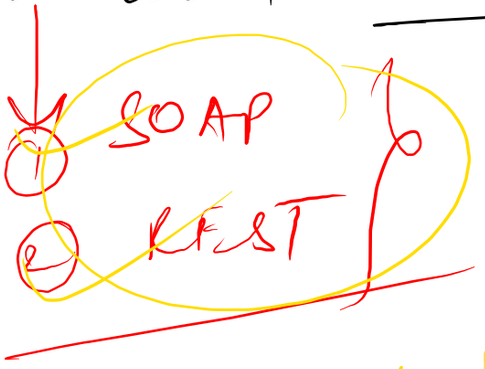
② fast failure recovery



stateless system


Stateless System
Δ1
Δ5
LB
Round Robin
cart intact
in-memory cart
M1
M2
M3
M4
M5
fast
slow copy
write
DB
Replicated
Read
Geographical
HA
caching
Redis
K V

communication protocols



HTTP

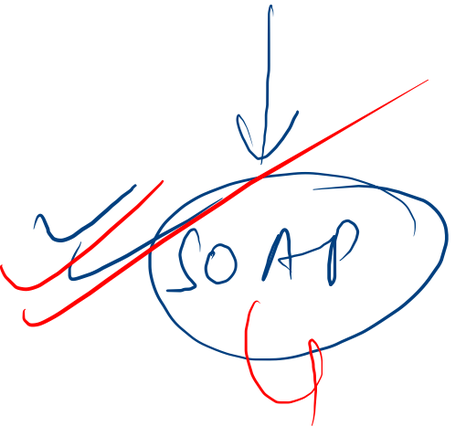
TCP / UDP

WebSocket / Server sent events

Long Polling

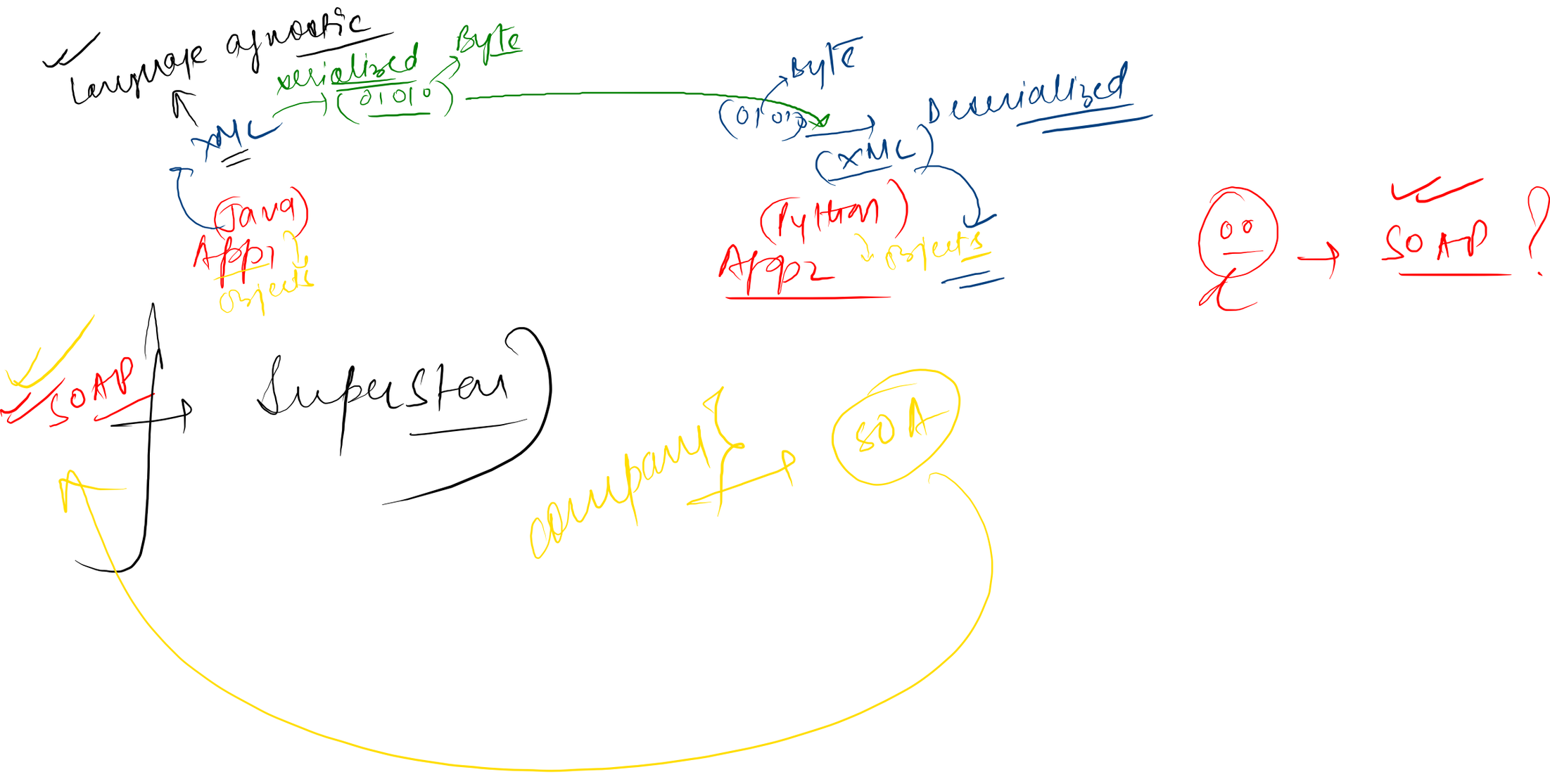

SOAP
NO
YES
10 years from now

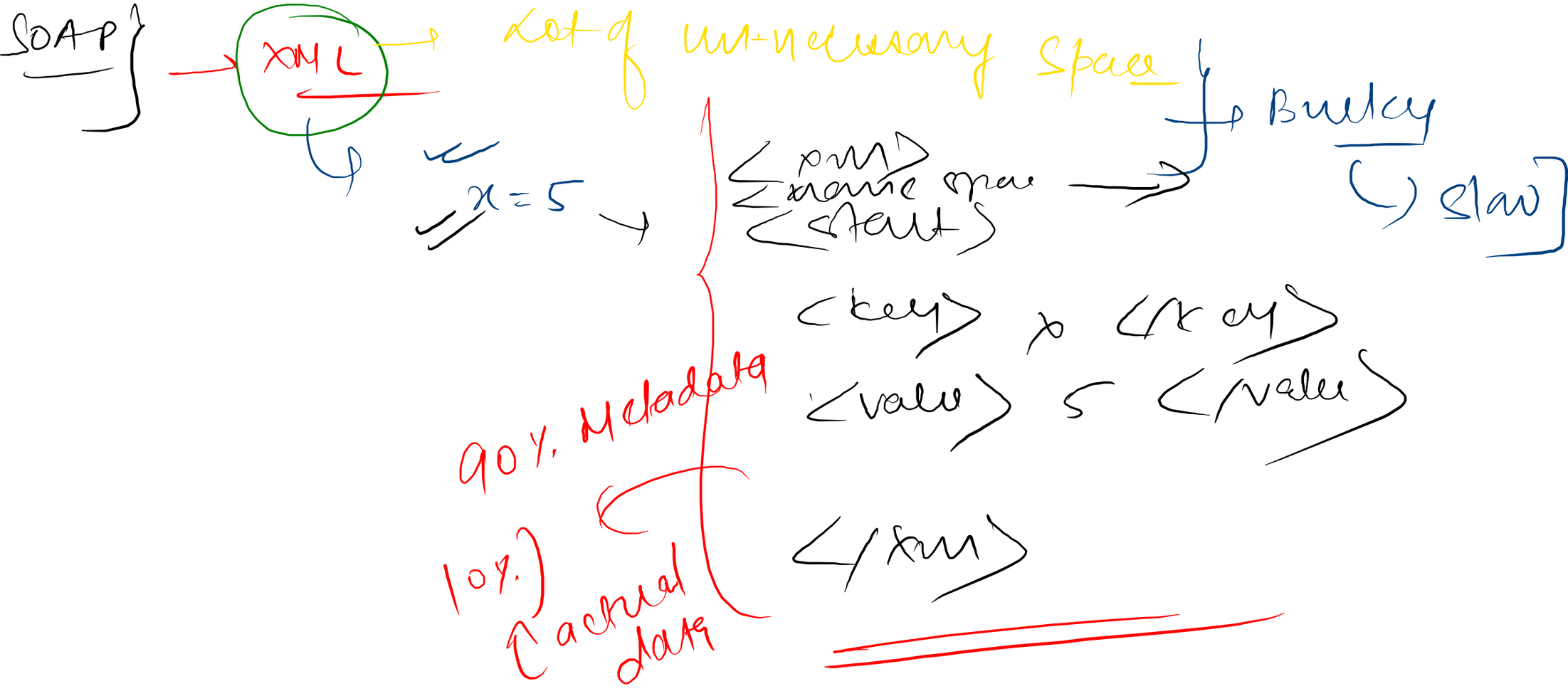
SOA

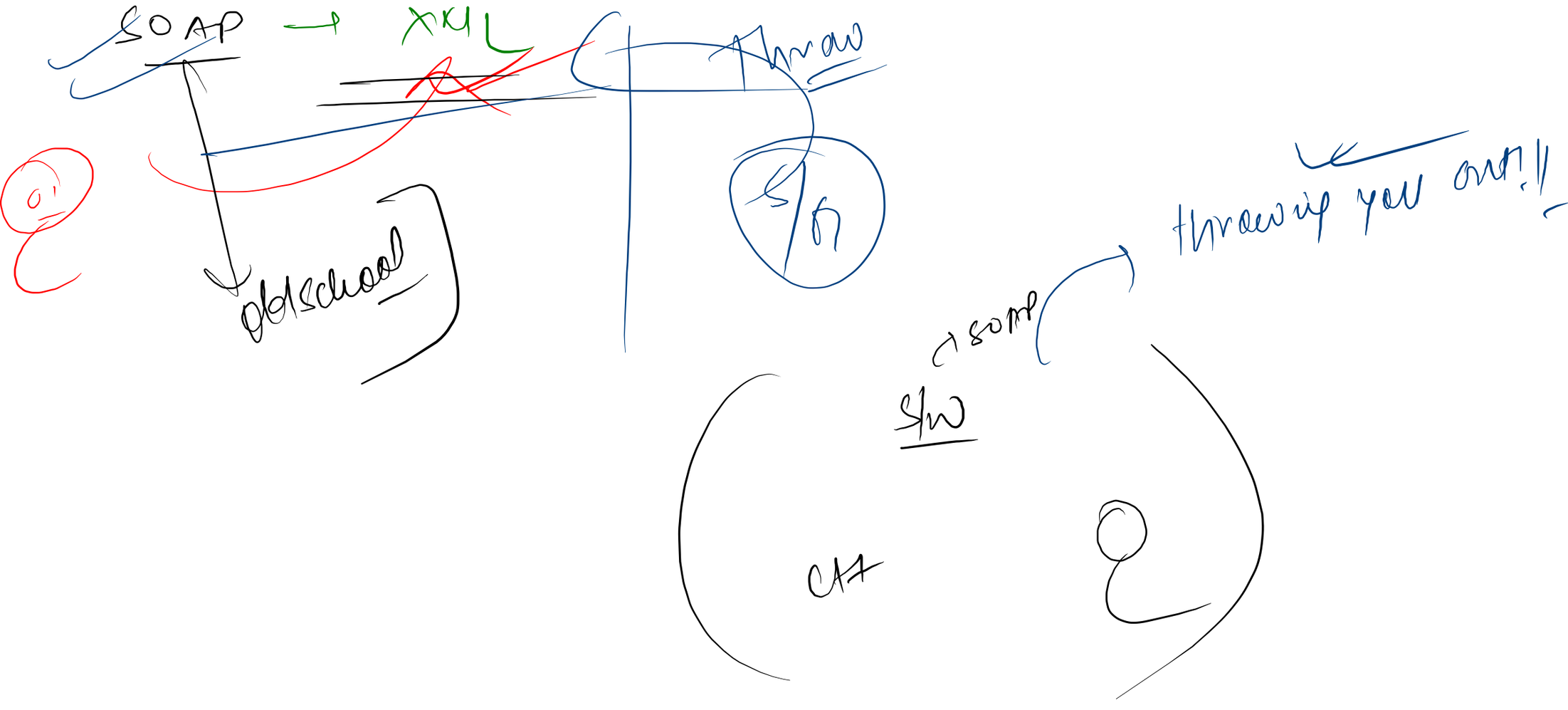


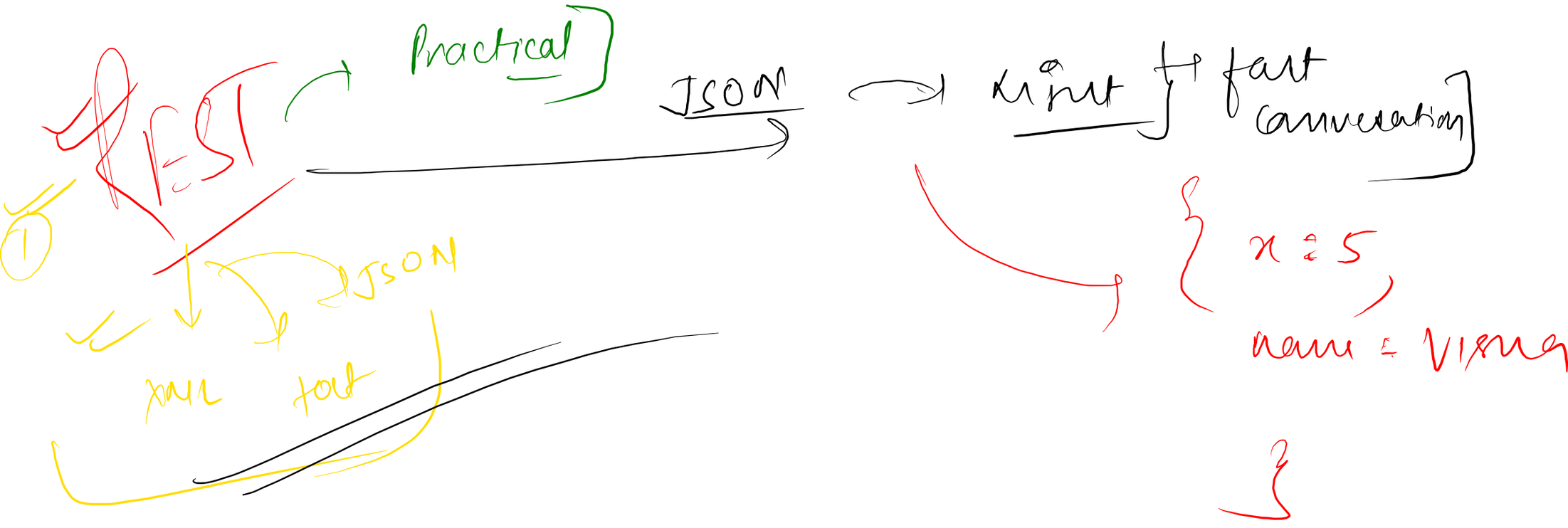
Simple Object Access Protocol

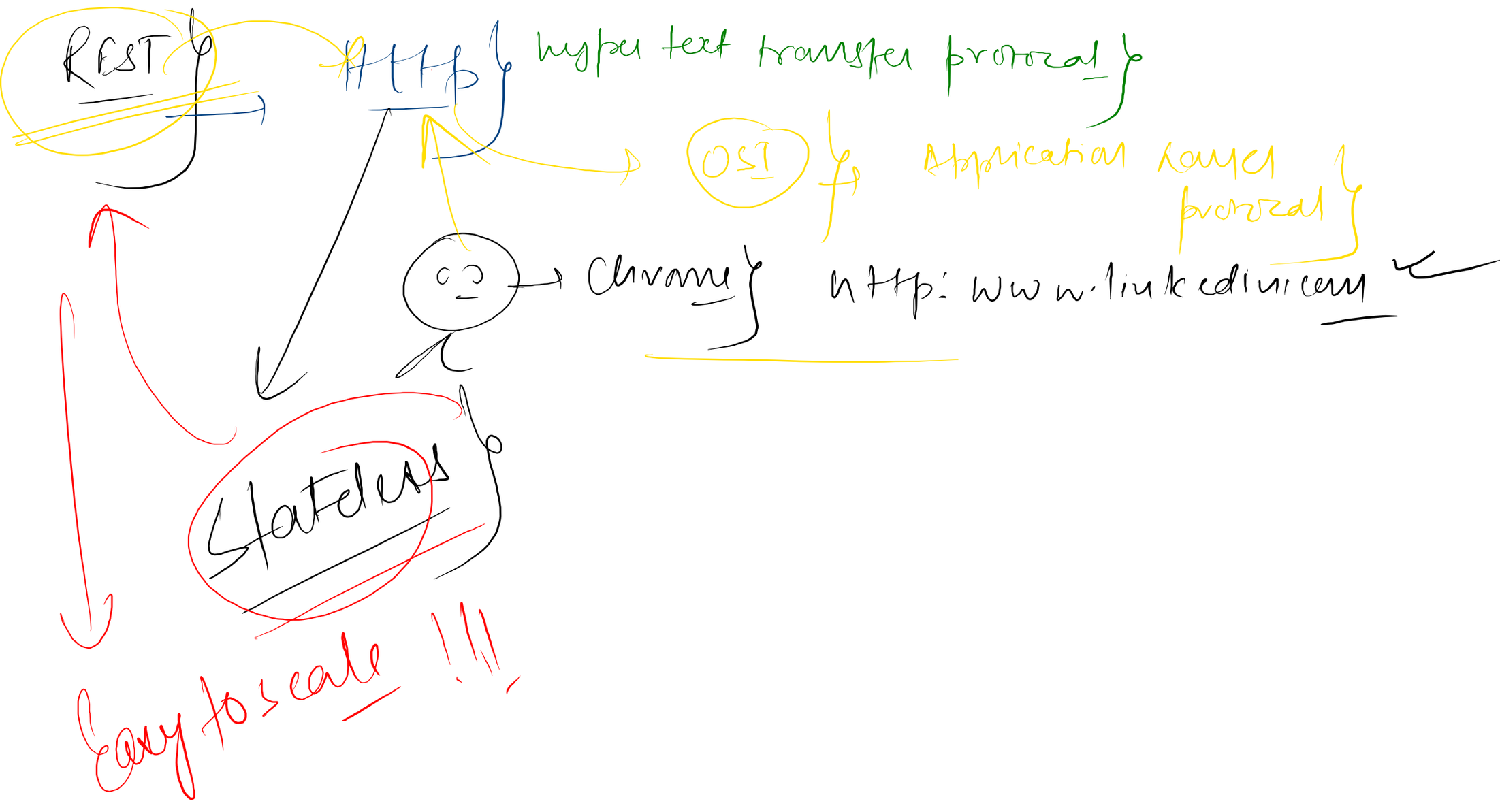
communicate
