

1. Good Evening
2. We will begin at 9:10 pm
3. Topic → Cookies, TCP, UDP

Agenda

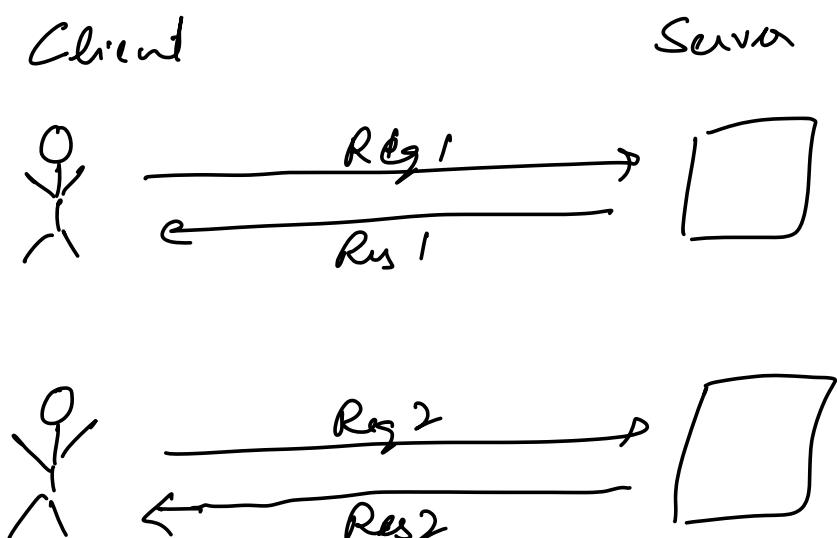
1. HTTP Cookies
2. TCP and UDP → Differences & Similarities
3. How TCP works
 - ↳ Connection establishment
 - ↳ Data flow → SYN & ACK
 - ↳ Sliding window.
4. Details of DNS
 - ↳ Hierarchical System

HTTP Cookies

Q1. What are cookies?

Q2. Why are they required

* HTTP is a stateless protocol



For Req2 Server wouldn't remember anything about Req1 from client

* [If Server doesn't remember anything] about client then

→ 1. How does login work?

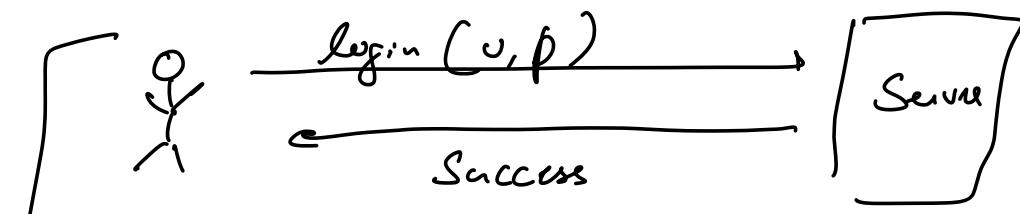
2.1 How do video-streaming websites show "Continue Watching" feature to guest users?

2.2. → How carts work on e-commerce websites

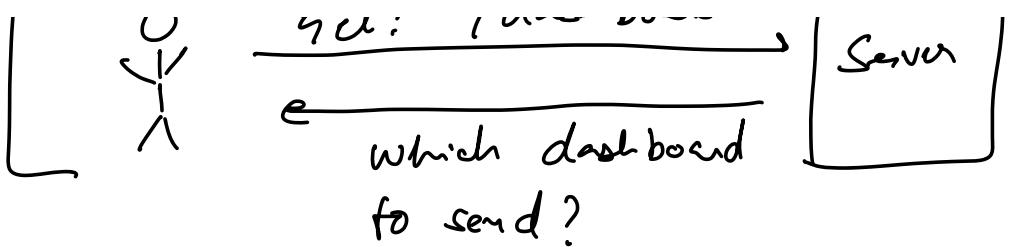
3. How do project managers track user activity & plan page structure for customer subscription?

→ Why do websites have to take user permission before saving cookies?

~~Login~~ Client Server

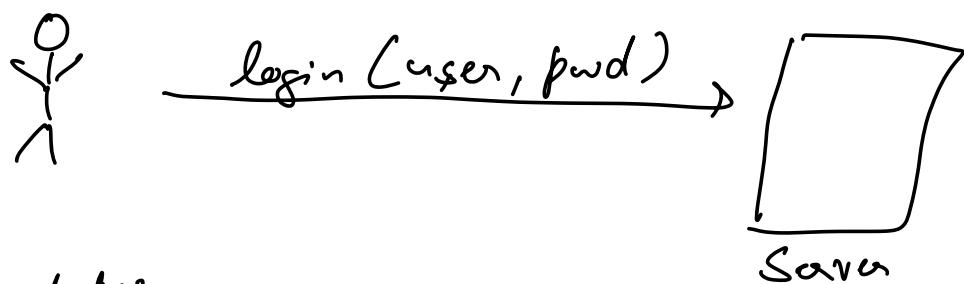


a +. /dark board]

