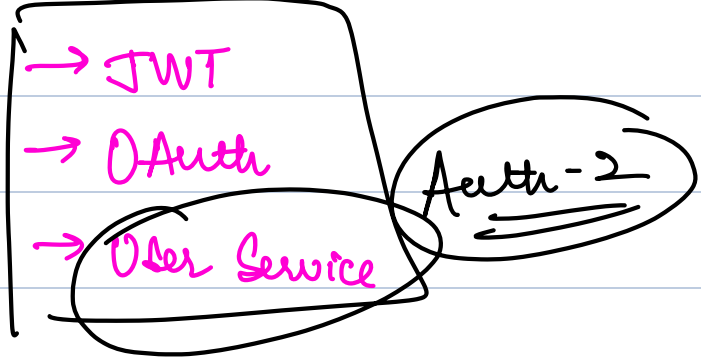
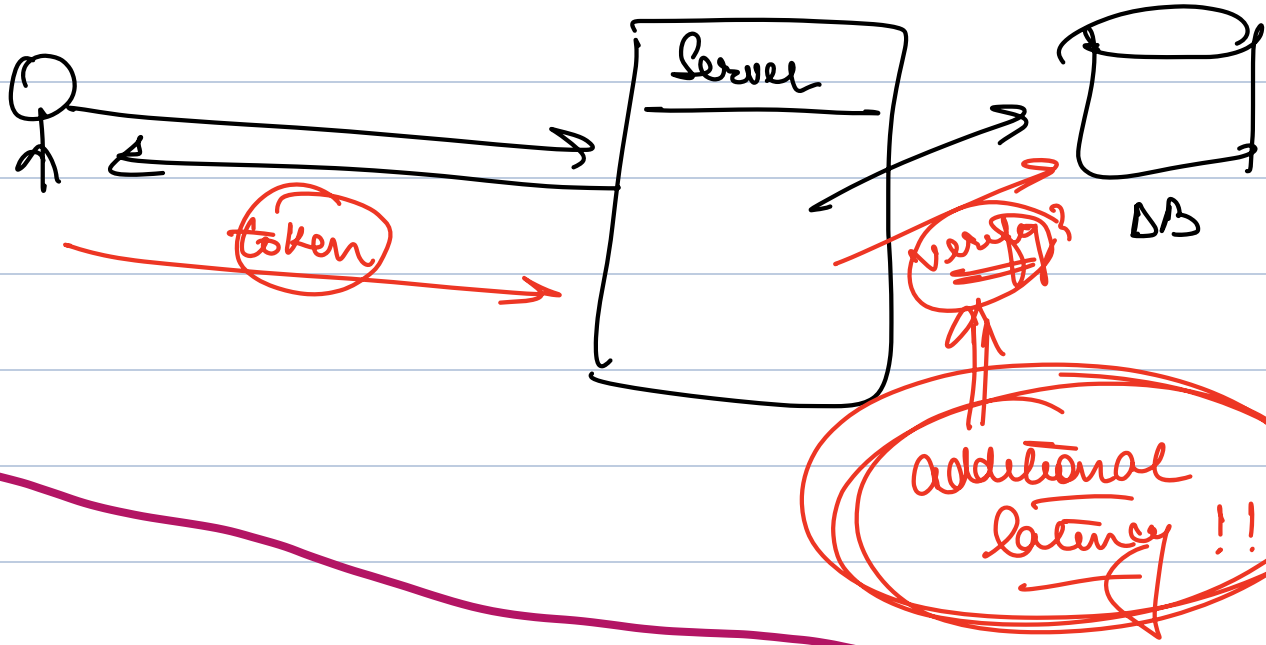
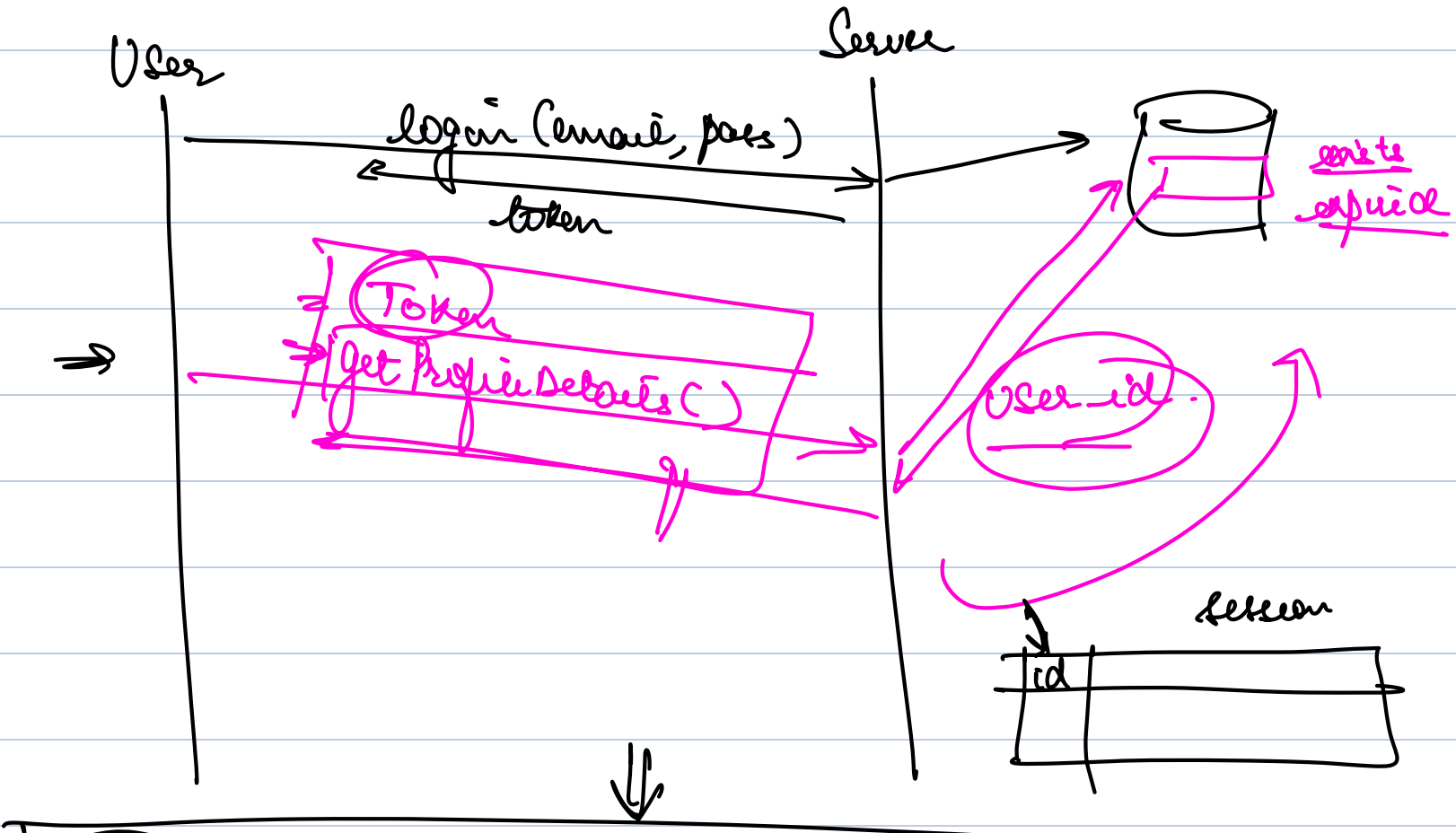




Better way to generate tokens!! Yes

JWT

JSON Web Tokens.



Token

Purpose of Token

1.) To validate the user is logged in

2.) To get the details of user.

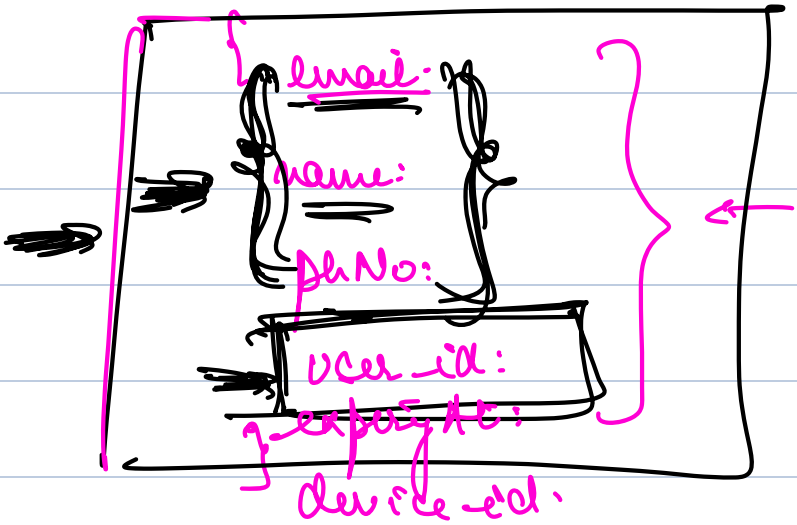
Can we do this in better way!