↳ share / transfer data


Language agnostic
XML
serialized
01010
Byte
Byte
01010
deserialized
(XML)
(Java)
App1
Objects
(Python)
App2
Objects
00
SOAP
SOAP
Superstar
company
SOA


SOAP
XML
Lot of un-necessary space
Bulky
slow
x = 5
<xml>
<name space>
<fault>
<key> x </key>
<value> 5 </value>
</xml>
90% Metadata
10% actual data


SOAP
XML
throw
old school
throwing you out!!
SOAP
S/W
CAT

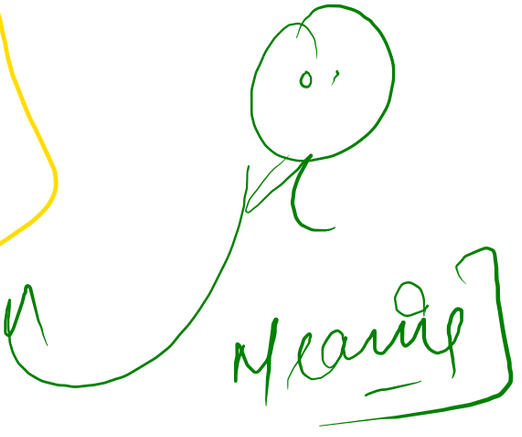

REST
Practical
JSON
Flask
fast communication
1
JSON
XML
text
x = 5
name = Vishnu

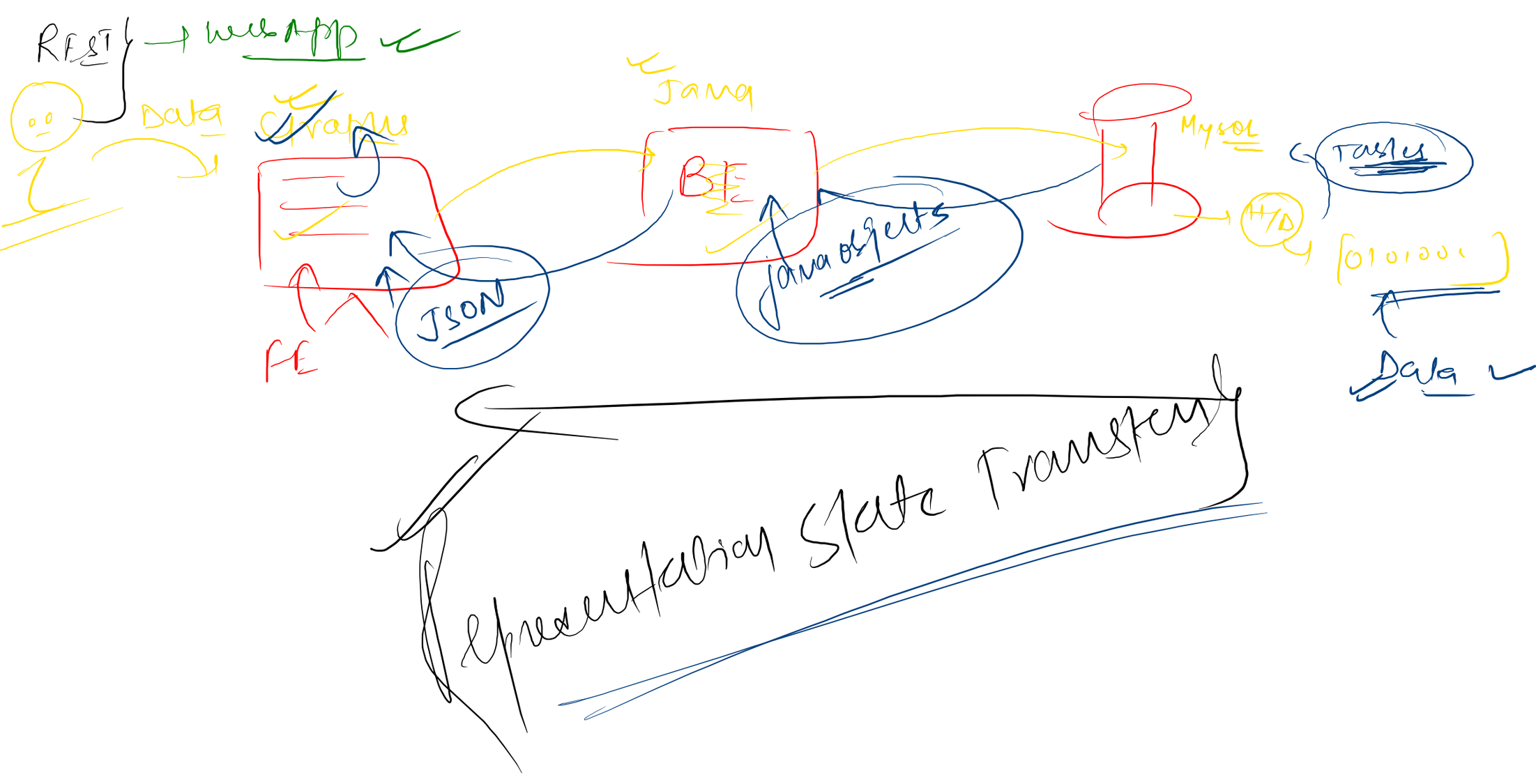

REST
HTTP
hyper text transfer protocol
OSI
Application layer protocol
Chrome
http://www.linkedin.com
Stateless
Easy to scale !!!