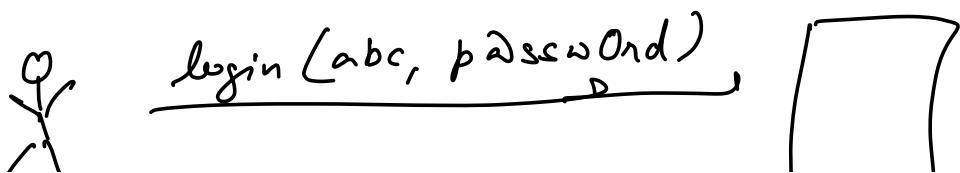


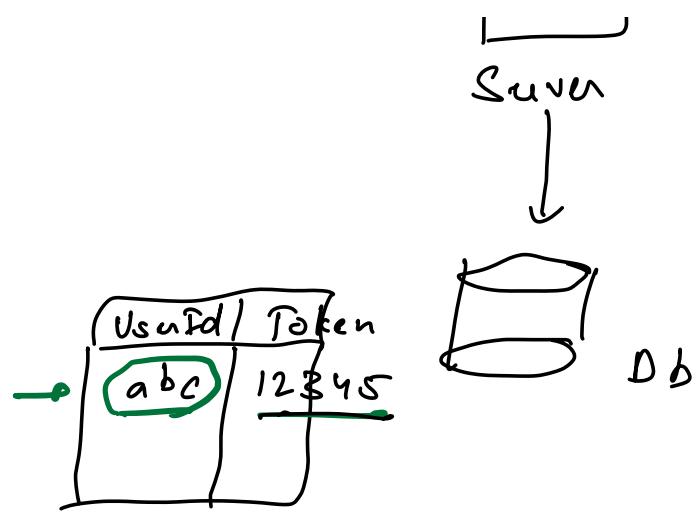
Steps

1. Client passes login details ✓

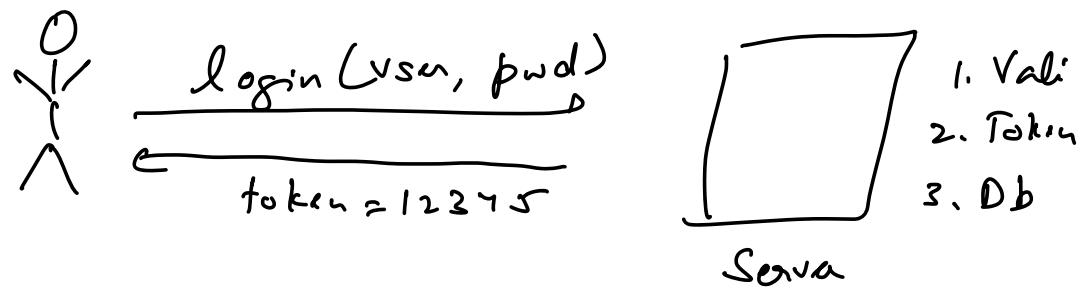


2. Server checks if usrid & pwd are correct or not?
3. After successful validation , server generates a unique token
4. Server stores the token in db against the usrid

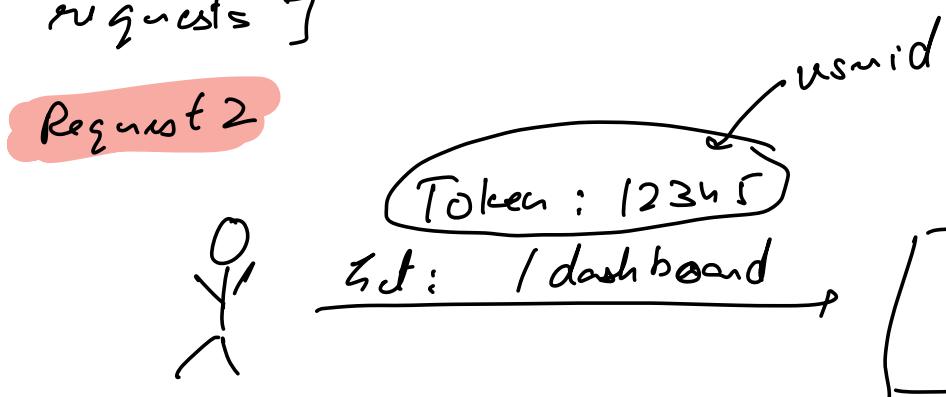




5. Server sends the token to the client



6. Client is supposed to ^{6.1} save the token & ^{6.2} [send it with all future requests]



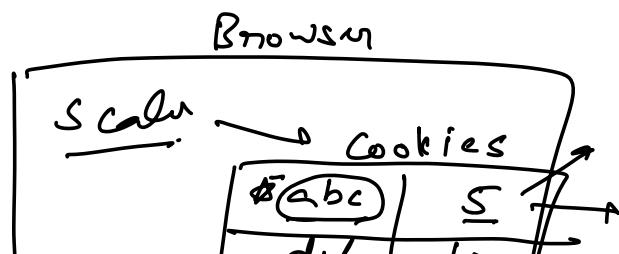
Server.

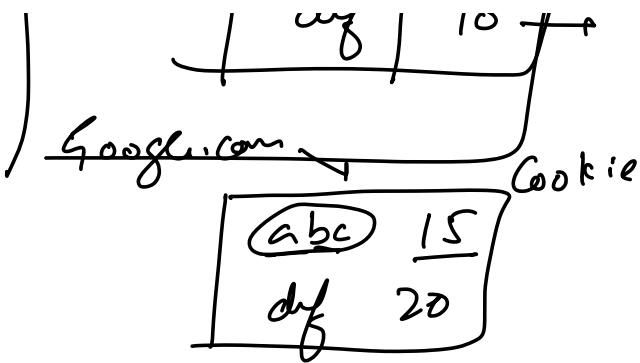
* Server will check db for the user who has this token.

- * On Backend token is stored in db / RAM]
- * On frontend side, token is stored in Browser Cookies

Cookies

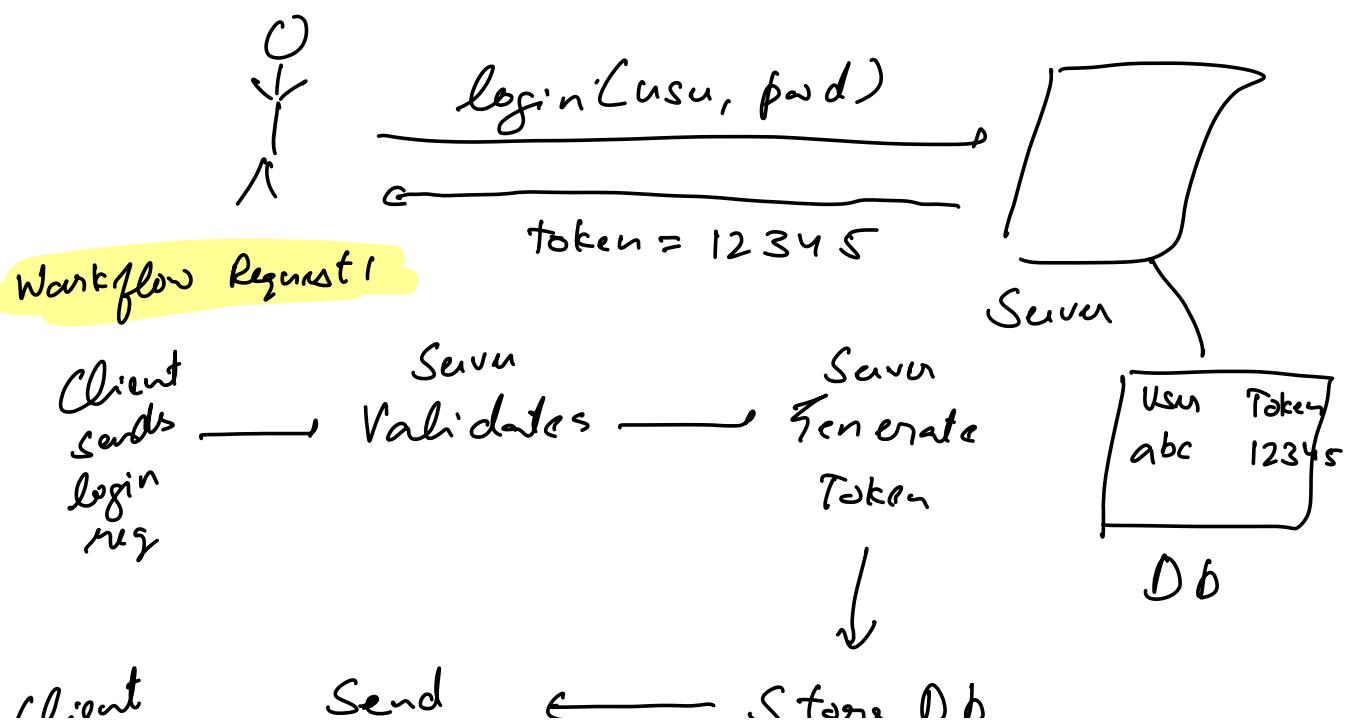
1. A way to store information on browser per domain [website]
2. Hashmap [key, value] Pair