GET DETAILS OF USER

→ what if in token itself user details are there?

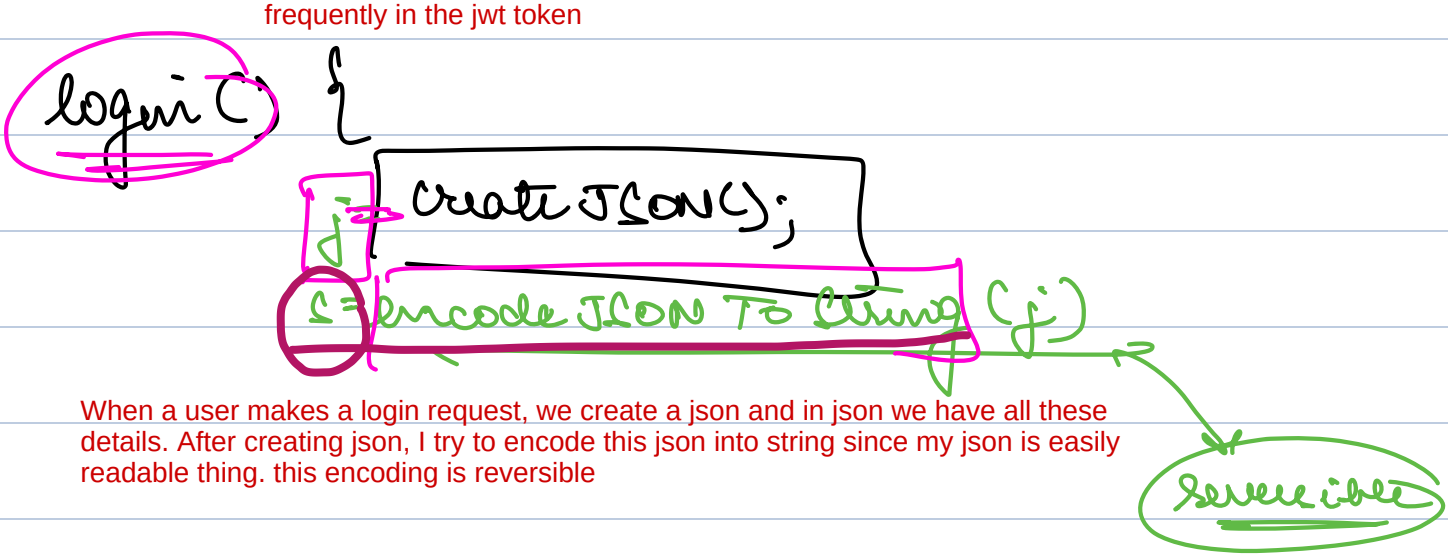
store details in token

a) create a json



In json you put all the attributes that you might want to fetch from token w/o needing to go to DB without needing to go to DB

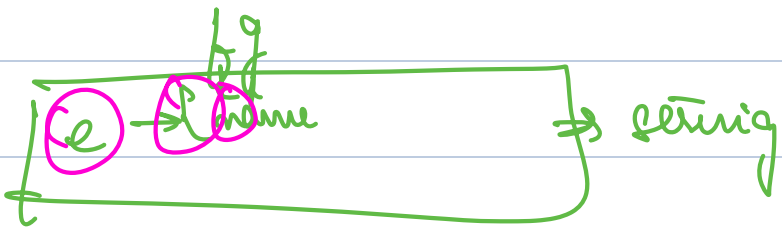
here we can get details using user_id then why you are putting details like email, name, phone #. We actually want to put common attributes which are required frequently in the jwt token



When a user makes a login request, we create a json and in json we have all these details. After creating json, I try to encode this json into string since my json is easily readable thing. this encoding is reversible