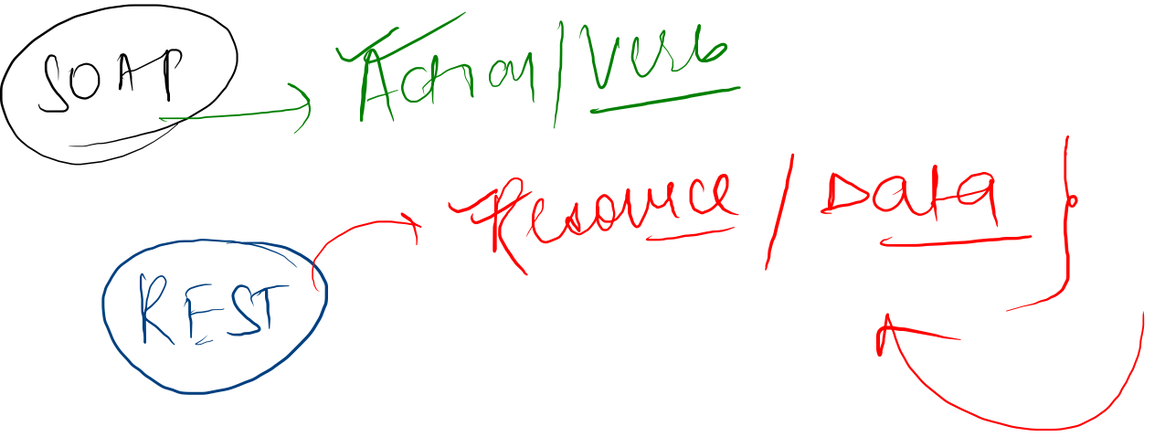
REST ?

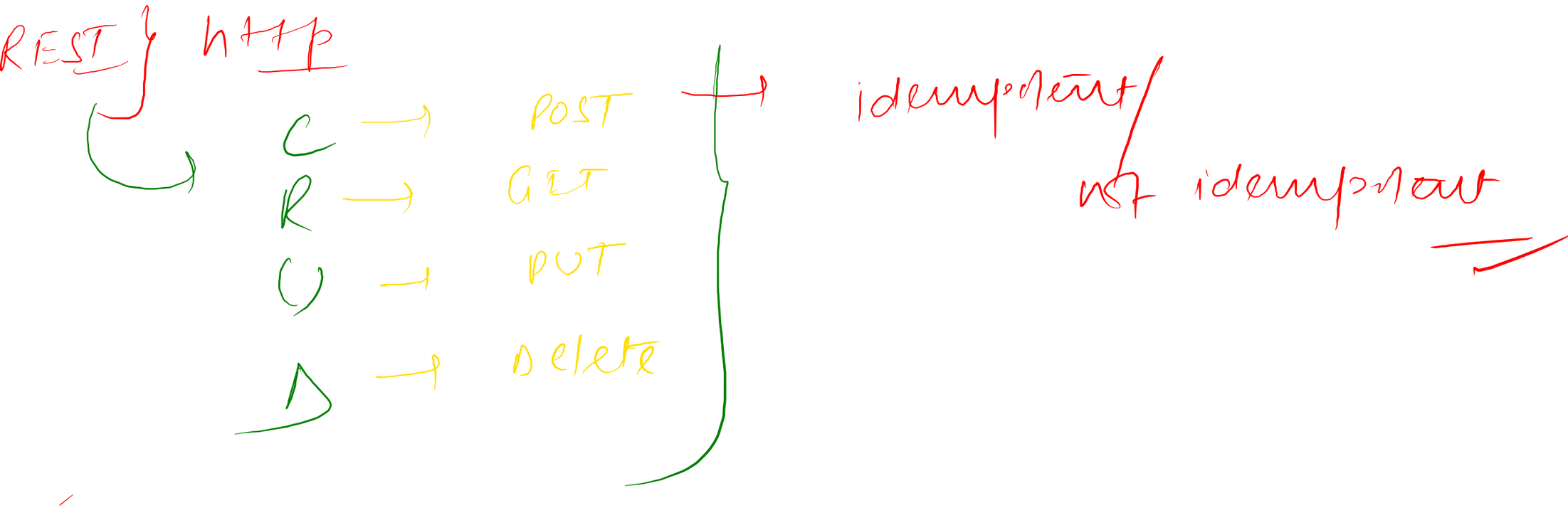
Representational State Transfer

Meaning ?




REST
Web App
Data
Angular
Java
MySQL
BE
JSON
Java objects
Tables
H/D
FE
Data
Representational State Transfer


SOAP
Action / Verb
REST
Resource / Data


REST
http
C
R
U
D
POST
GET
PUT
Delete
idempotent
not idempotent

idempotent

call once / call Multiple times

No side effects

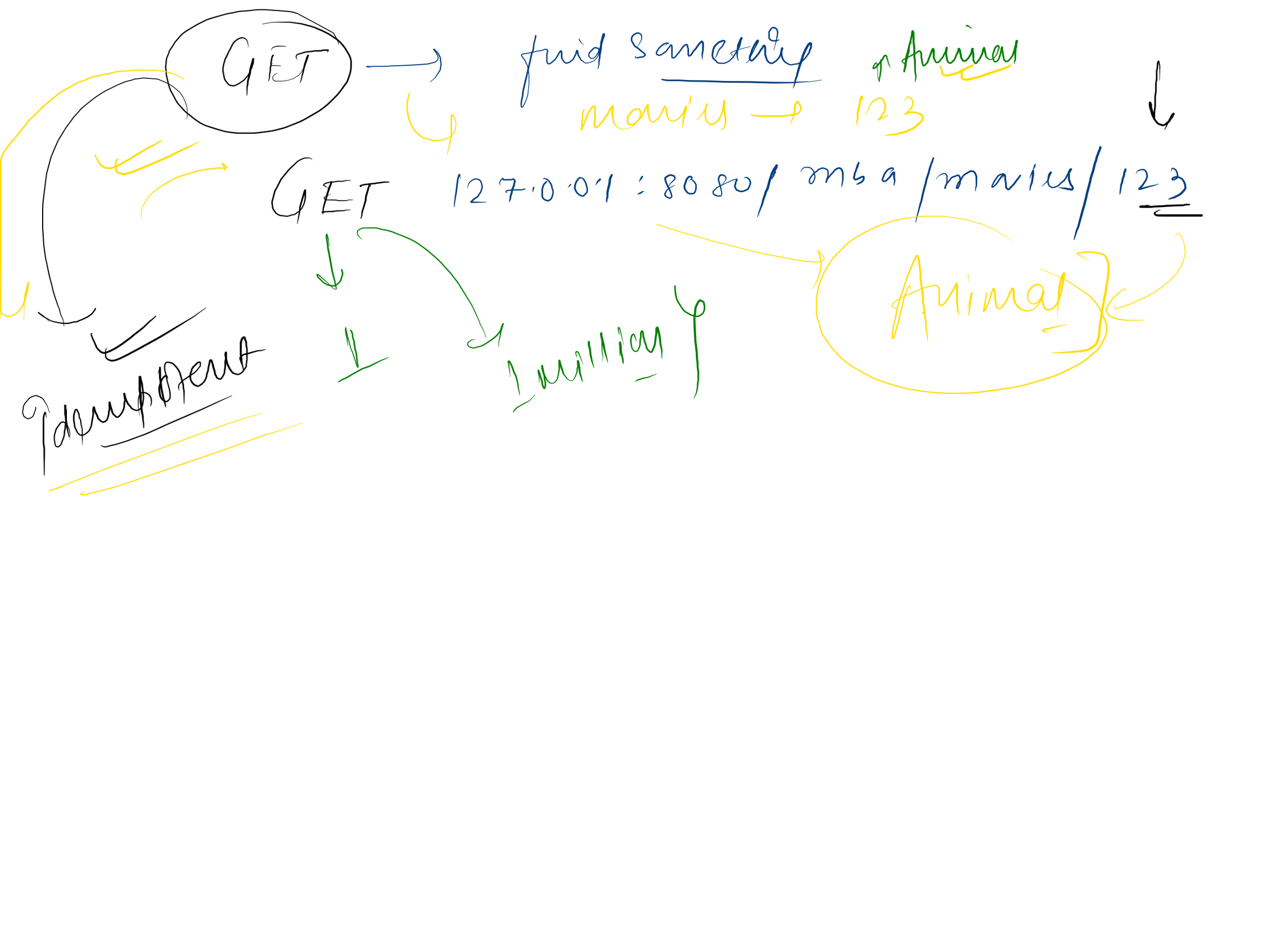
same outcome

GP

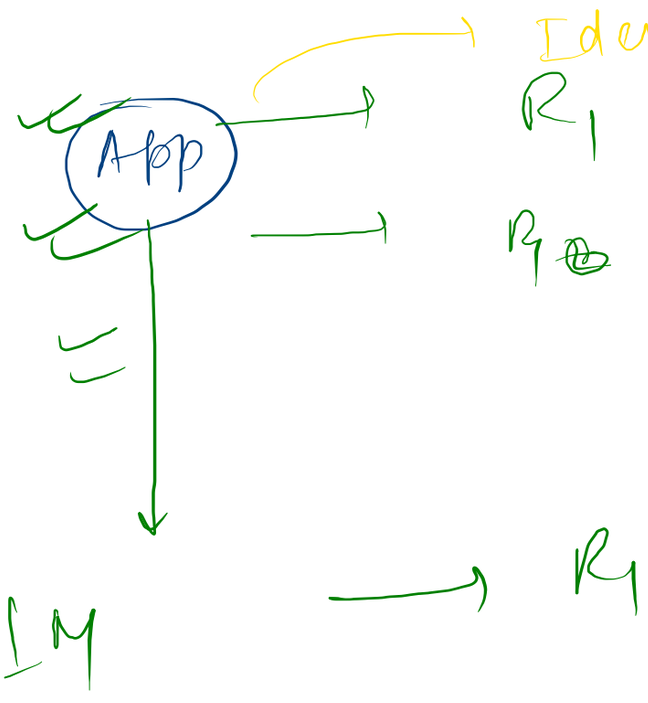
Payment → Rs 100

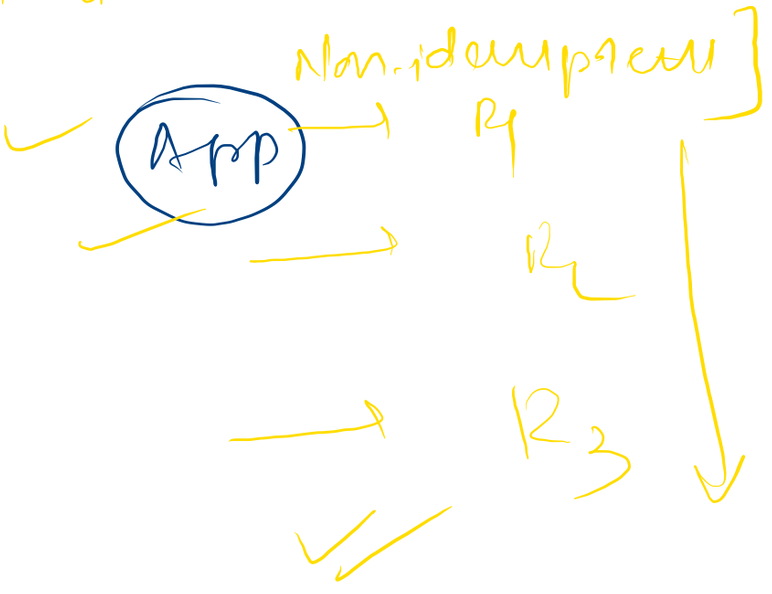
Problem

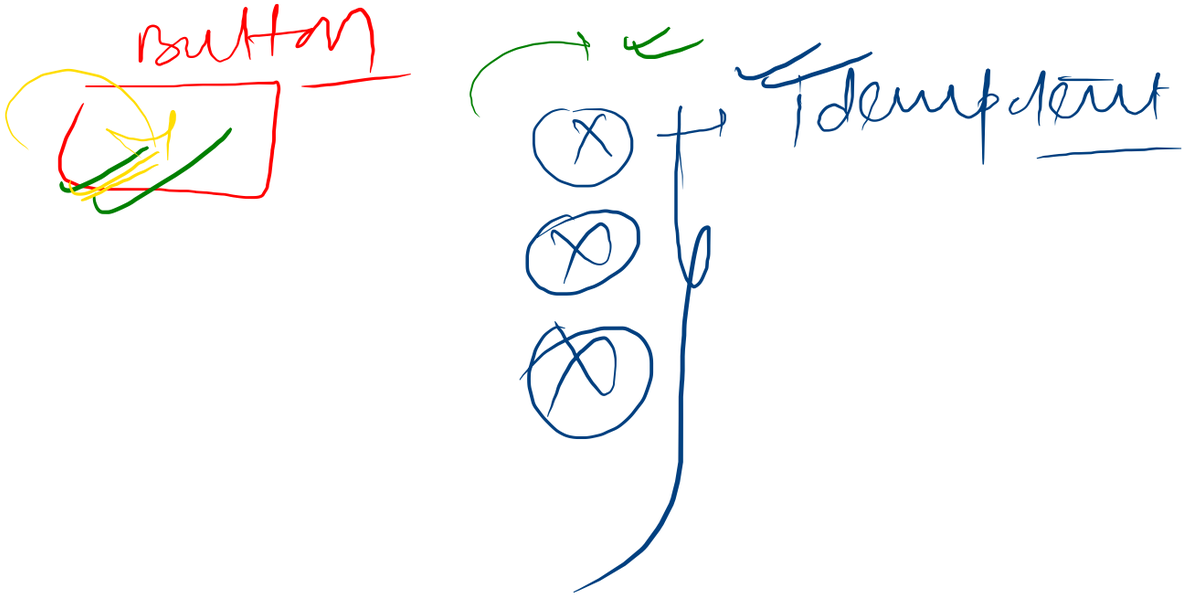
Pay 1/10


GET
fuid Sametip or Animal
mavis → 123
GET
127.0.0.1 : 8080 / mba / mavis / 123
Animal
Animal