3. Once cookie is stored in browser, for future requests it will be sent to the server automatically.

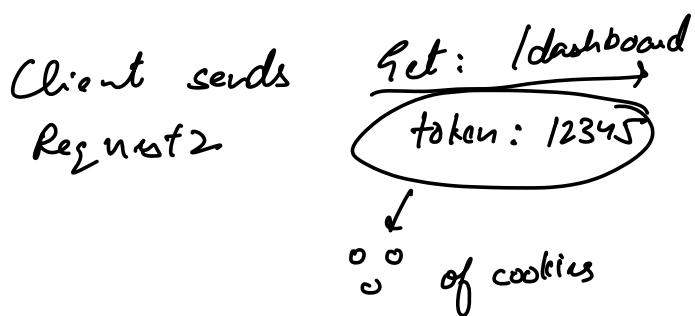
Recap → login



will store token in cookies

Response - with token to user

Workflow Request 2



* [Please hold your doubts for 10 minutes]

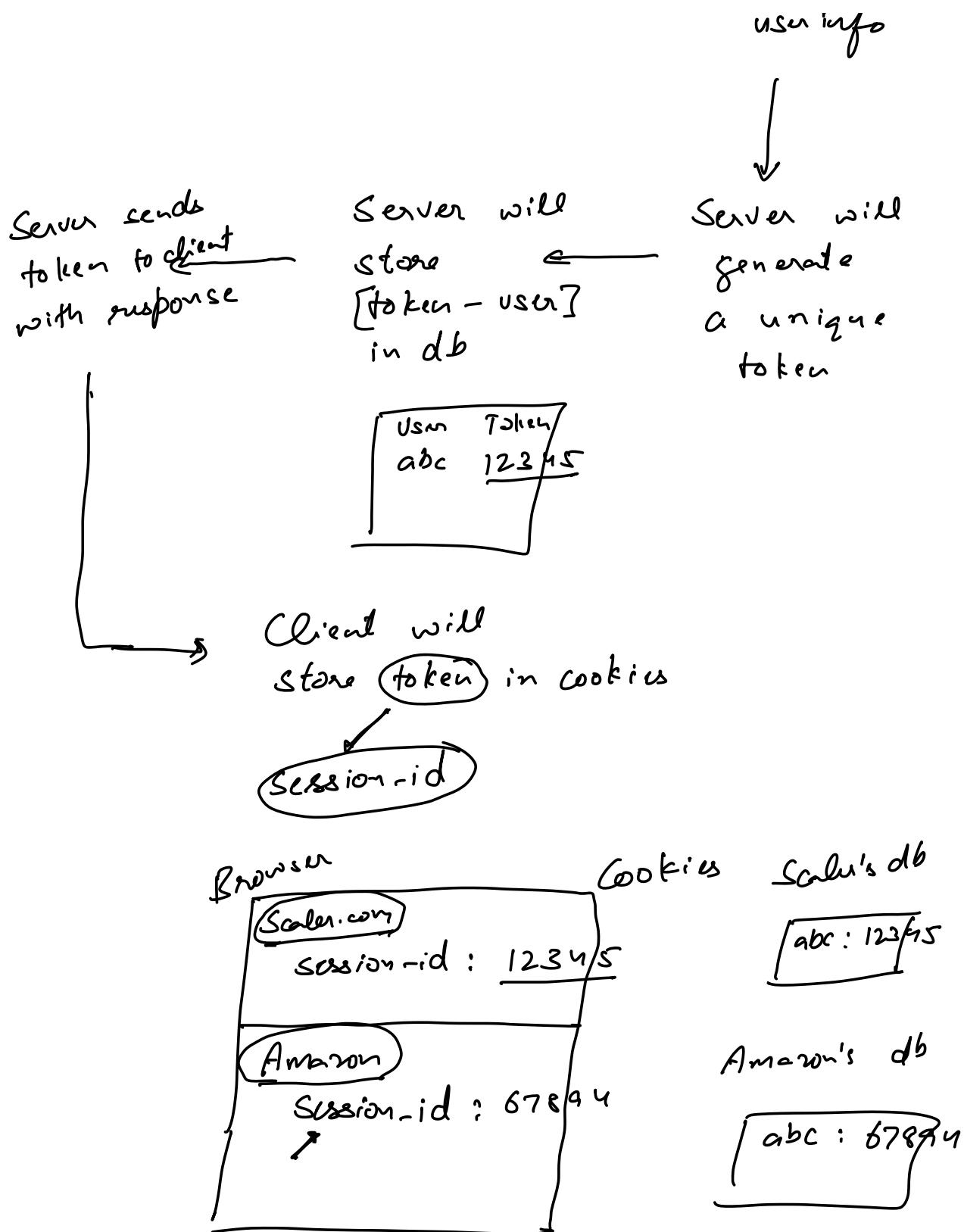
Workflow Req 1

1. User trying to access `scaler.com` for the first time.

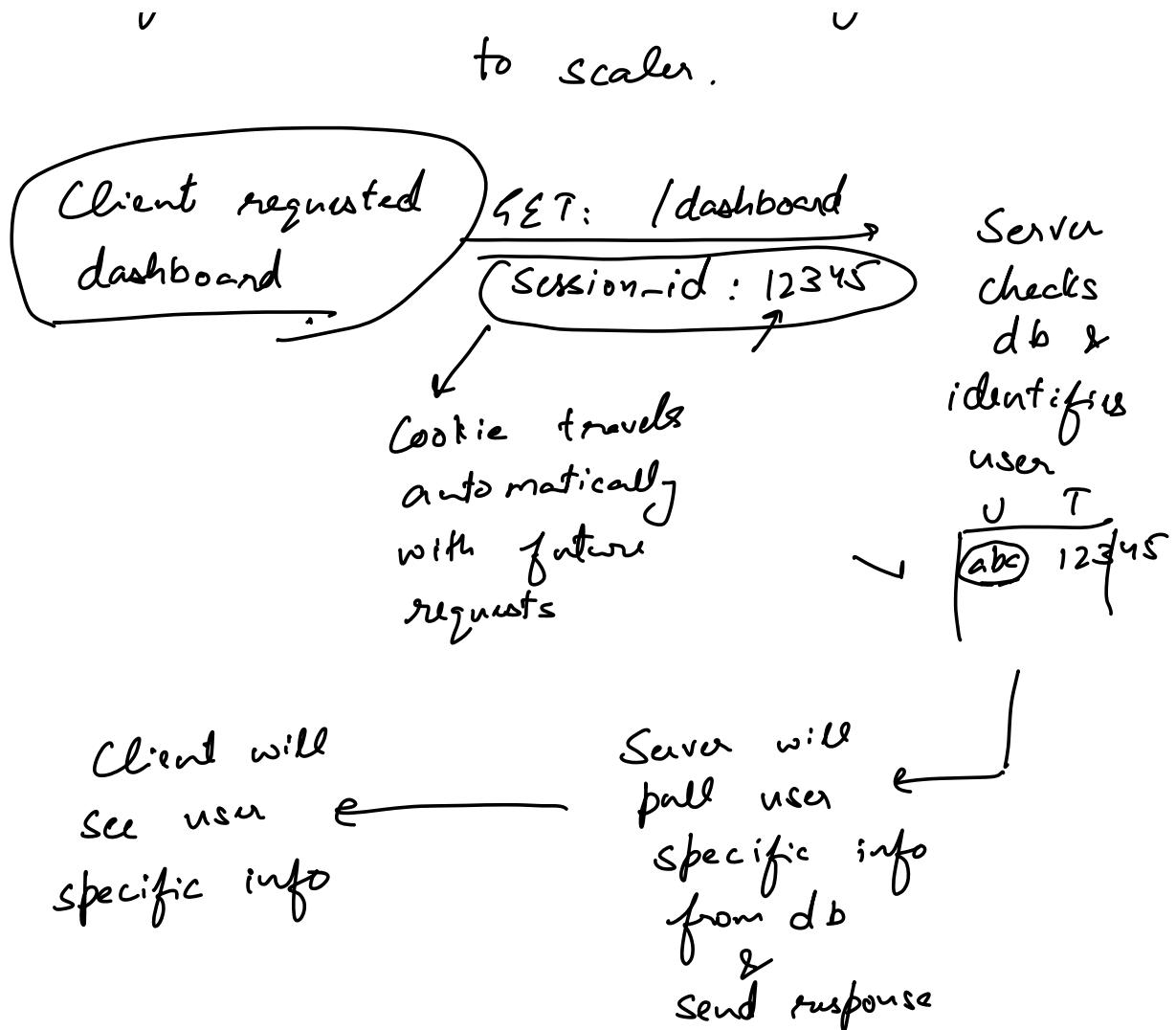
Client sends login req

login(user, pwd)

Server will validate

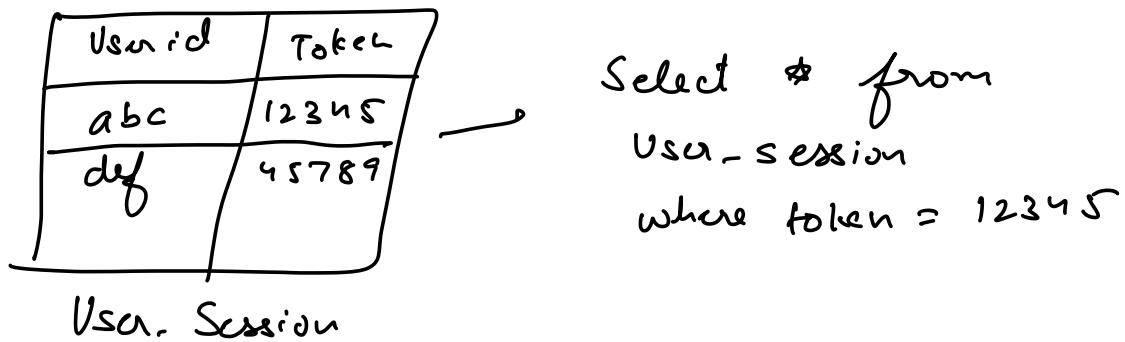
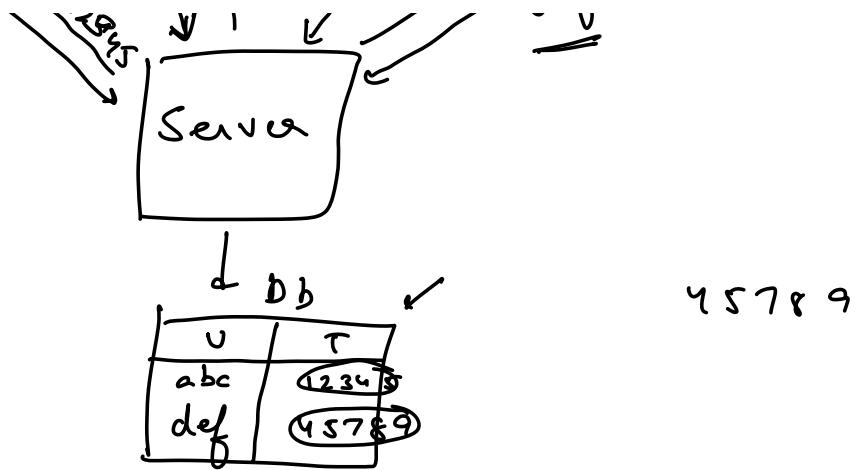