token = s

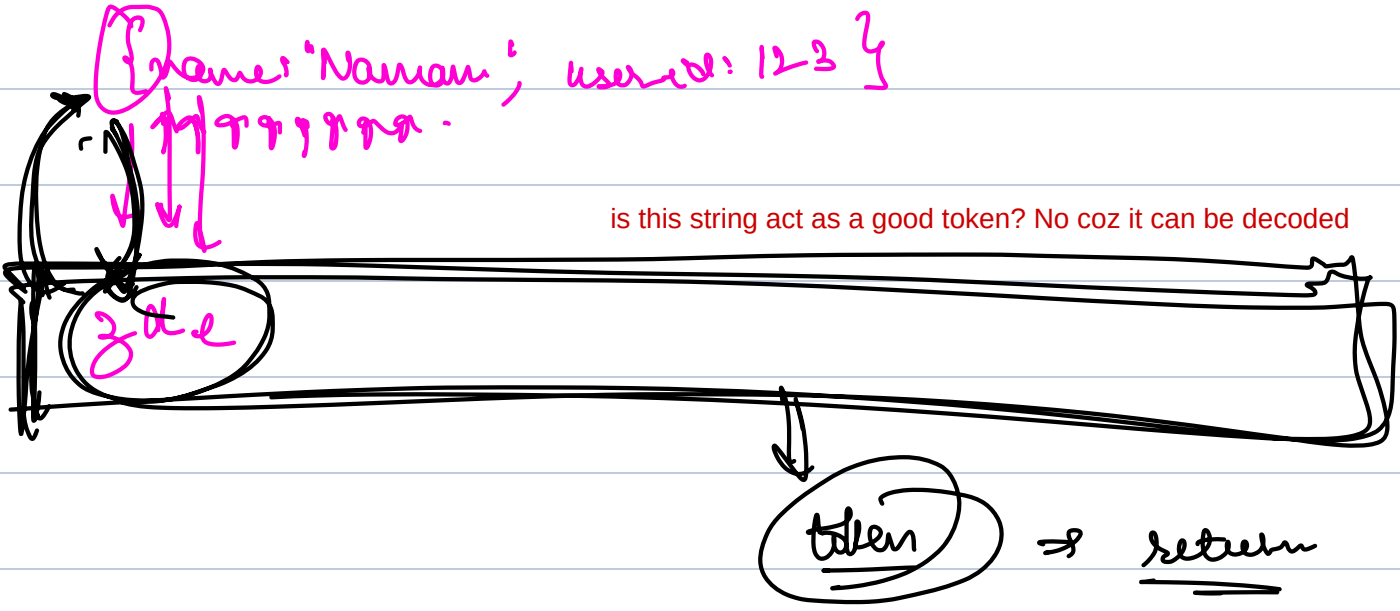
return token

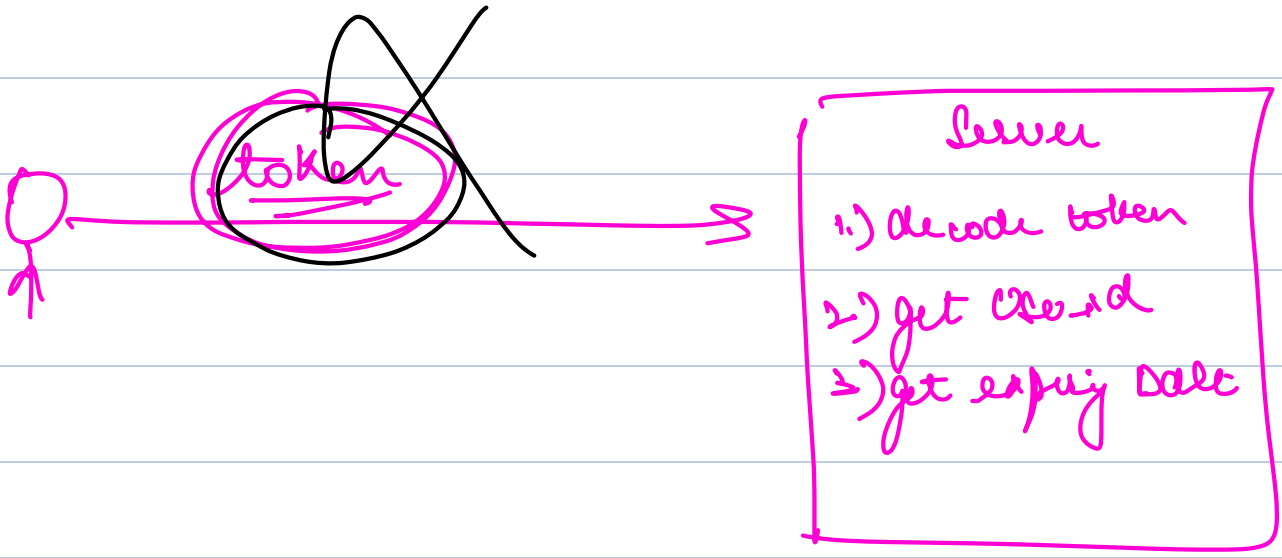


encoding character by character

Base 32

{name: 'Naman', userId: 123}

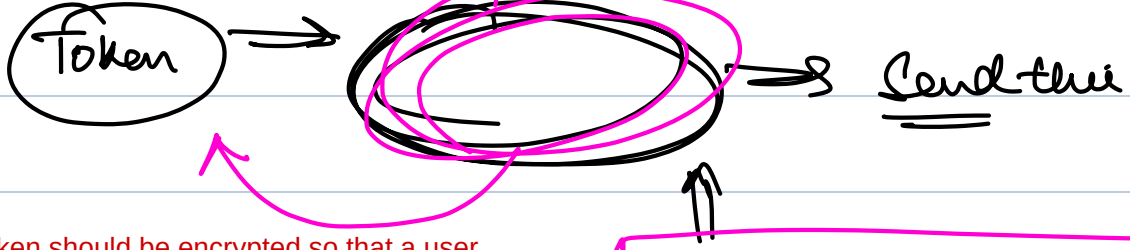




Token ✓

Safe ✗

Manipulate ✗



token should be encrypted so that a user can't do on their end.

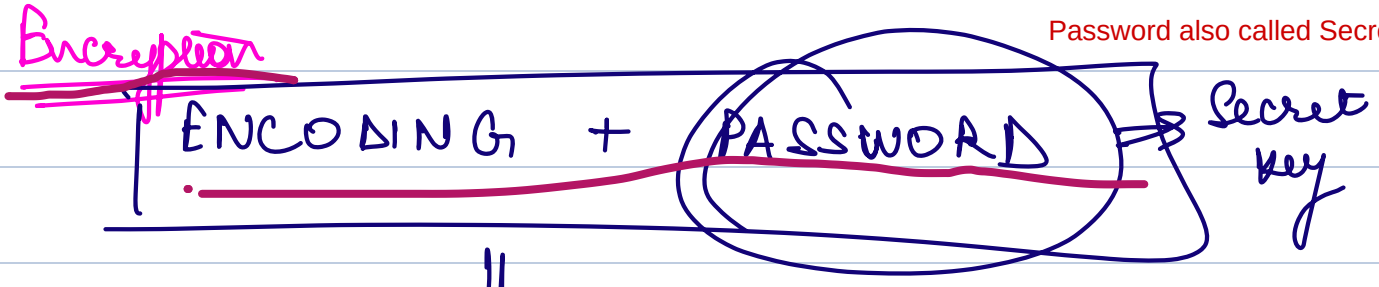
Only server can do on their end

→ Something that a user can't do on their end

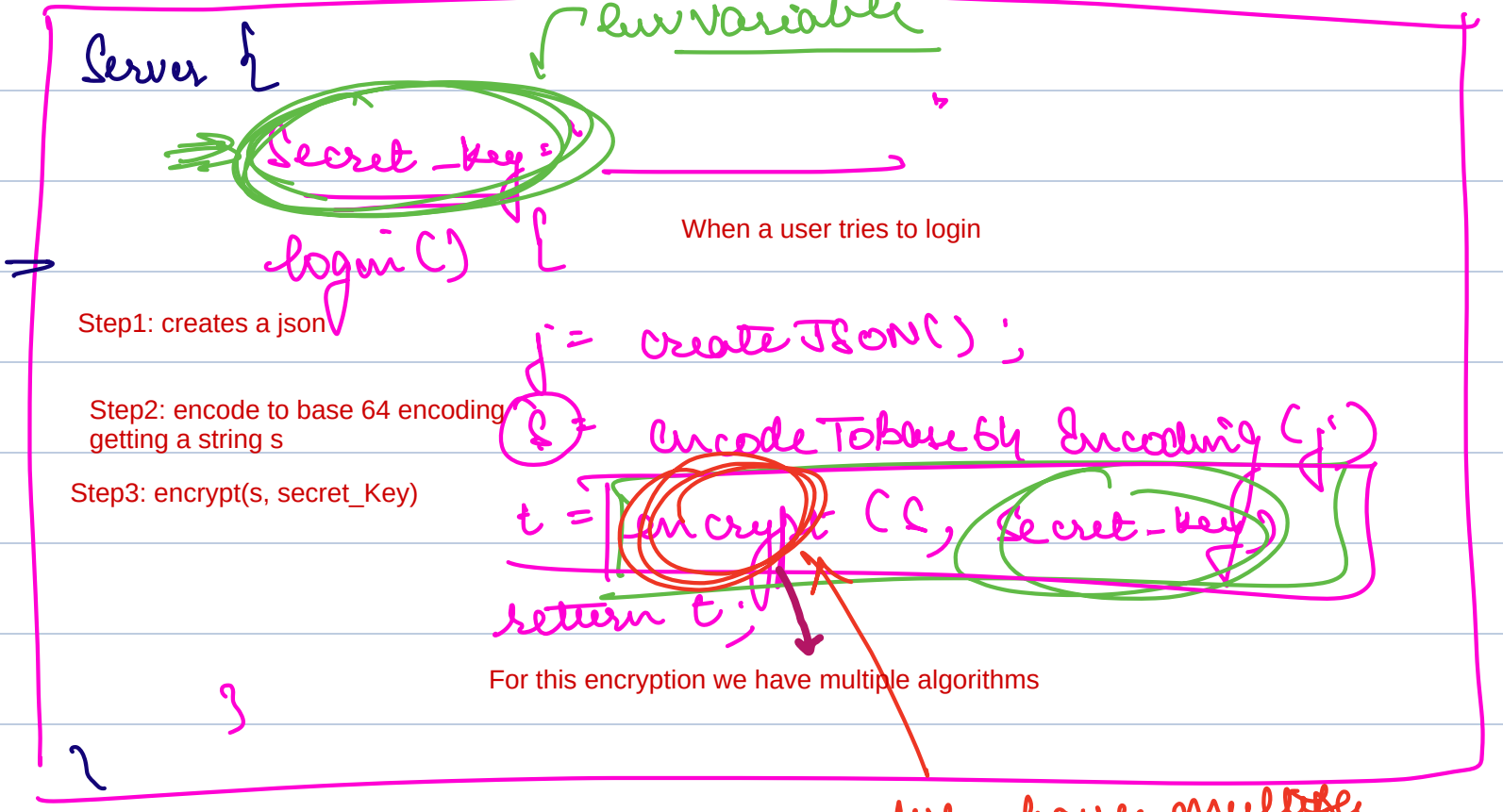
→ only a server can do at their end.

Encryption

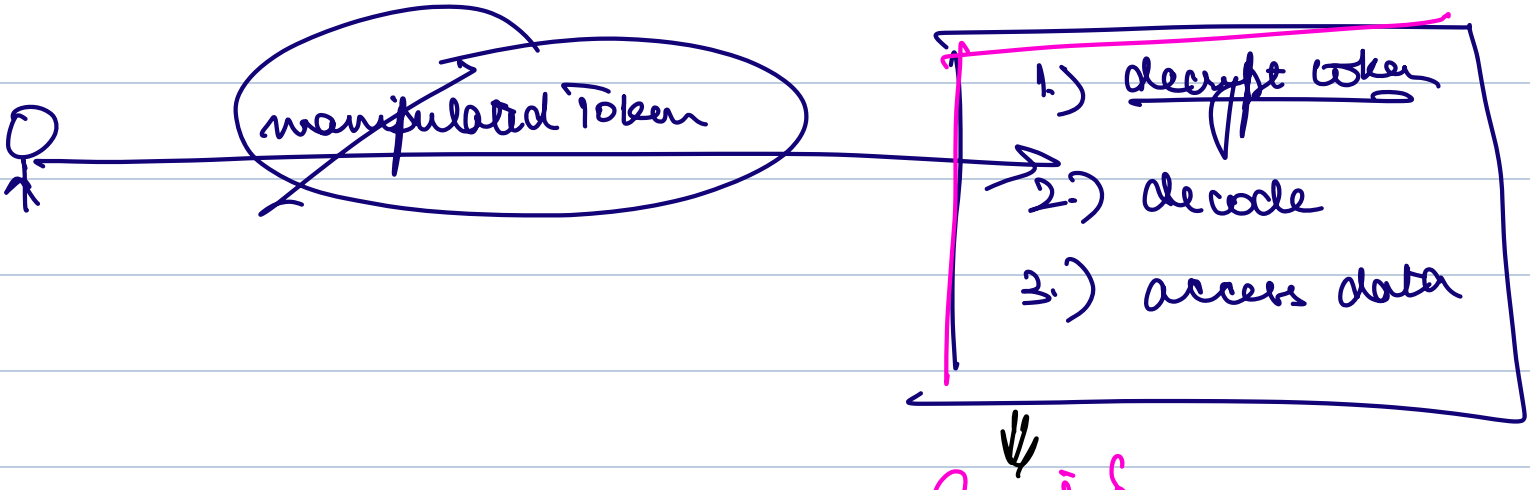
Password also called Secret key



→ you can decode BUT IFF you have the password.