Idempotent






Button
Template

GET

PUT → idempotent

Update

id = 5
name → x
y

PUT 127.0.0.1:8080/app1/users/5

{
"name" : y
}

idempotent nature

1 → name → y
x
name = y

2 → name → y

3 → name → y

Million → name → y

PUT → update

PUT 127.0.0.1:8080/vms/vehicles/123

{ "color" : "black" }

Same Result

Idempotent

vehicle ← 123

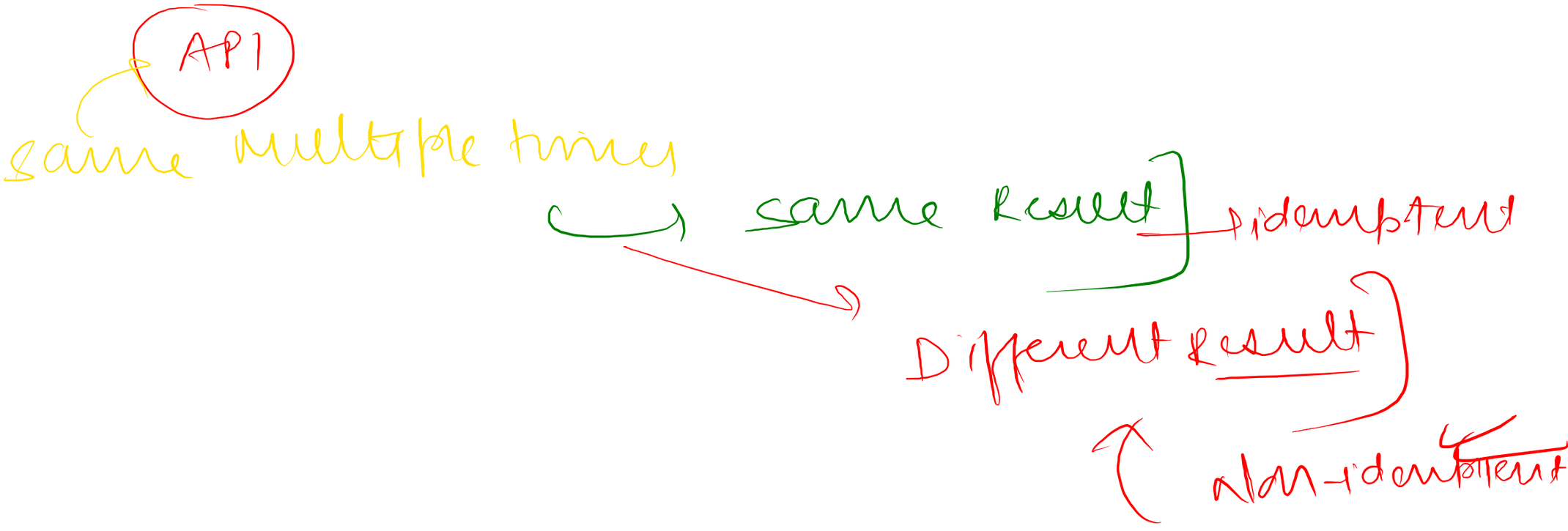
color = yellow ← black

1 PUT → clr = black

2 PUT → clr = black

3 PUT → clr = black

1 million → clr = black


API
same multiple times
same result
idempotent
different result
non-idempotent

Create a movies

↓ POST

POST 127.0.0.1:8080/mba/movies

{ name: xyz
language: English
}

Non-idempotent

2nd

2nd

{id:3
name:xyz
len:xy
}

→ Newresource

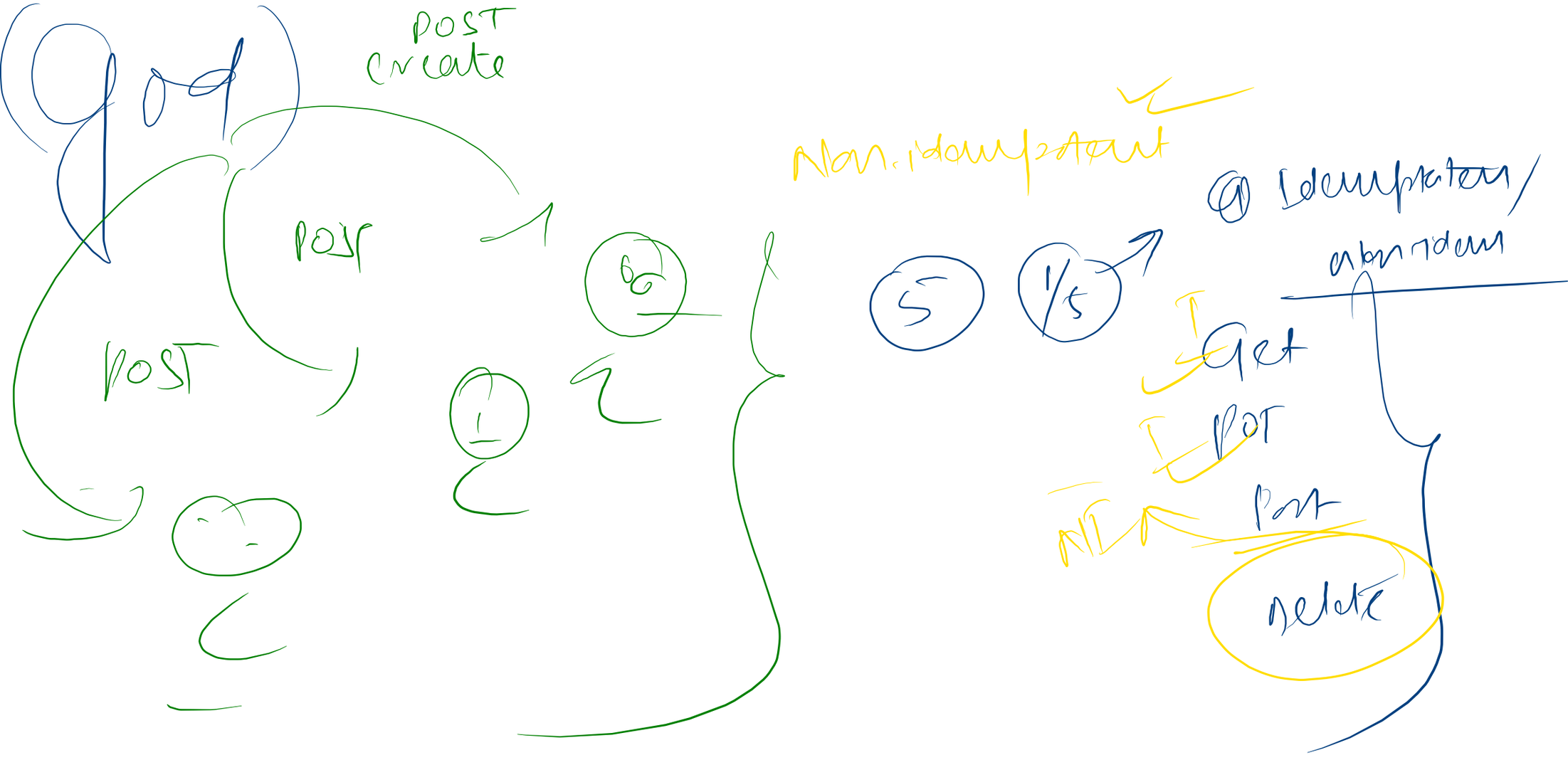
{ id: 1
name = xyz
language: eglish
}

{ id: 2
name: xyz
len: xy
}

POST → Non-idempotent

creation of the resources

create something

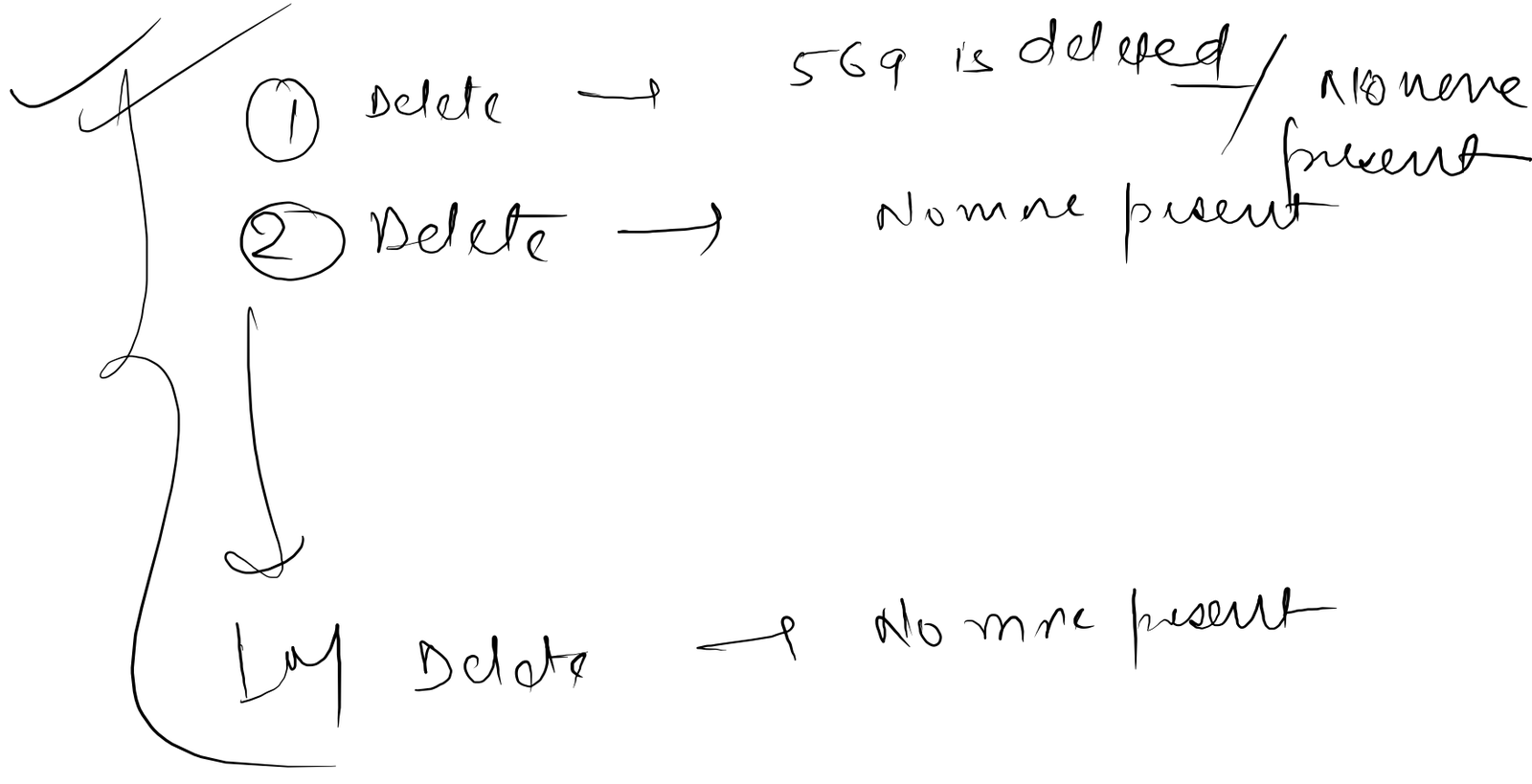

POST
create
POST
POST
GET
PUT
PUT