Workflow 2 → 2nd request from browser

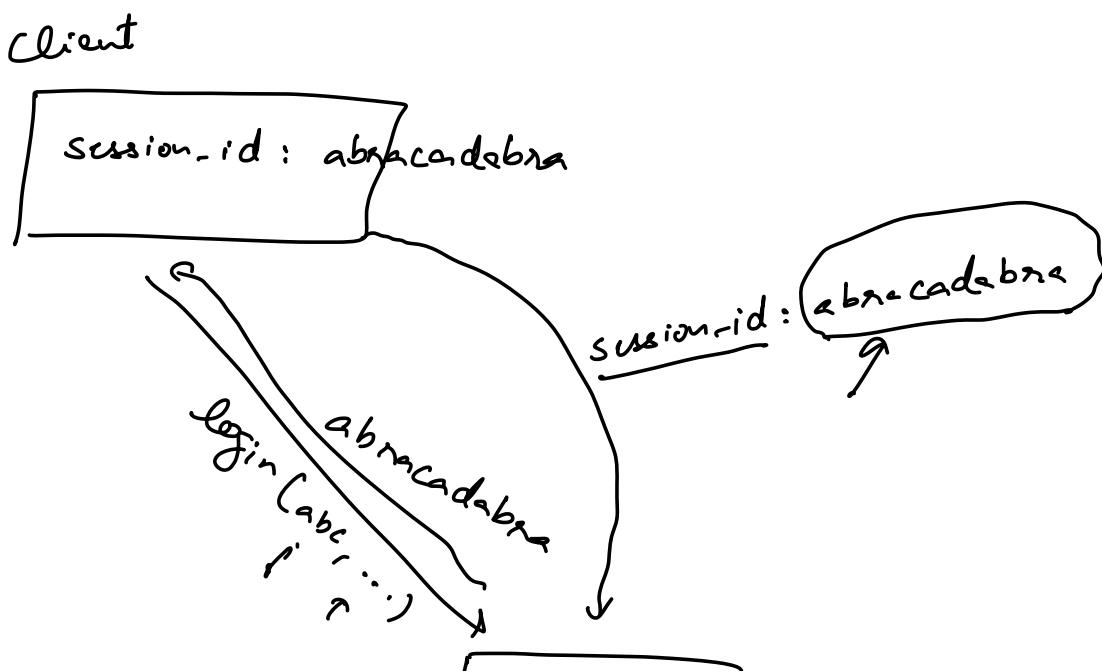


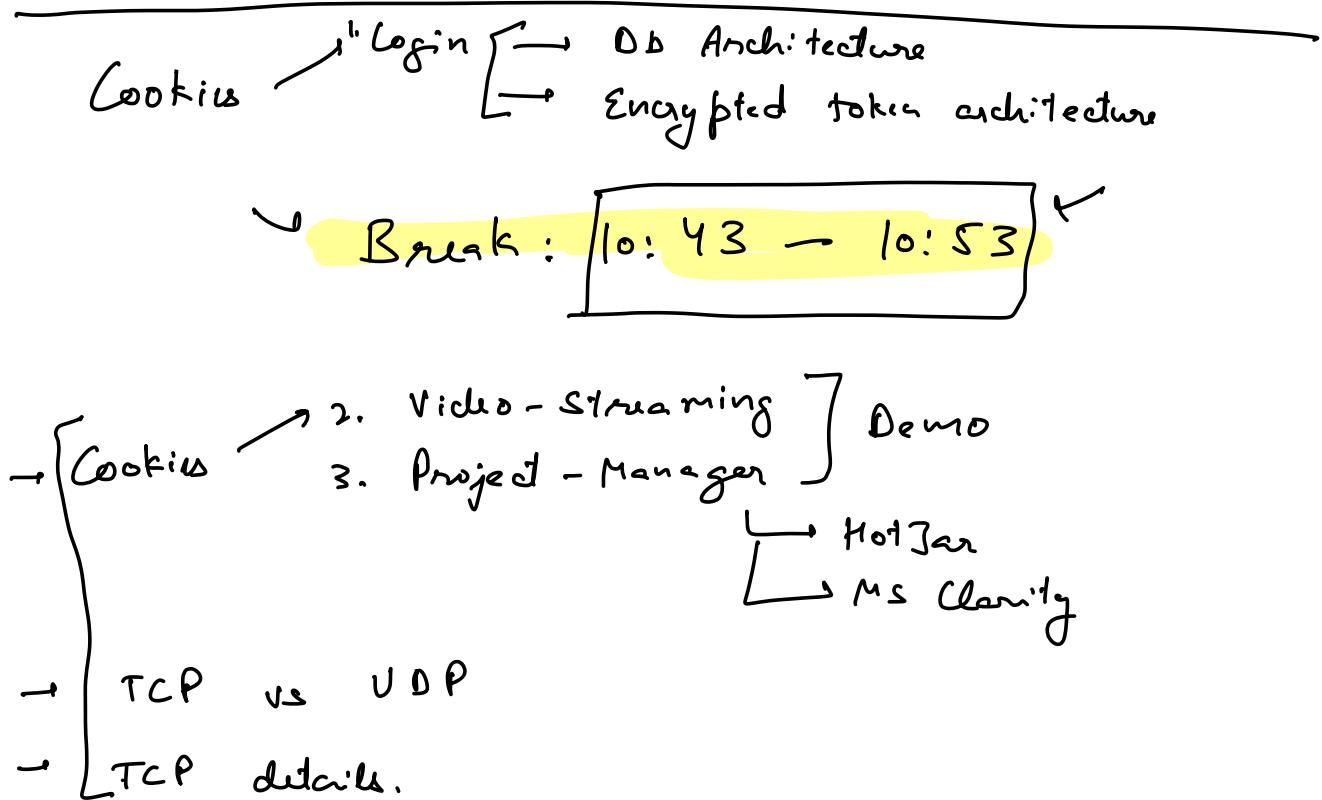
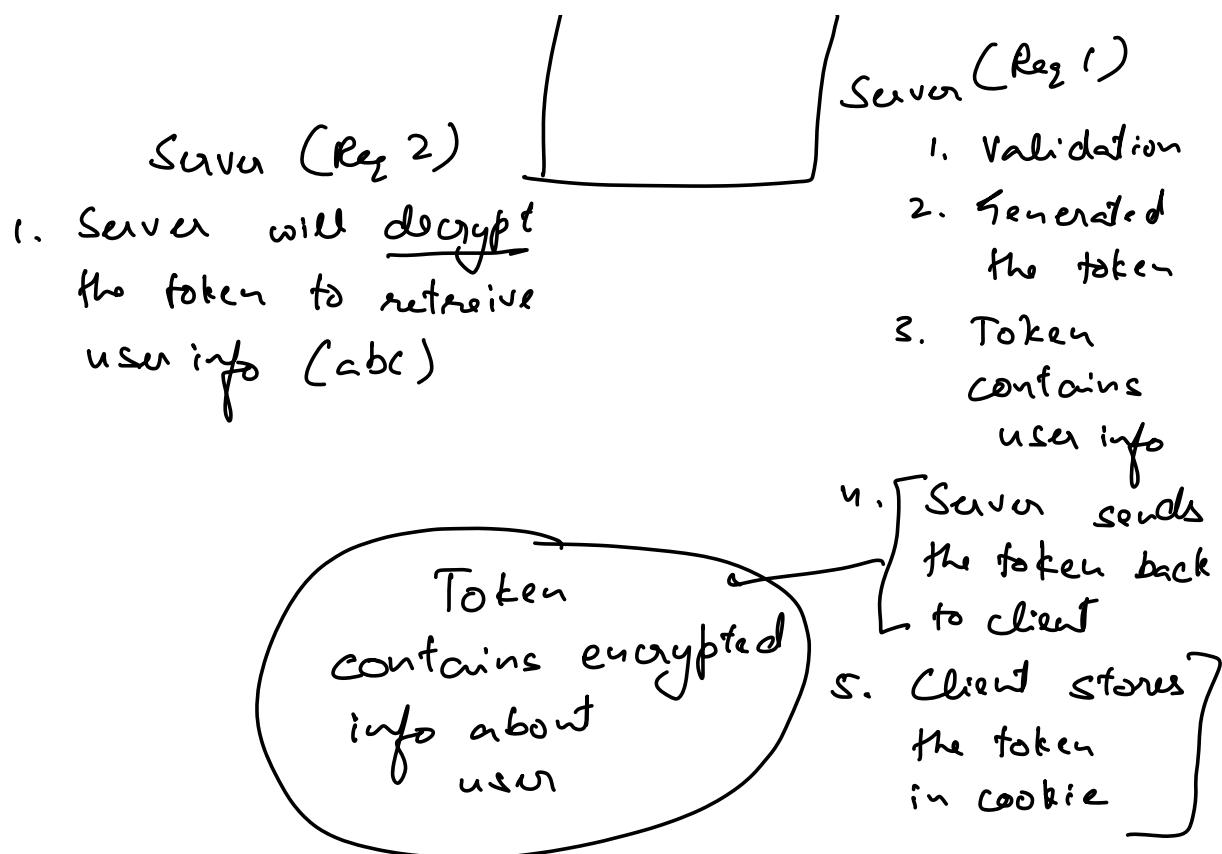
Scenario : 2 clients making requests to server