we have multiple algos to encrypt



Gmail {

1.) decrypt
2.) decode
3.) email ID
4.) go to DB and fetch mails
5.) return

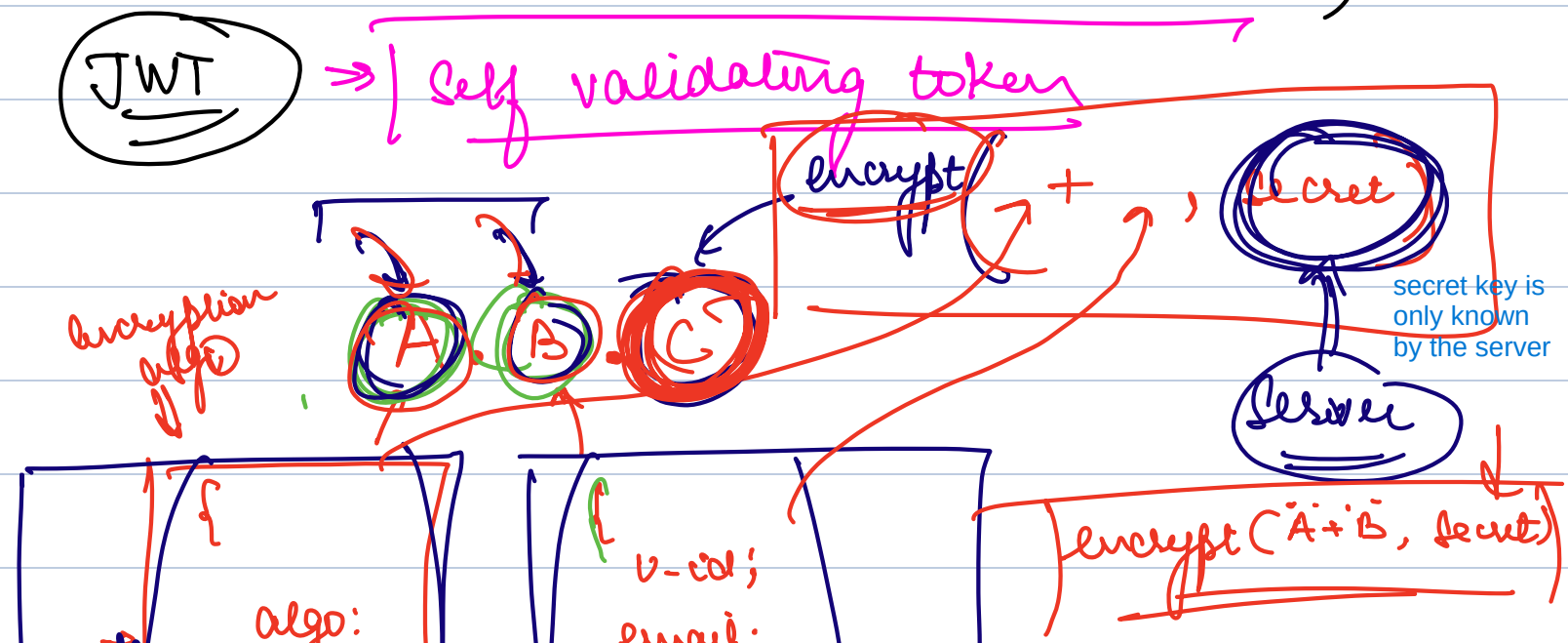
"Validating the token on server itself"

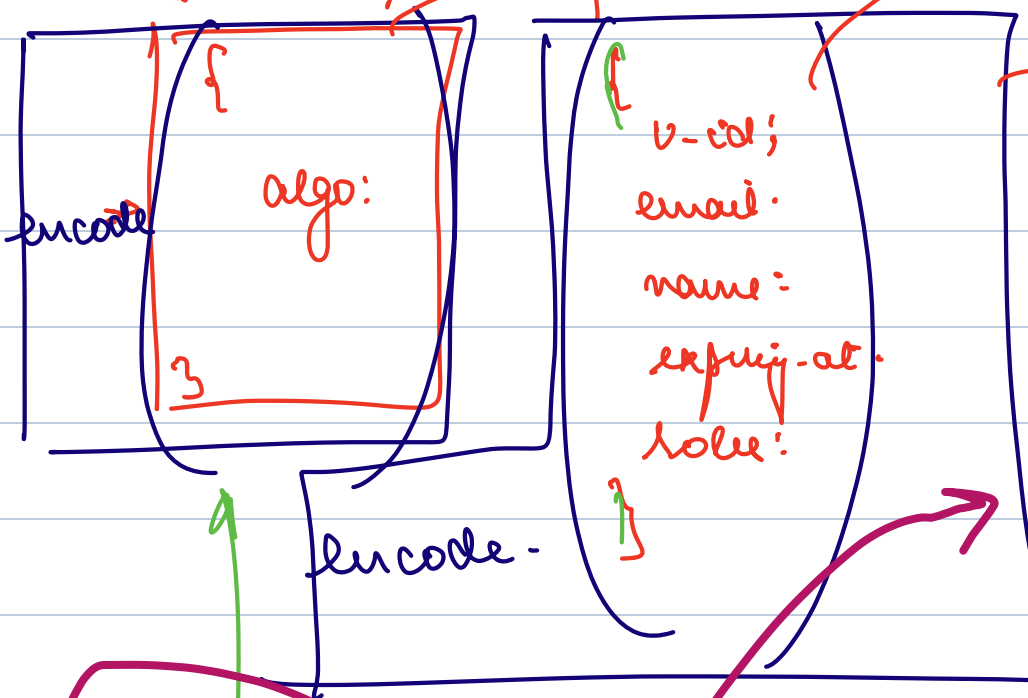
⇒ reduce latency

⇒ make req that require auth fast.

make request that require authentication fast

JWT ⇒ Self validating token





JWT tokens are divided into 3 parts separated by a dot A.B.C. First part (A) has information about the algorithm, which encryption algorithm we will use -> not the json but the encoding of the json

2nd part would be the user information -> not the json but the encoding of the json

So in part A & B we donot want to put any confidential information user related since it is not encrypted but the user can maipulate it? Yes. So part C is the Encryption of Part A + Part B with a secret key

→ Only encoding

→ no encryption

→ user can decode at

⇒ even if user manipulate part A and B, they can't gen c

their end.

→ Don't put any confidential info there

as it requires Secret Key

Even if user can manipulate, Part A and Part B, try can't manipulate Part C as it requires secret key. Thats why JWT are secure

```
login() {
    j1 = createJSONOfAlgoC();
    S1 = encode(J1)
    j2 = createJSONOfUser();
    S2 = encode(J2)
    S3 = encrypt(S1 + S2, secret)
    return S1 + "." + S2 + "." + S3
}
```

Part A

String 1 S1

Encode Part A

Part B

Encode Part B

Part C = Encrypt S1 + S2, secret key

JWT looks like

Token

(S1.S2.S3)

```
→ fetchEmails(token) {
    S4 = decrypt(S3, secret)
    if (unable to decrypt) OR {
        (S4 != S1 + S2)
        Manipulation !! → reject
    } else {
        j2 = decode(S2)
        u_id = j2.u_id
    }
}
```

I will first fetch the mails.
1st decrypt(S3, secretkey) and store in S4

If unable to decrypt, or S4 is not equal to S1 + s2. then user has manipulated -> reject the request

else if I am able to decrypt then decode s2 and by decoding s2 I would have my user_information user_id.