DELETE

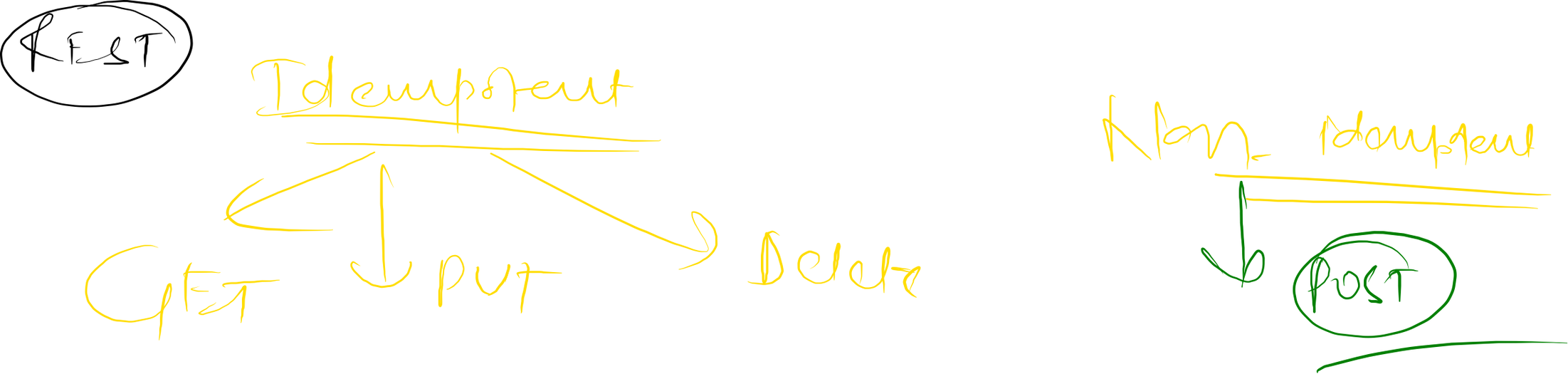
Delete → Idempotent

DELETE 127.0.0.1:8080/mba/movies/569

Movie → id = 569 ← Remove

Idempotent / Non-idempotent




REST
Idempotent
GET
PUT
Delete
Non Idempotent
POST

Put and Patch (APIs)

Put / Patch → updation

PUT 127.0.0.1:8080/api/users/123

user

name : Virat 29

age : 99

sub : DPS

class : 11T

Put → complete replacement

Patch → partial

Jupiter

name : Virat

age : 29

sub : DPS

class : 11T

age : 29

Key data

Pass the key which is used

Entire object

8:35 PM

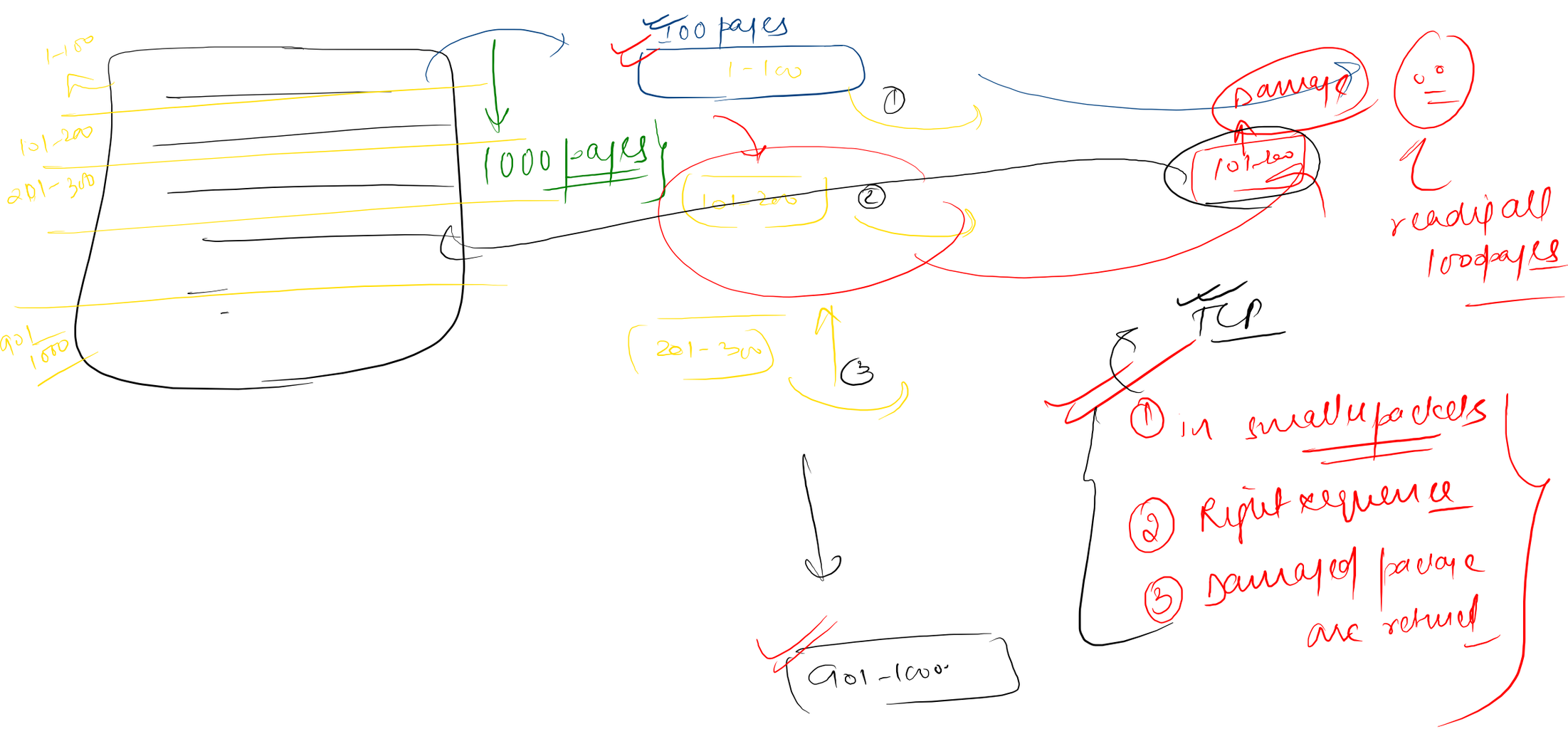
10 minutes

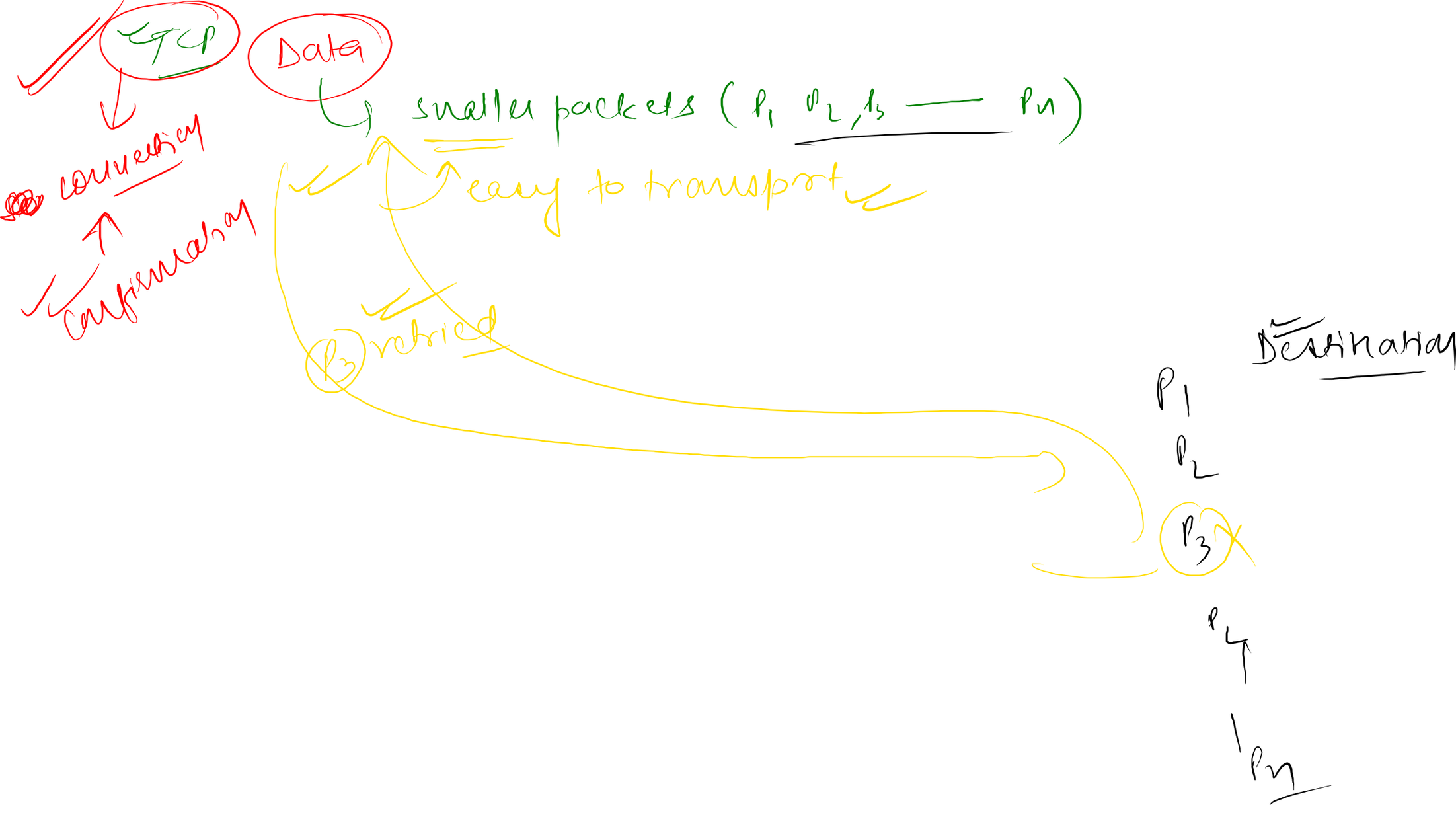
Break

communication entity

TCP → Transmission control protocol

Provide information
that communication is not / not → Retry

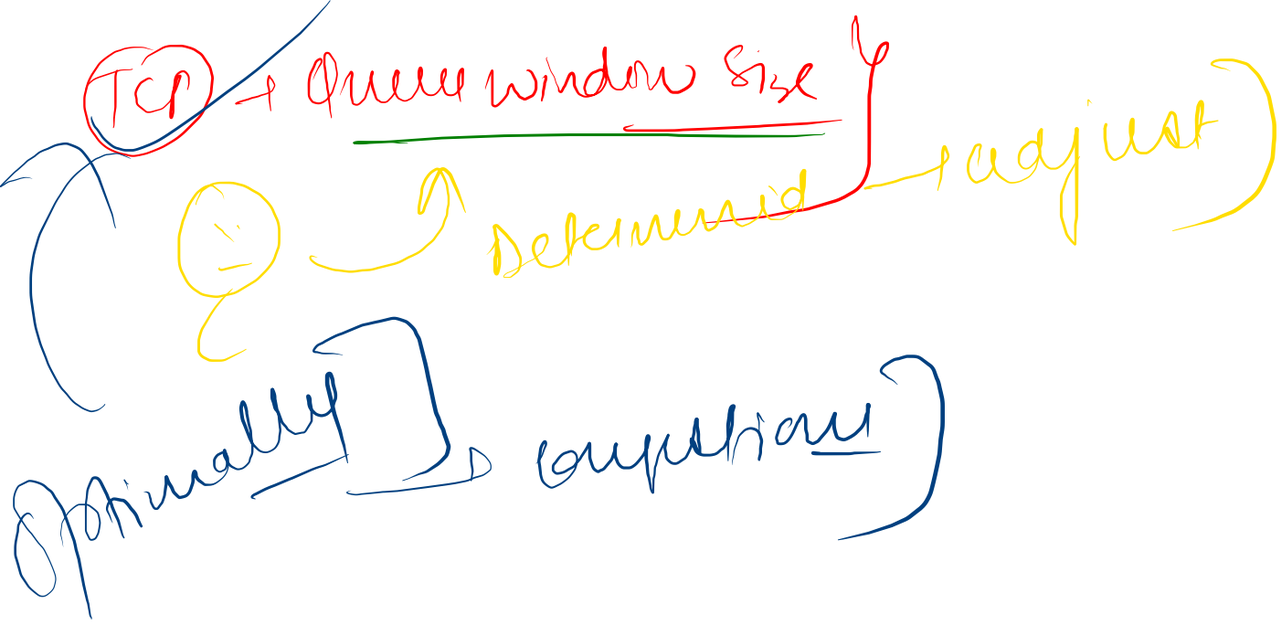

1-100
101-200
201-300
901-1000
1000 pages
100 pages
1-100
1
101-200
2
201-300
3
901-1000
Damage
101-200
read all 1000 pages
TCP
① in small packets
② Right sequence
③ Damaged package are retried