Other possible architecture

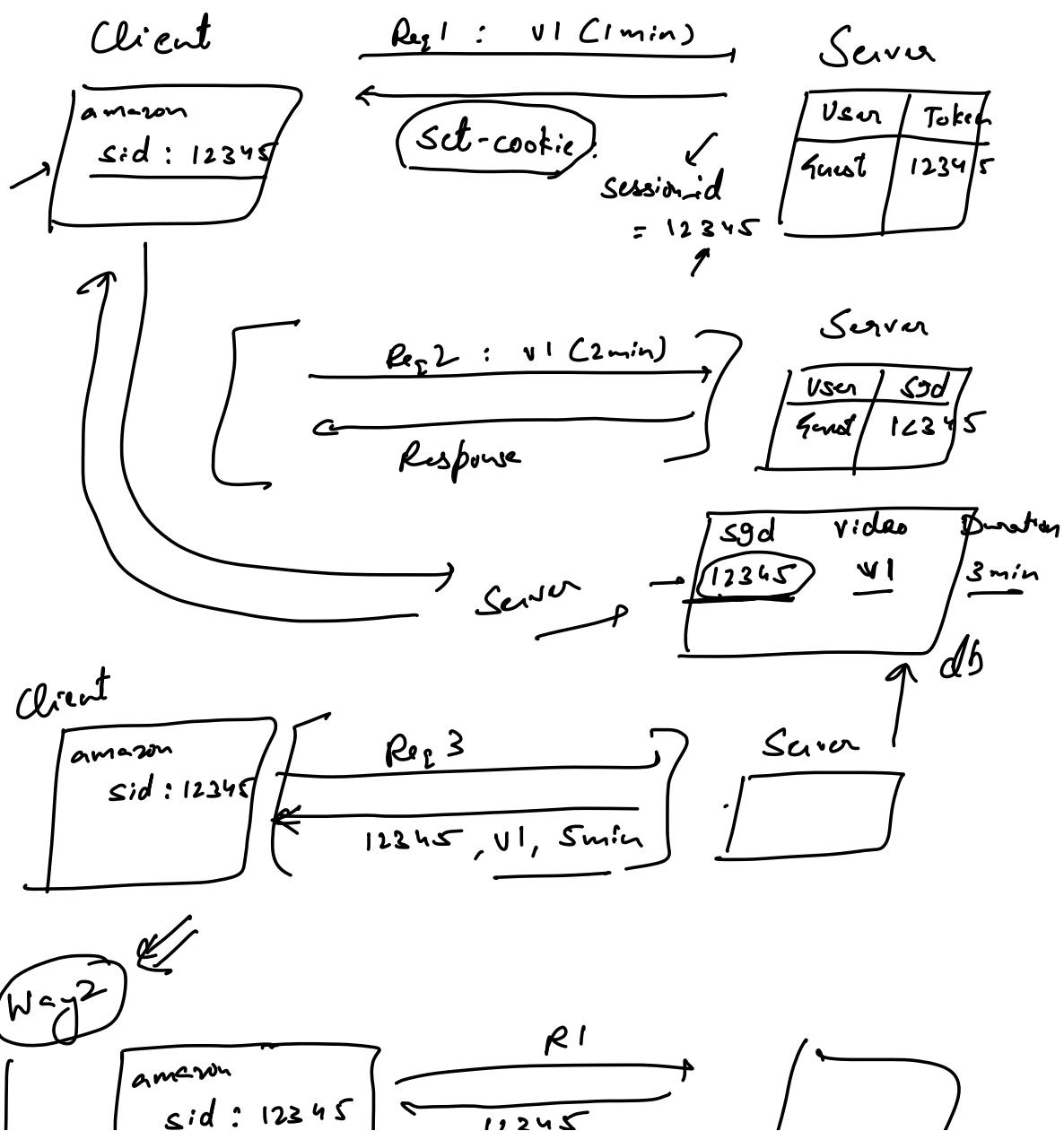


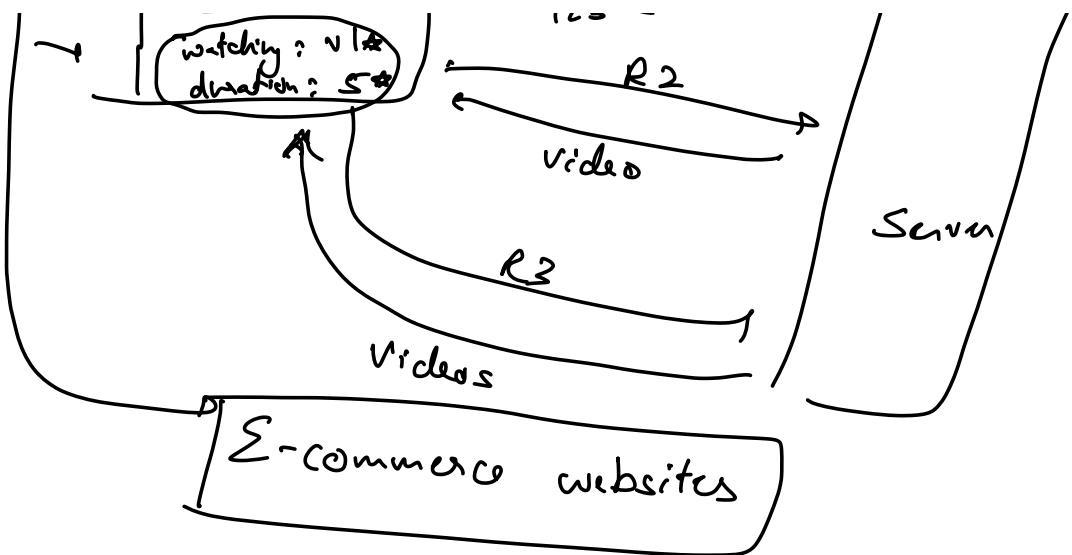


Cookie : Showing relevant info to guest users

e.g. Continue watching in video-streaming
Cart in ecommerce

Way 1. ↗ Preferred Way





watching : v1, v2, v3, v4, v5

v1-duration : 5

v2-duration : 6

Way 2 benefits over Way 1

Way 2
information only in cookie

Way 1
information for guest user in server on db

[3rd use case → How do project manager plan their pages?]