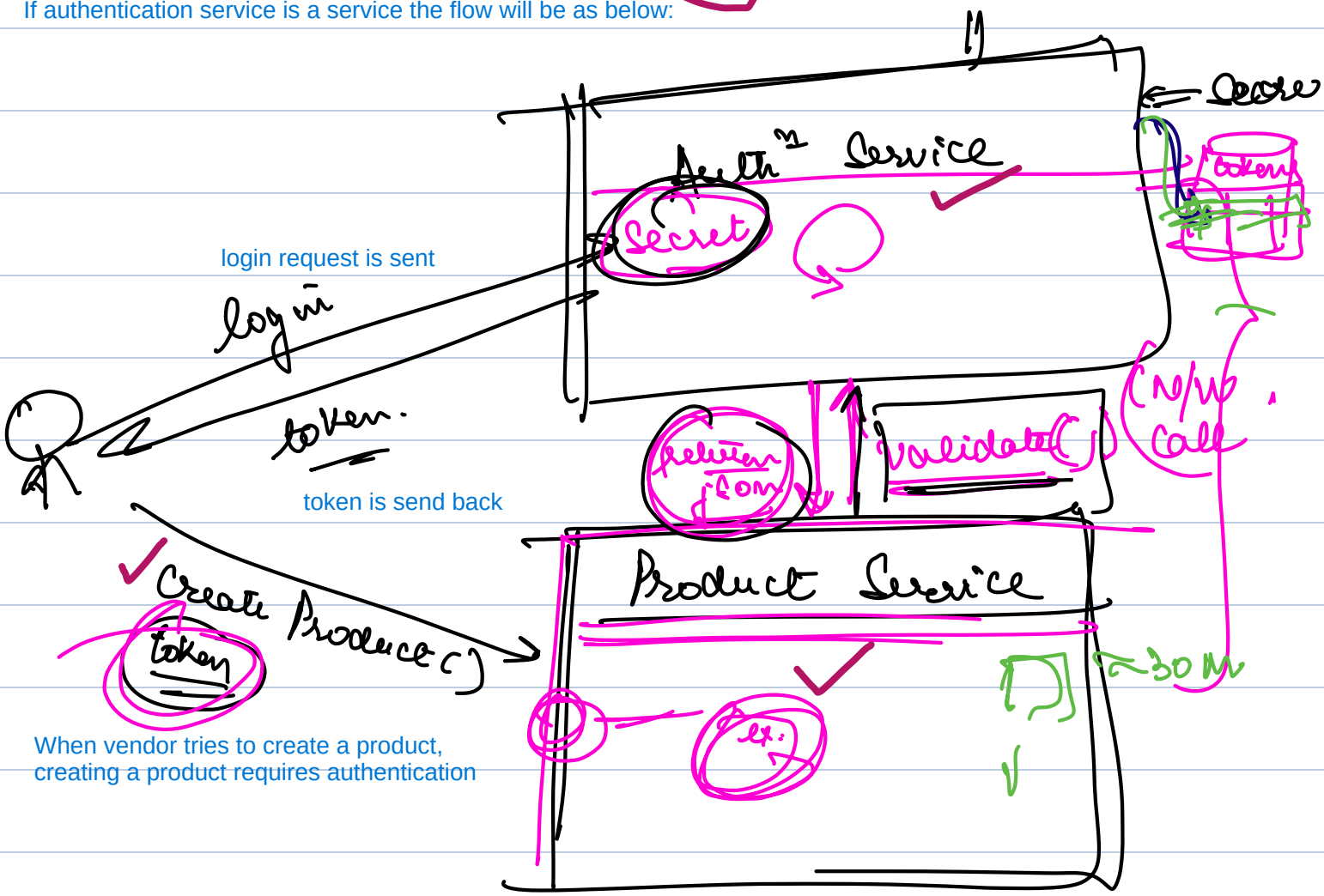
Whenever a company tries to move from Monolithic to Microservices in 99% cases first authentication part moves in

Service By Service

⇒ pick up part that is most commonly accessed by others into an independent codebase.

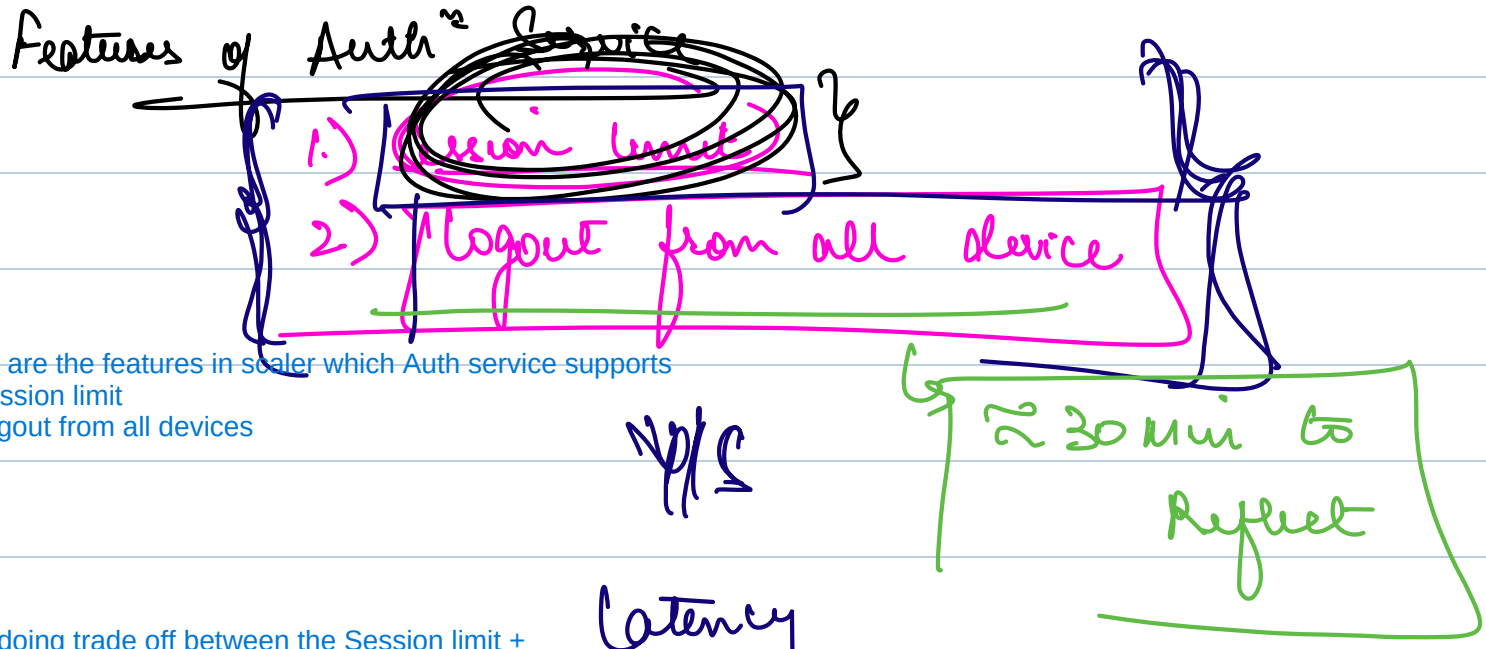
99.9% ⇒ Auth module

If authentication service is a service the flow will be as below:



When vendor tries to create a product, creating a product requires authentication

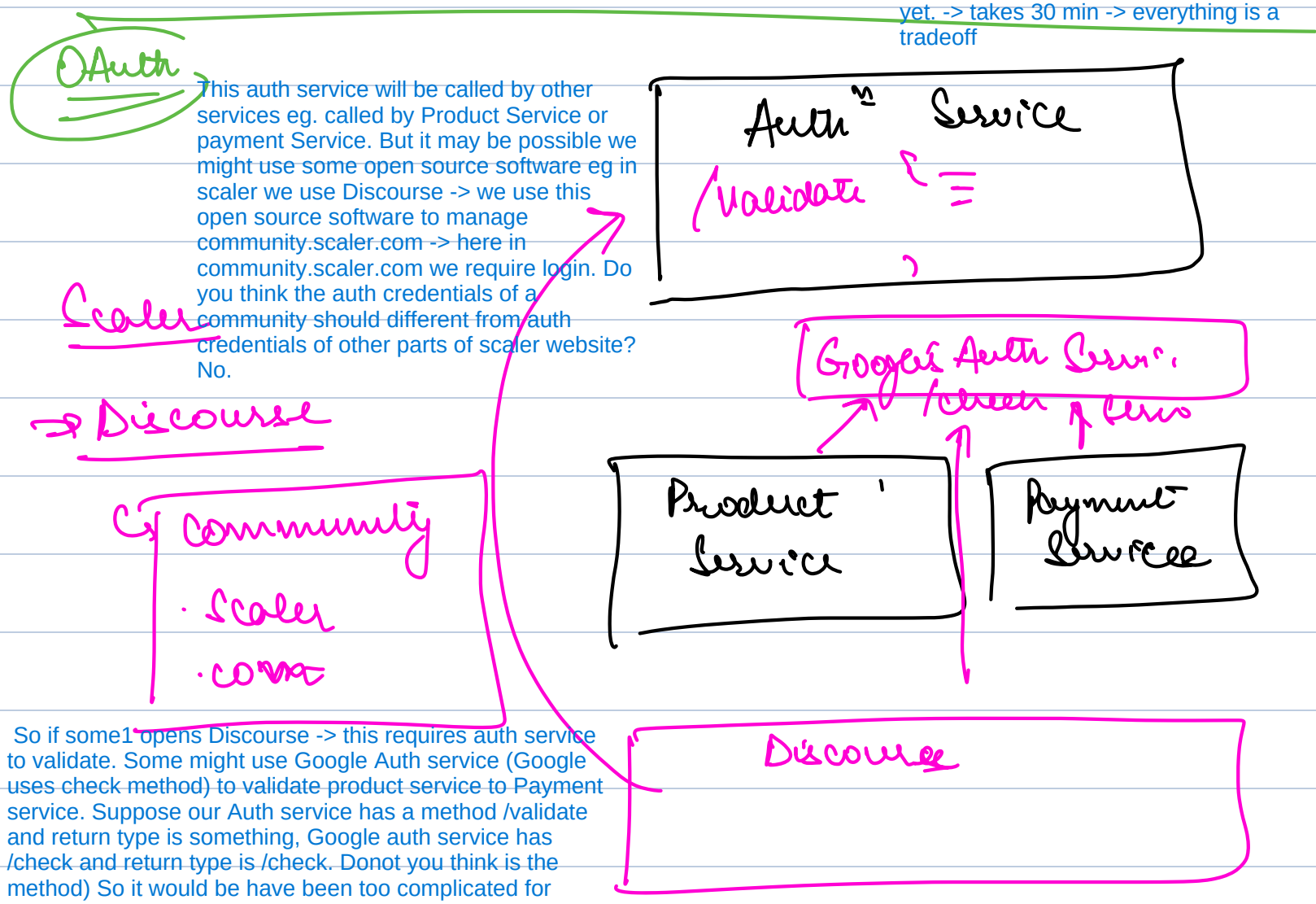
For the Product Service to decide if they allow this product to be created or not what should the product service also know?
Secret key
Now secret is known by Auth service as well as Product Service. Is this is a good idea that secret key is present in almost every service No --> so we keep secret with authentication service not with product service. Product service reaches out to Auth service to validate the token and then returns the json with user info. Netwrok call will happen -> latency but it is a