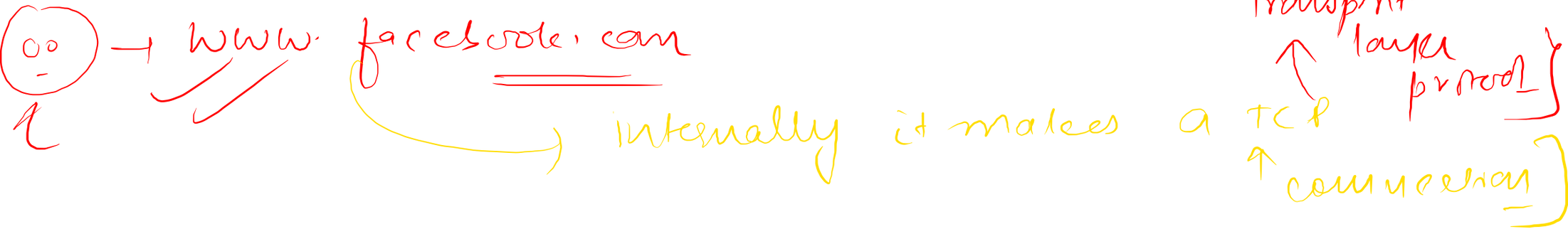

TCP
Data
smaller packets (P1 P2 P3 —— Pn)
easy to transport
connection
confirmation
P3 retried
Destination
P1
P2
P3
P4
Pn

TCP → ① connection
② confirmation

connection → slow connection

communication

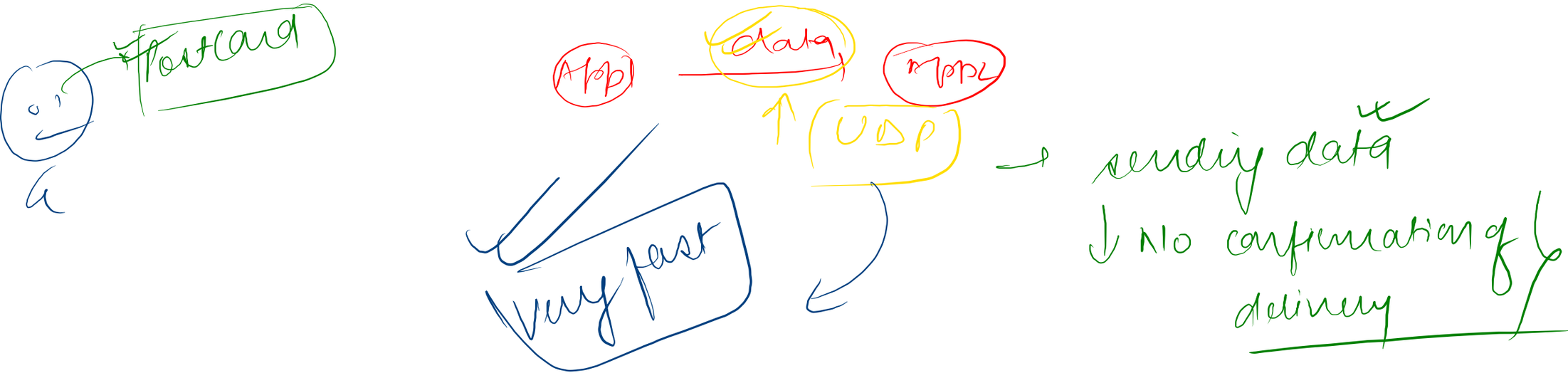

TCP
Queue window size
determine
adjust
Optimally
congestion


www.facebook.com
Internally it makes a TCP connection
Transport layer protocol

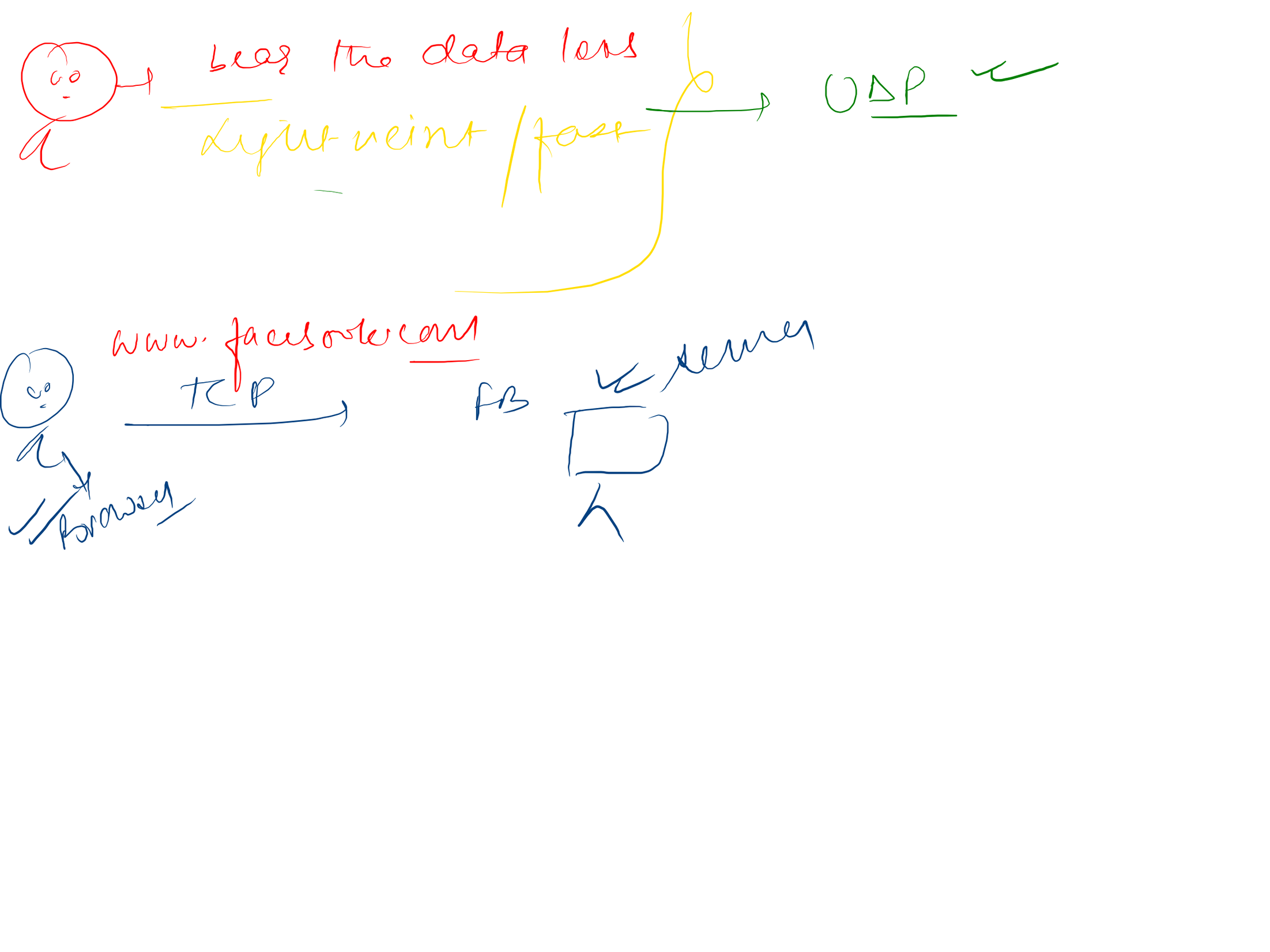
UDP → User Datagram Protocol

→ send the data

↑ don't care → checks to see if they received

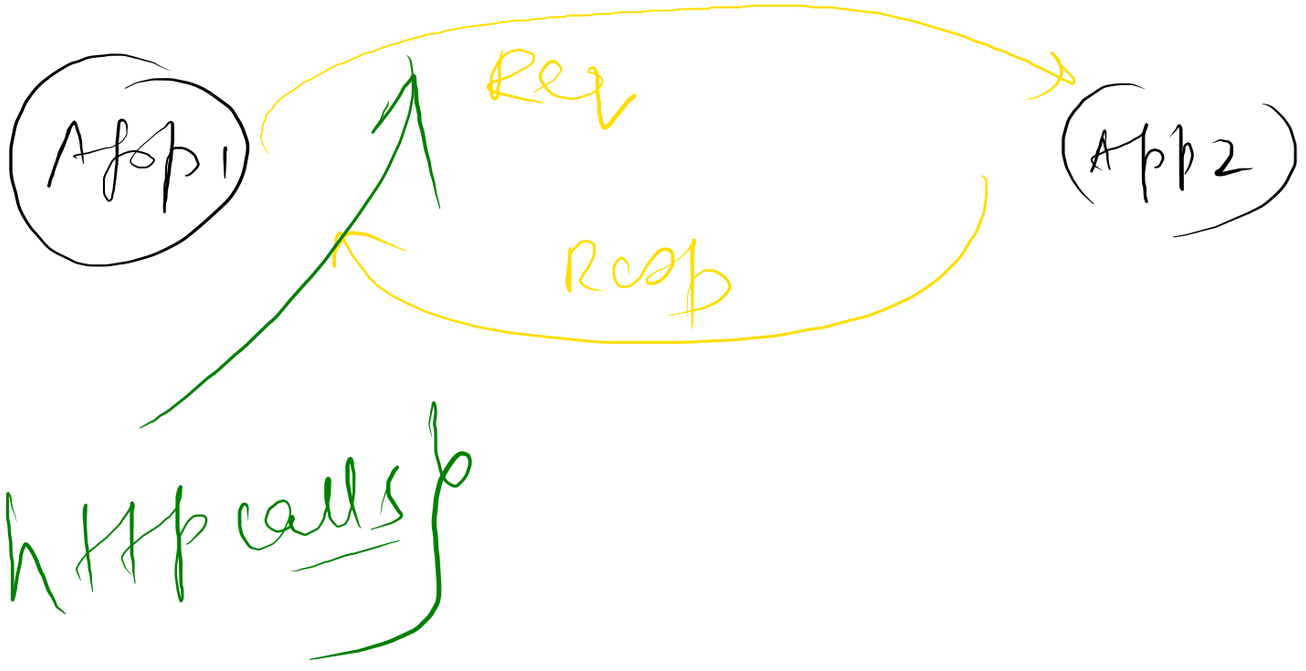

Postcard
App1
Data
App2
UDP
very fast
sends data
No confirmation of delivery


Zoom
UDP
Video streaming
Transfer very fast


Less the data loss
Light weight / fast
UDP
www.facebook.com
TCP
FB
Server
Browser