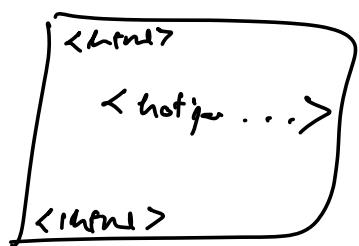
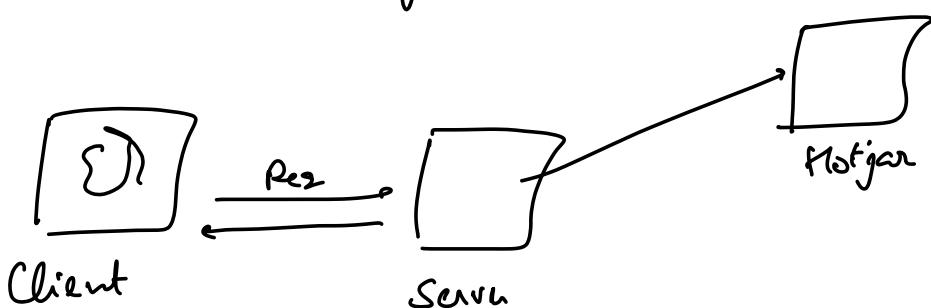
* Track user Info

{ → Mouse hovered over a particular button
→ when did you scroll? }

* How many percentage of users hover on a particular button?