What are the features in scaler which Auth service supports
1. Session limit
2. Logout from all devices

We are doing trade off between the Session limit + Logout form all the devices Vs Latency

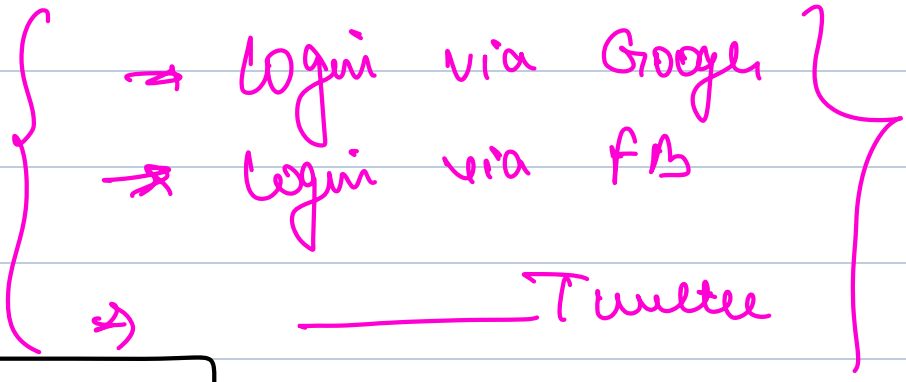
In Open AI, its takes 30 min to reflect to logout from all devices coz opnai doesnot go to the db on every request so what openAi does. OpenAI caches if this token is valid or not on server itself. On the server itself it has said it has this token is valid, in the db the token would be removed but in the cache might not have nbeen updated yet. -> takes 30 min -> everything is a tradeoff

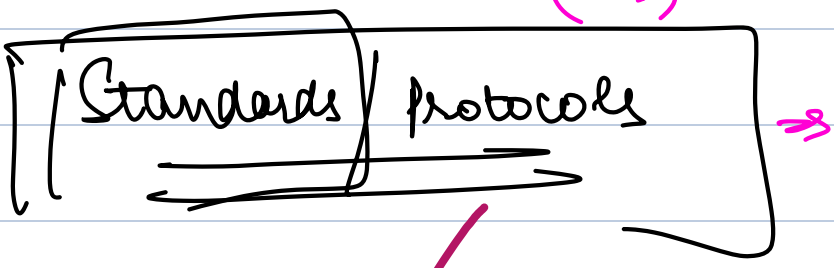


This auth service will be called by other services eg. called by Product Service or payment Service. But it may be possible we might use some open source software eg in scaler we use Discourse -> we use this open source software to manage community.scaler.com -> here in community.scaler.com we require login. Do you think the auth credentials of a community should different from auth credentials of other parts of scaler website? No.

So if some1 opens Discourse -> this requires auth service to validate. Some might use Google Auth service (Google uses check method) to validate product service to Payment service. Suppose our Auth service has a method /validate and return type is something, Google auth service has /check and return type is /check. Donot you think is the method) So it would be have been too complicated for different services to work together. It would have led to complicated codebases

Whenever such things happen we use Standard / protocols is used.

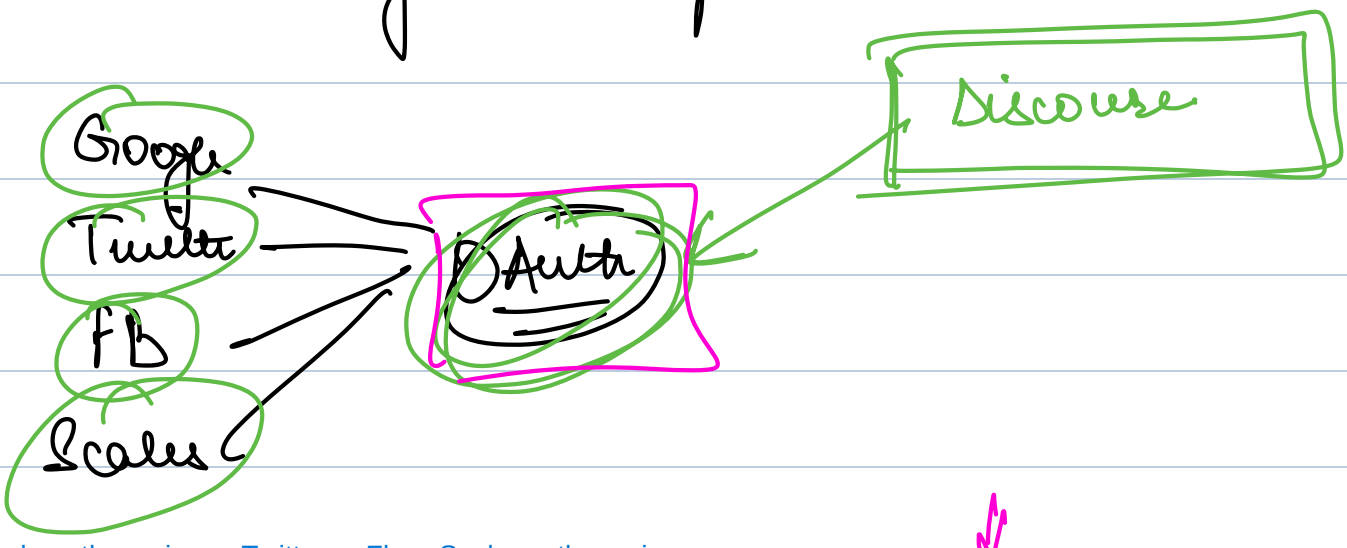




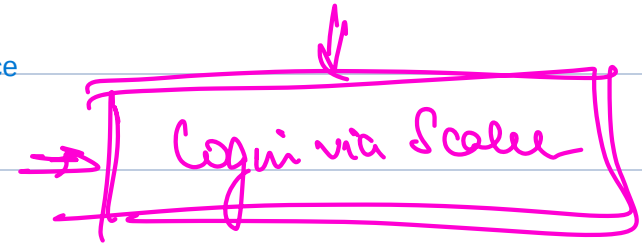
This creates a standard API that can be implemented by any authetication provider

⇒ OAuth Standard ⇒ Managing authentication and authorization

⇒ Creates a standard API that can be implemented by any auth provides.



So if Google auth service or Twitter or Fb or Scaler auth service is using OAuth standard so lets say we have Discourse to manage authentication, so it only has to provider support for OAuth providers.