HTTP Connectivity

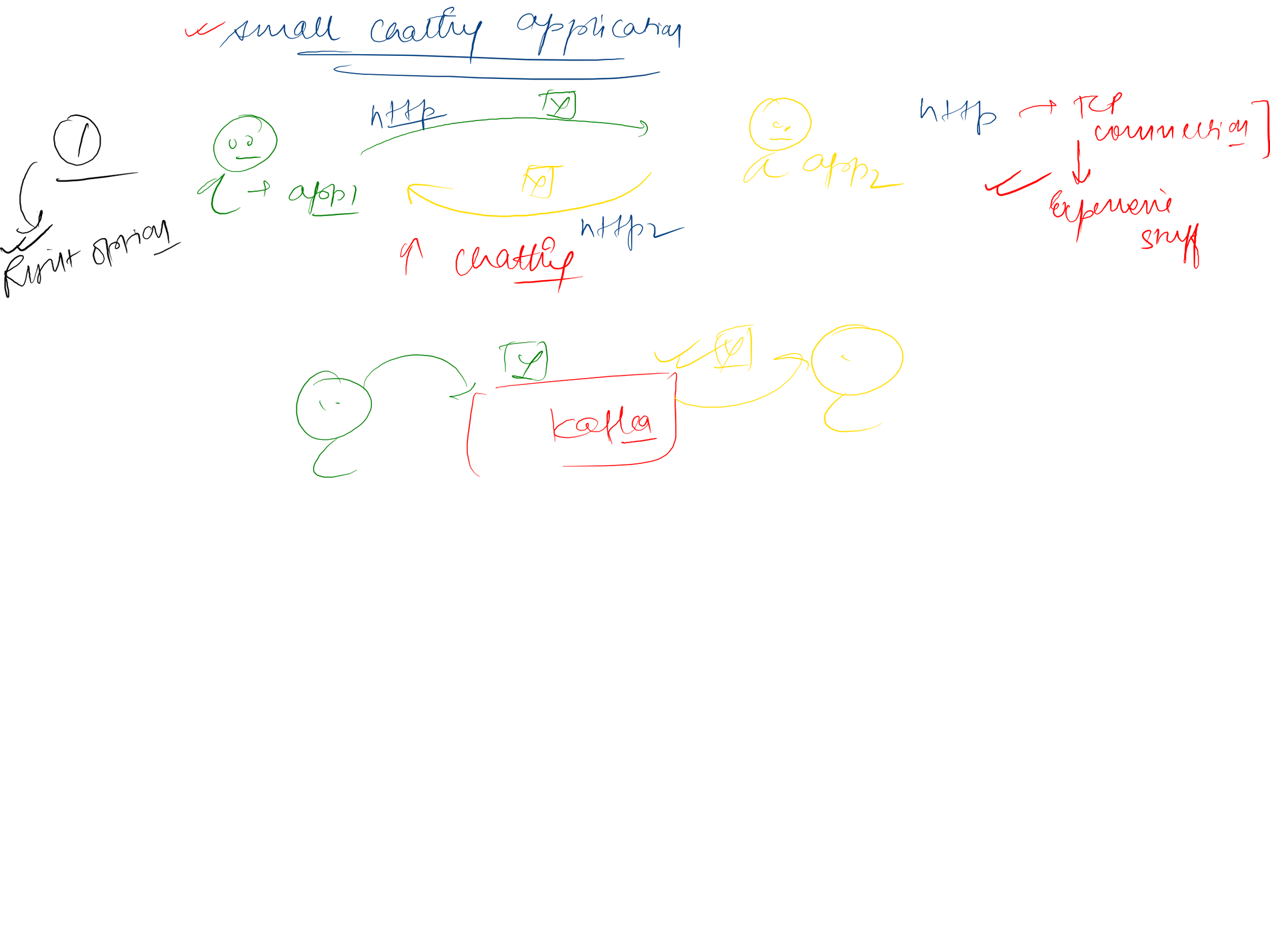
Application layer protocol


App1
Req
App2
Resp
http calls

facebook.com ✓ http://www.facebook.com

↳ Application layer called http

↳ Transport layer calls is TCP


small chatting application
1
Rust app
http
app1
app2
chatting
http2
http → TCP connection
experience stuff
kafka

Web Sockets

Long term connection

2 way communication

TCP

short term

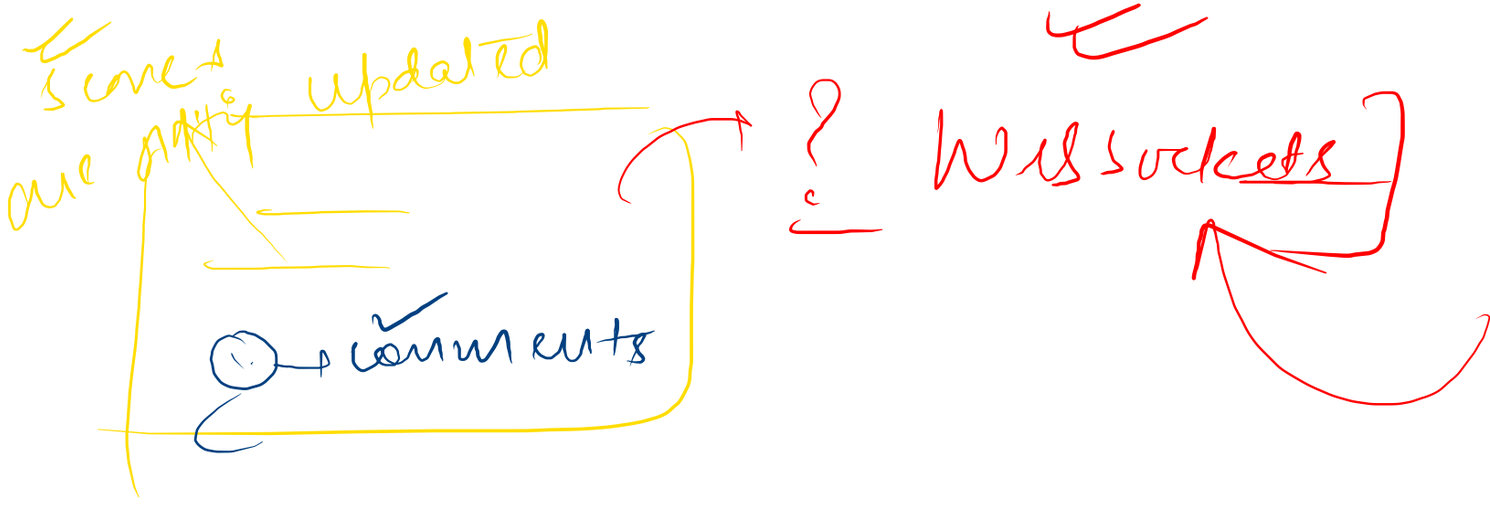
only

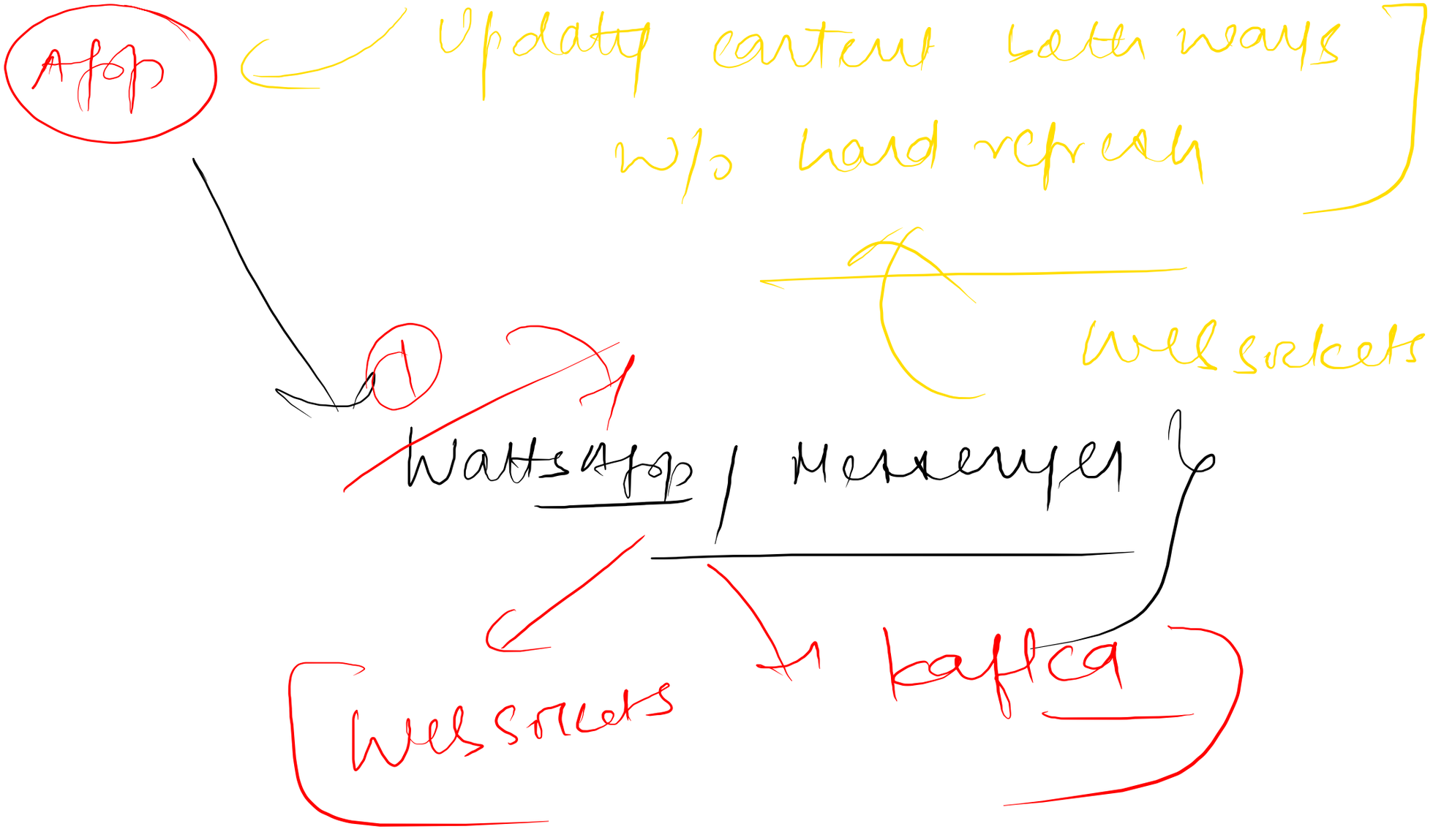
single communication

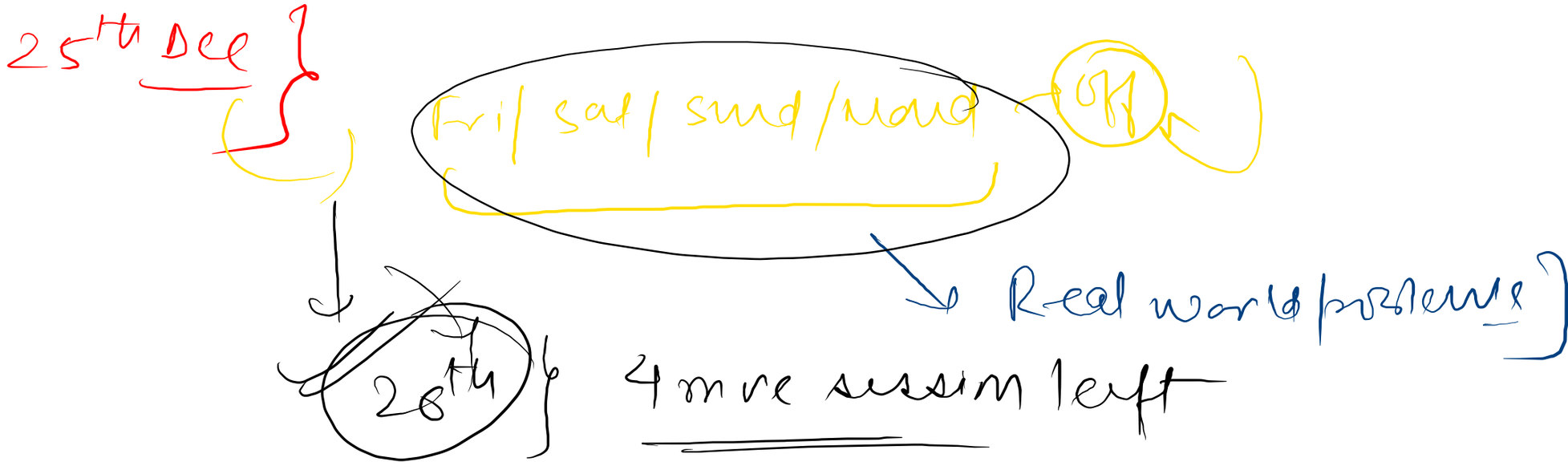
connection tunnel

2-way communication

expr avicinfo




App
Updating content both ways
w/o hard refresh
WebSockets
WhatsApp / Messenger
WebSockets
kafka