* Hotjar

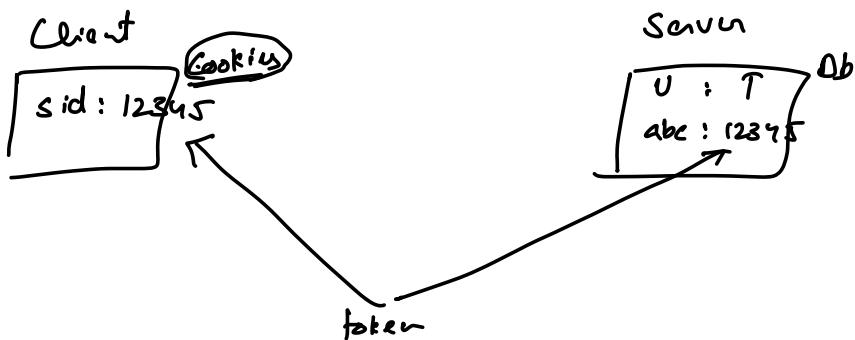
* MS Clarity



Summary

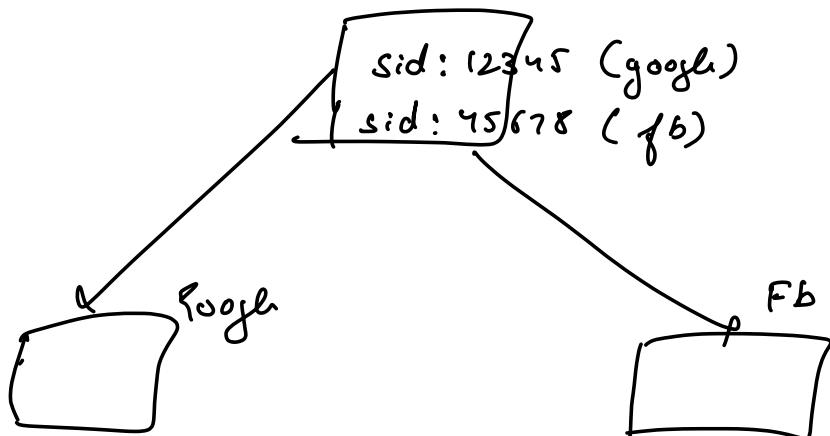
1. HTTP was stateless

2. Cookie → Client

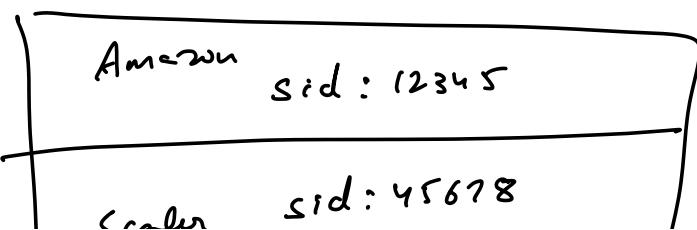


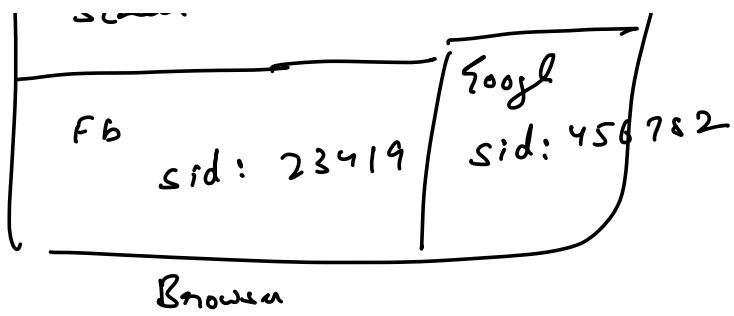
3. On the client side how

are fb cookies conflict
with google cookies prevented

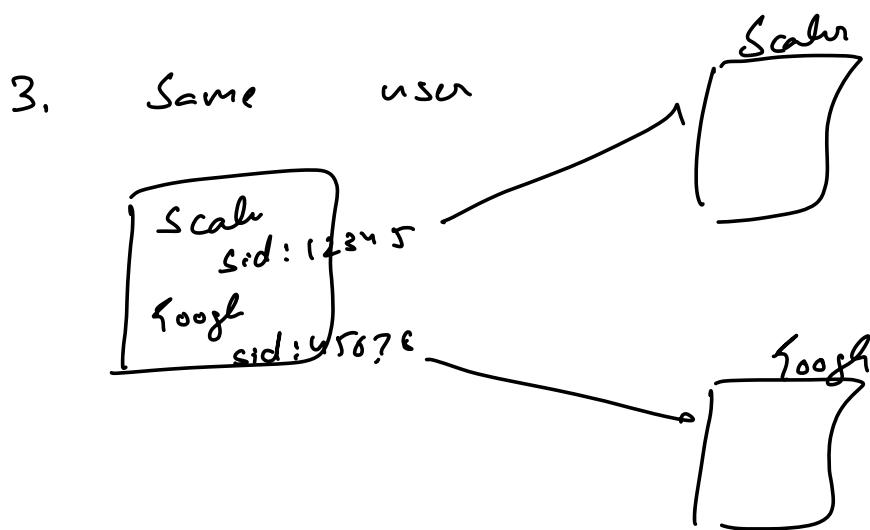
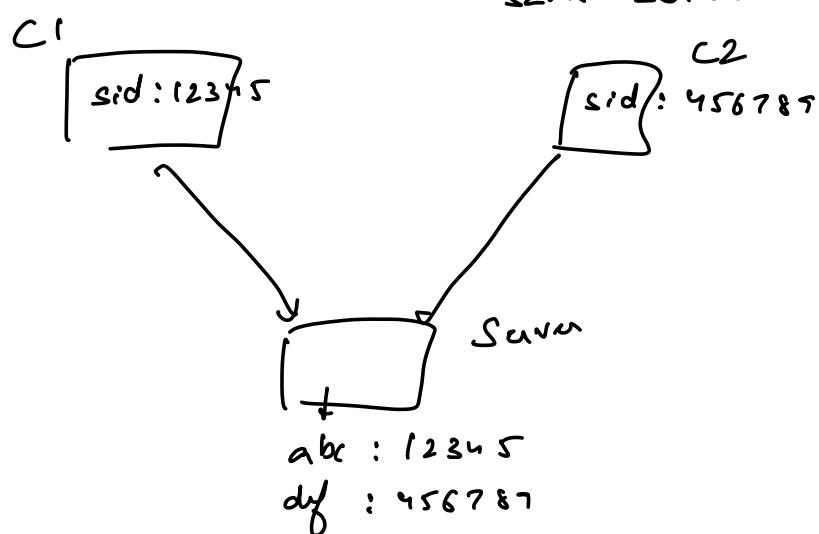


Cookies are Hashmap (key-value)
which are domain specific.

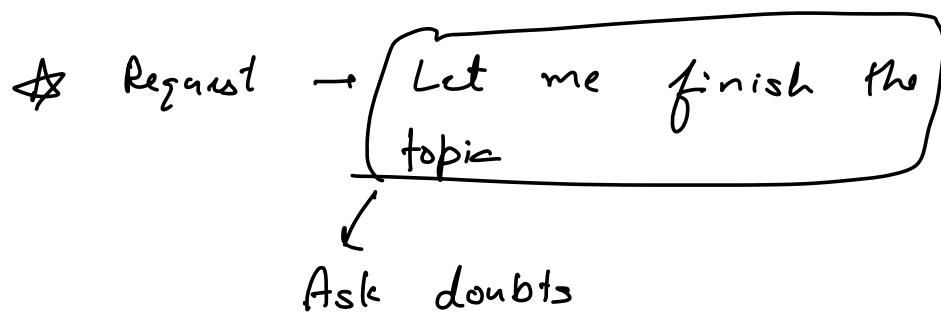
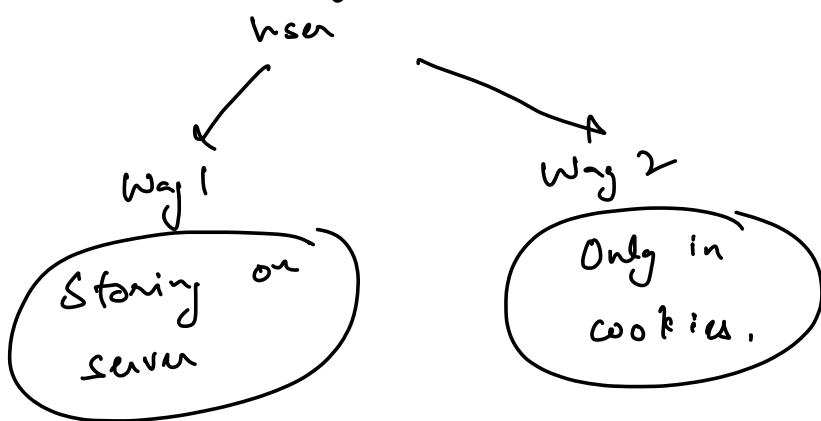




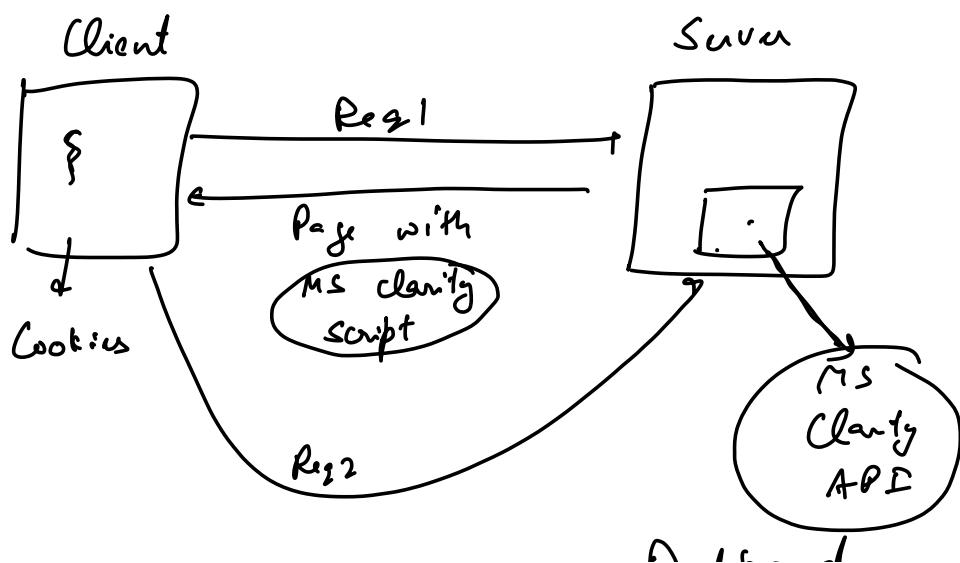
4. Workflow : Two users on
two machines
connecting to
same server



5. Configurable Page for Guest



★ Swap → RabbitMQ (P2P)
★ Mohit → Chrome Parts.



Has no owner