How OAuth Works

4 Participants

- User
- Application
- Resource Server
- Authorization Server

Resource Server is the owner of the information that user wants to fetch. So in our case is Gmail server

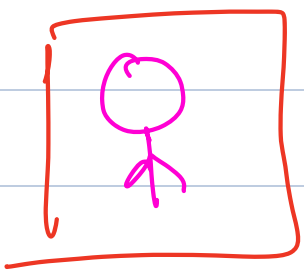
Auth service manages access to resource ion resource server.eg in the case whenever you want to give access to gmail or calendar server you have to given access, there is page where you say allow

Resource Server → the owner of the info that user want to fetch

GMail Server

Authorization Server → Server that manages access to resource on resource ser

Google's Auth Server



person will always interact with the application

Application

Outlook

Our application is Outlook

→ the person who wants to take an action

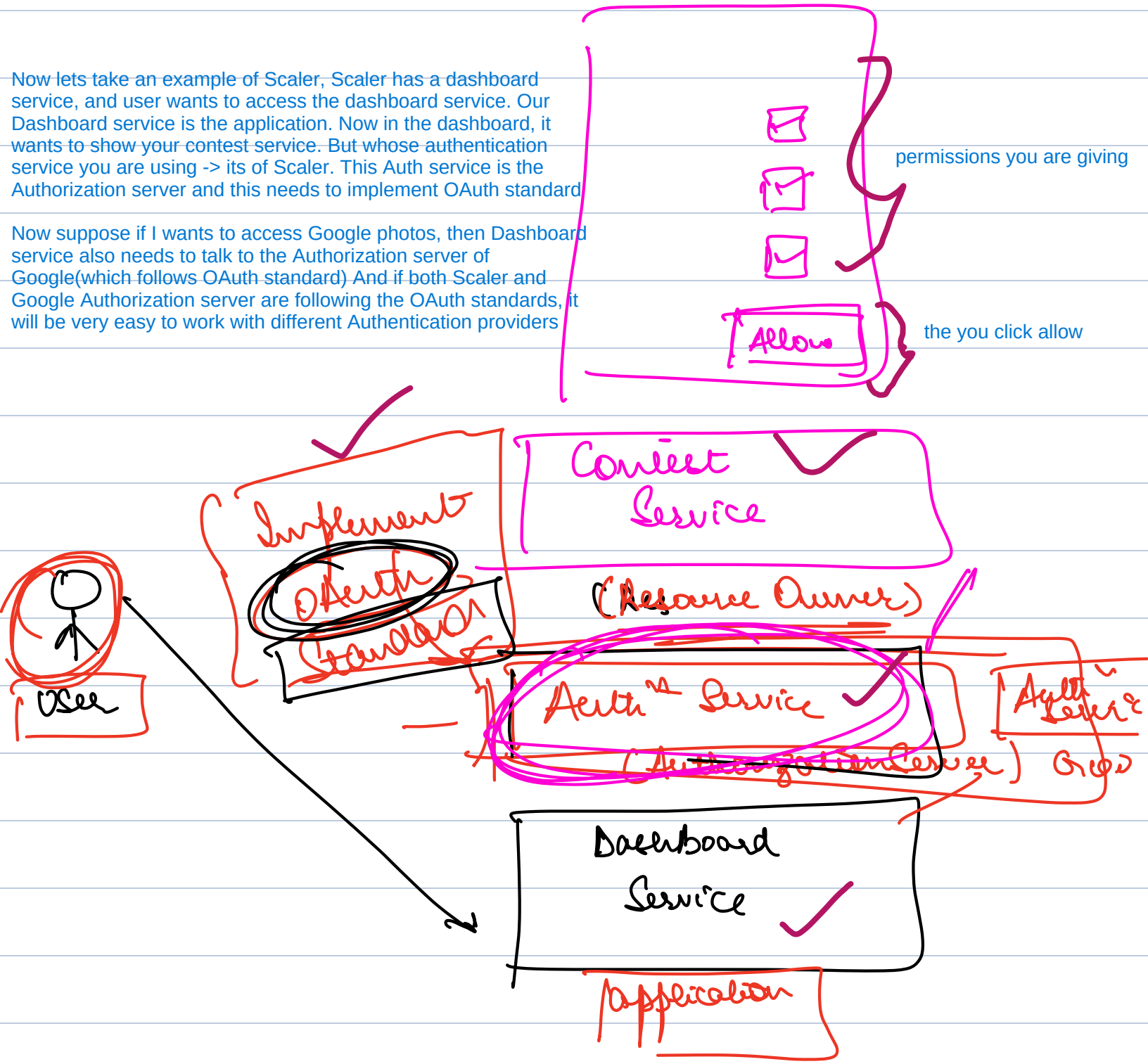
Example : (OUTLOOK) Mobile App that allows you to sign in via google and see your google mails

Building a mobile app

Authorization server is also managed by Google

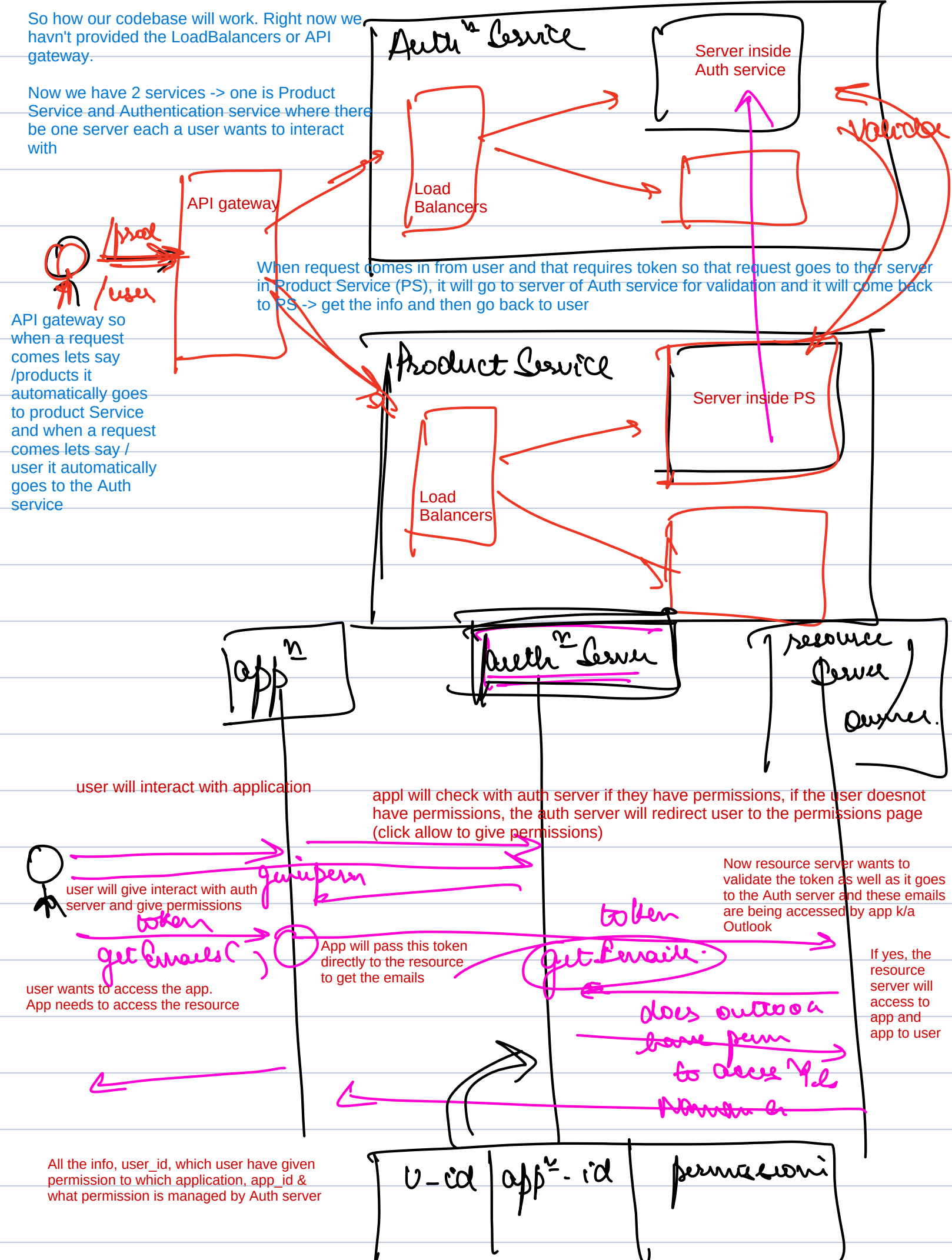
Now lets take an example of Scaler, Scaler has a dashboard service, and user wants to access the dashboard service. Our Dashboard service is the application. Now in the dashboard, it wants to show your contest service. But whose authentication service you are using -> its of Scaler. This Auth service is the Authorization server and this needs to implement OAuth standard

Now suppose if I wants to access Google photos, then Dashboard service also needs to talk to the Authorization server of Google(which follows OAuth standard) And if both Scaler and Google Authorization server are following the OAuth standards, it will be very easy to work with different Authentication providers



So how our codebase will work. Right now we havn't provided the LoadBalancers or API gateway.