25th Dec
Fri/ sat/ sund/ mond
OK
Real world problems
28th
4 more sessions left

Assignment

Web Crawler

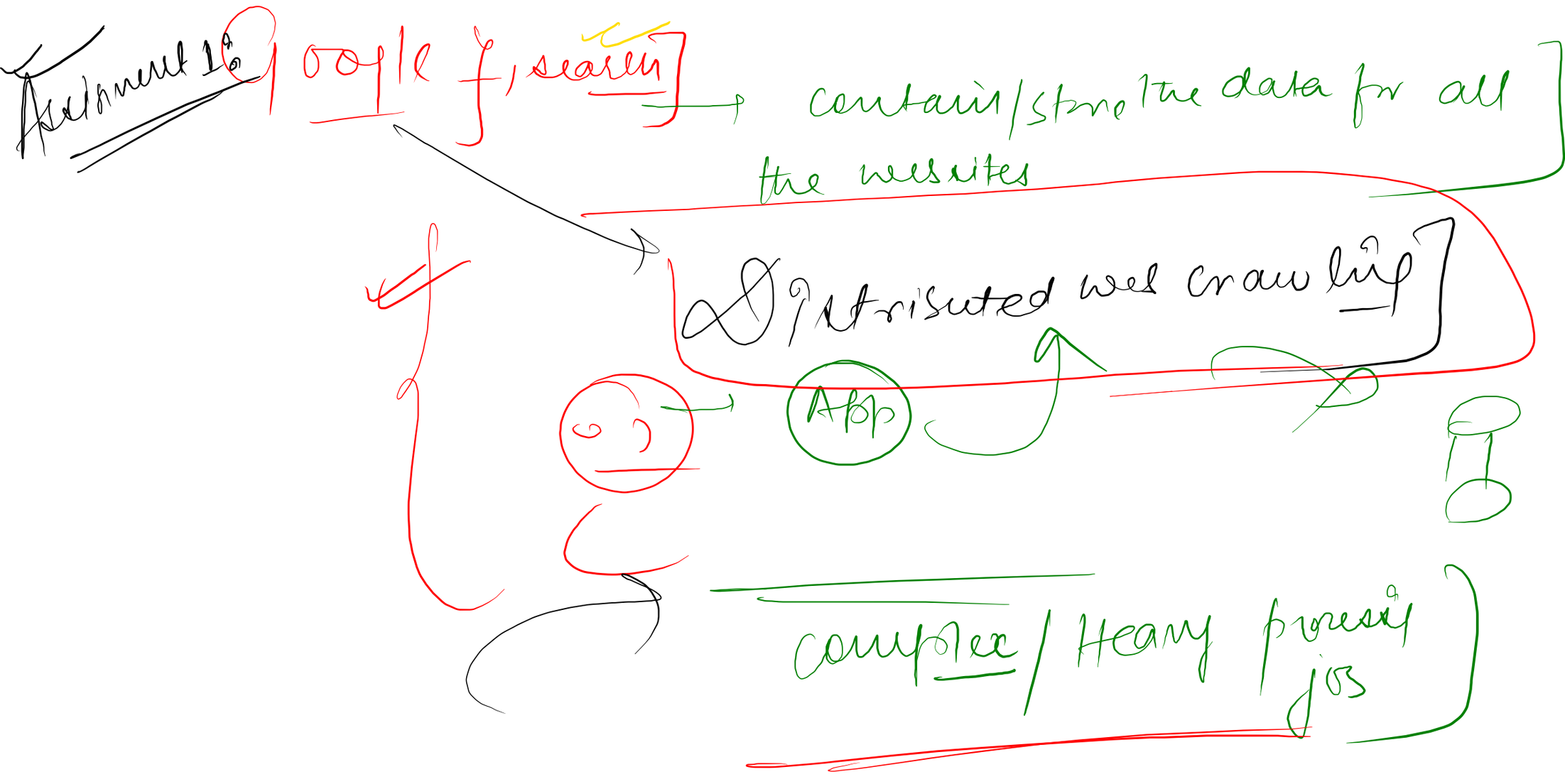
app/soft

↳ website

scan

Your website files

↳ W.


Assignment 1
Google search
contain/store the data for all the websites
Distributed web crawling
App
complex / Heavy processing jobs

NoSQL → Natively scalable / way ? → BASE

Distributed system

Distributed system

Column DB → store columns together → faster aggregation

Aggregation

sum salary
avg salary
max salary

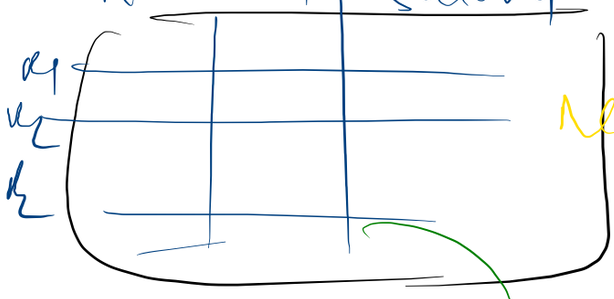
name | age | salary

one place

1 million

Row based

Name Age Salary



Column based

Map

Near(salary)

Memory

R1, R2, R3

Raw is stored together

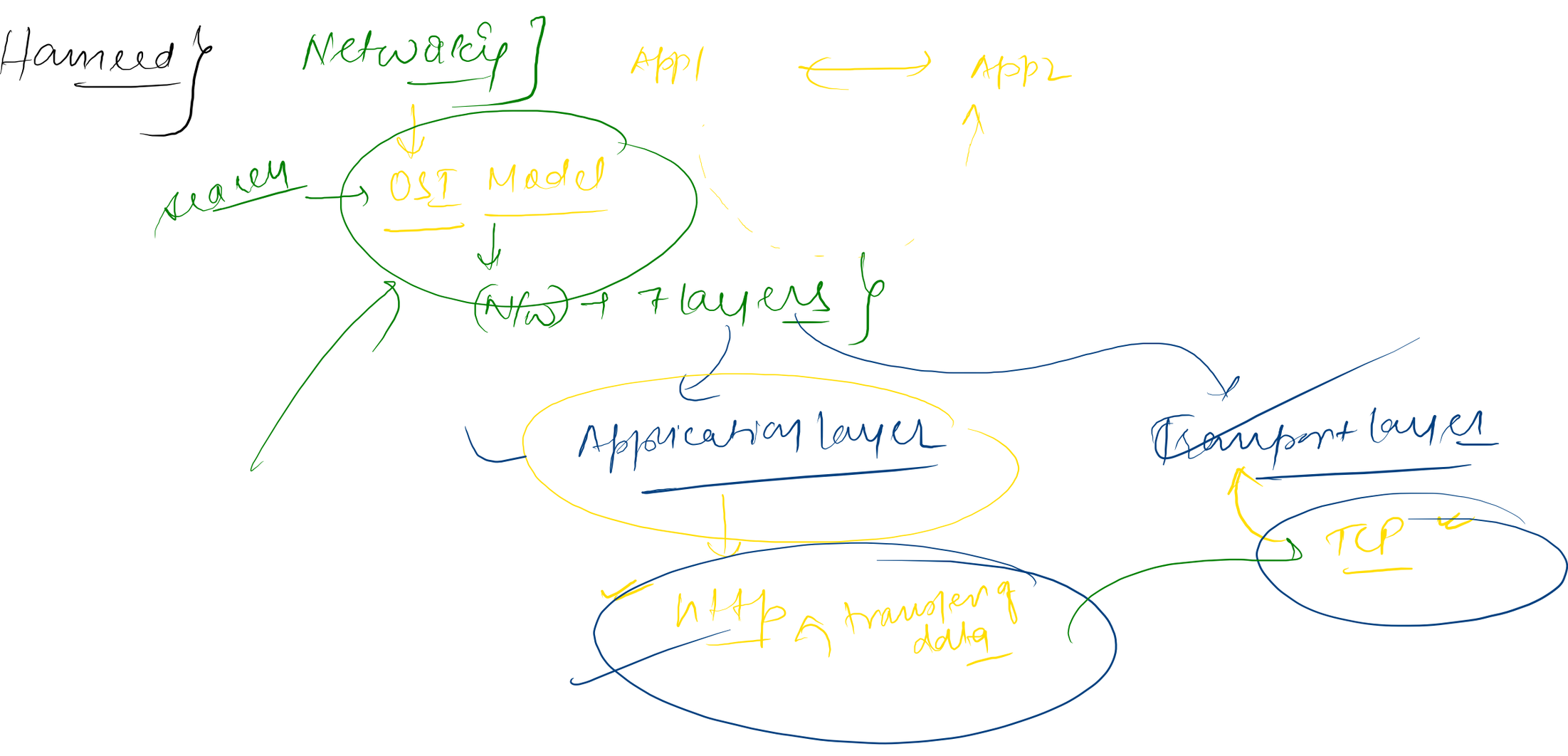
Memory

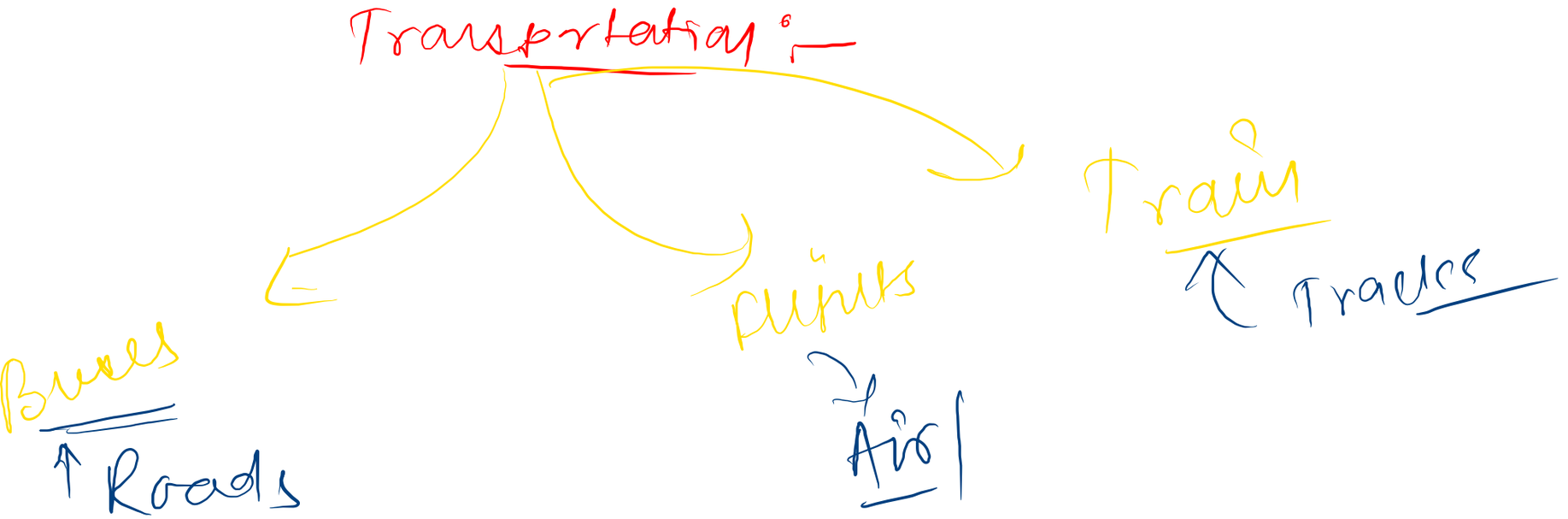
Name

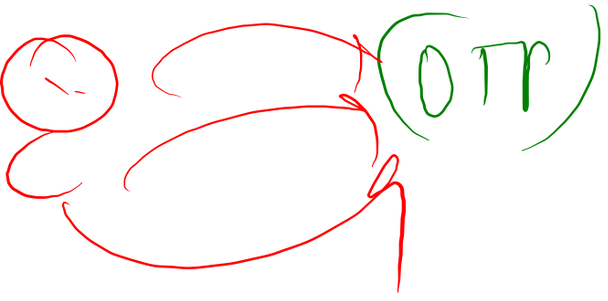
Age

Salary

26th is the Next class!!!


Hameed
Network
App1
App2
layer
OSI Model
(N/w) → 7 layers
Application layer
Transport layer
TCP
HTTP → transfer of data


Transportation
Buses
Roads
Planes
Air
Train
Tracks



3459
5169
7348

non-deterministic