Now we have 2 services -> one is Product Service and Authentication service where there be one server each a user wants to interact with



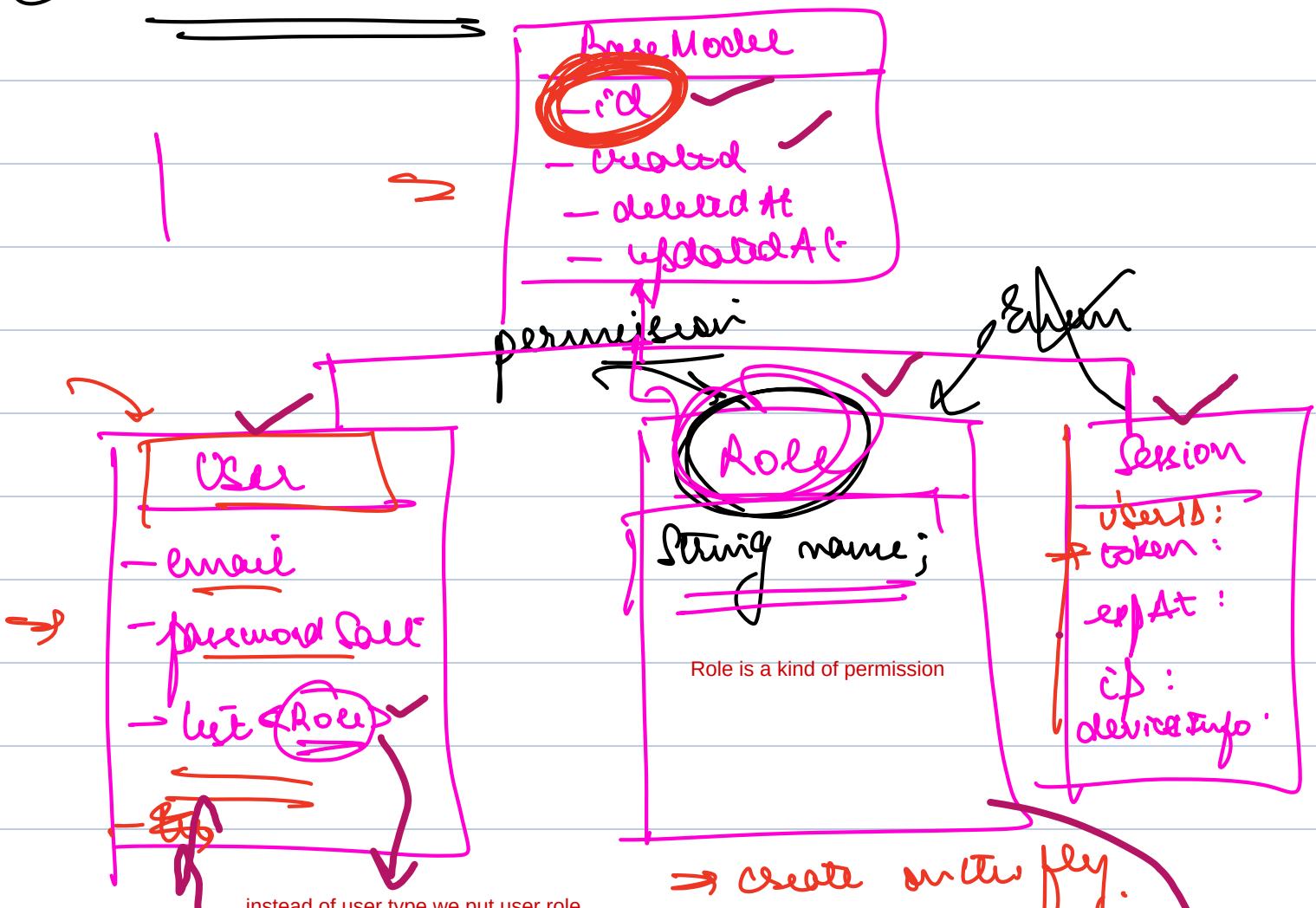
LLD of User Service

Design of User Service

1st step of LLD is decide models

1) Decide Models

User Service will have Base Models/Class, User, Role, Session



Role is a kind of permission

instead of user type we put user role which includes, students, mentee, instructor, admin that defines what permissions need to be given

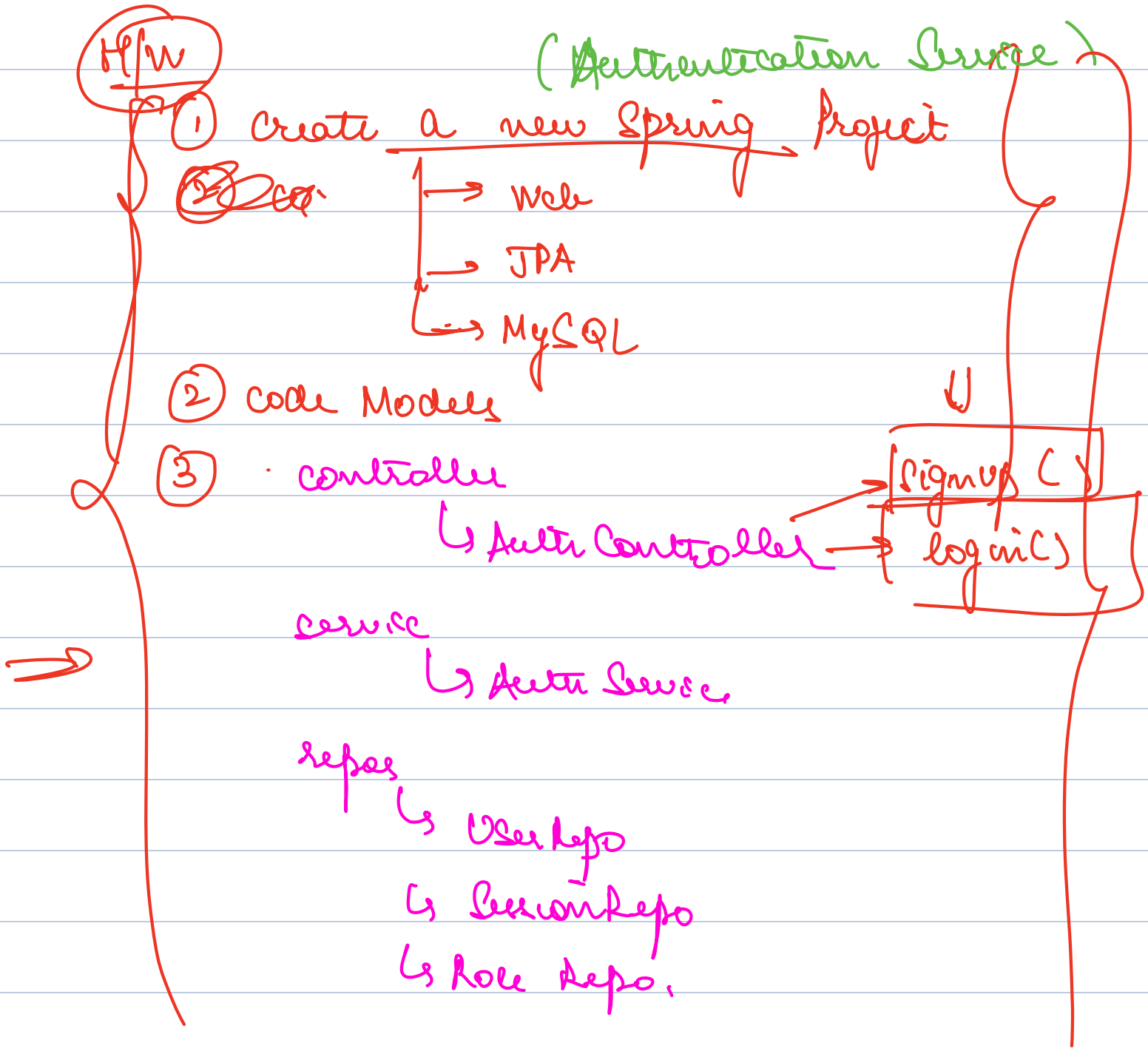
⇒ create on the fly.

- PHOTOS_READ
- PHOTOS_CREATE
- MAIL_READ
- MAIL_SEND

Eg: permission like PHOTOS_READ, PHOTOS_CREATE, MAIL_READ, MAIL_SEND

1) Signup
2) updateRole(u_id, [roles])

(3) login


H/W
(Authentication Service)
1 Create a new Spring Project
Web
JPA
MySQL
2 Code Models
3 · controller
Auth Controller
signup()
login()
service
Auth Service
repos
User Repo
Session Repo
Role Repo.

1