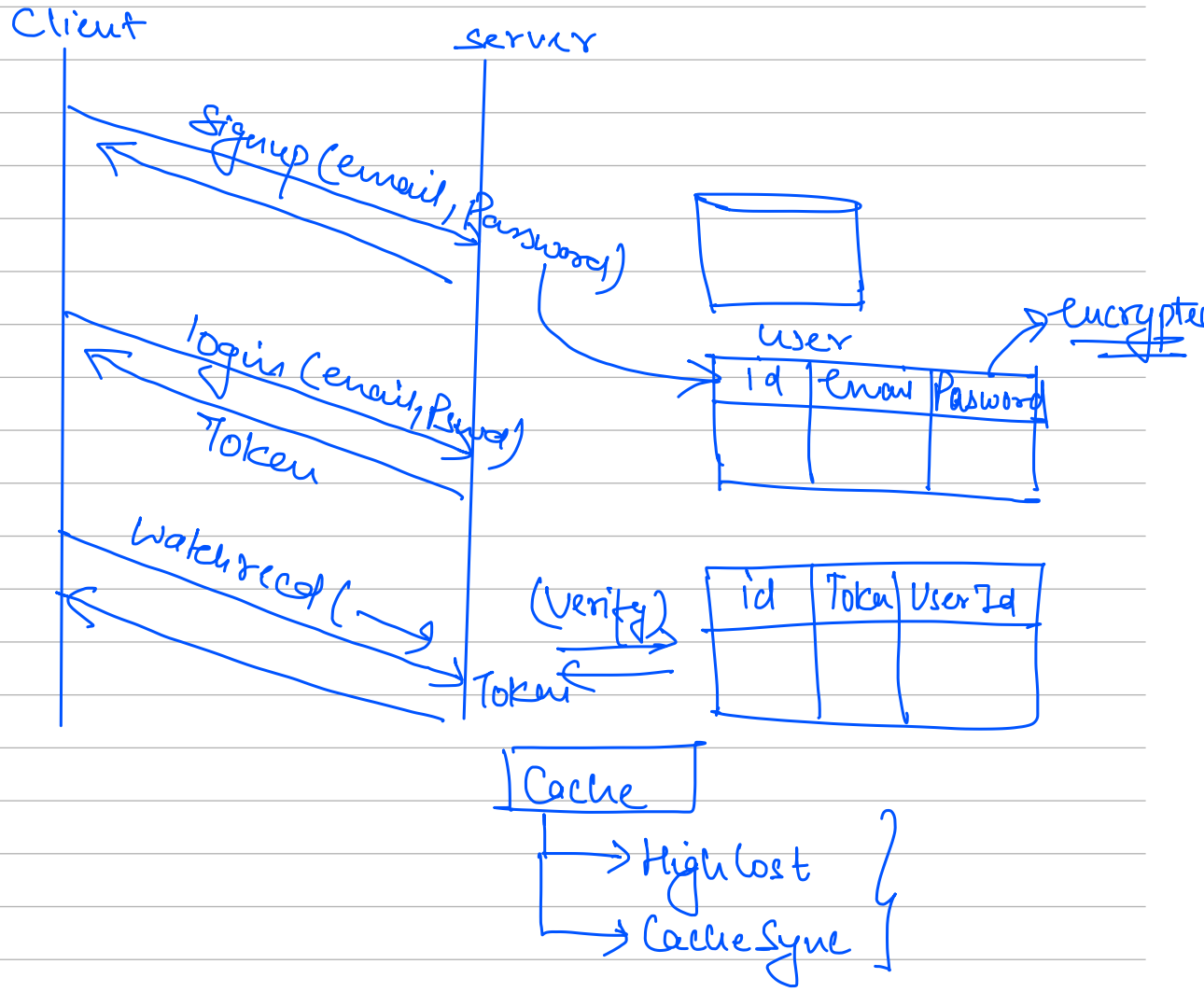
Agenda

1) JWT
2) OAuth
3) Start User Service

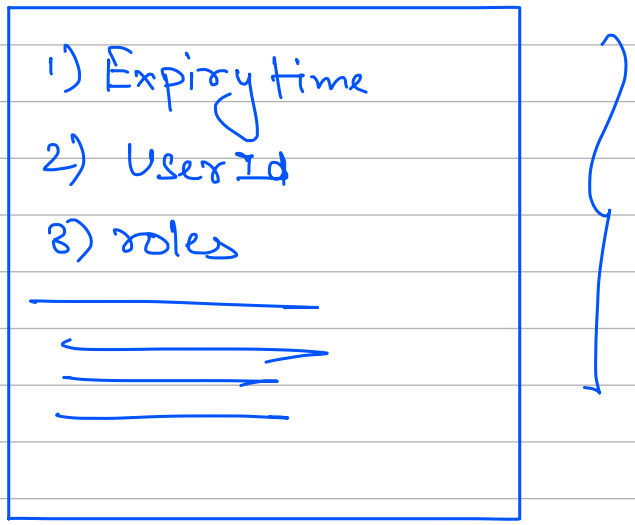
JWT → JSON Web token



① What if we can validate the token without even going to DB?

↳ If token contains all the required info to validate

Self Validating Tokens



Headers → Key Value Pairs

Username = Jayesh
Token = {
expiryTime: ——
username: ——
roles = [→] Just Admin
}

```
if (token.username = username)
{
    return true;
}
```

Valid Token

Token

64 Char
128 Char
] Alphanumeric

Encoding

Base64

1 = a
2 = b
⋮

Convert one form to another

Token = JSON String

[Base64 Encode]

Token = Base64 (JSONString)

① Can anyone see the token ✓
② Can anyone tamper the token ✓

Encryption = (Encoding + Secret key)

[Briefcase (encoding) + lock
Gold]

```
login (email, password) {
    // Verify email
    // generate JSON
    token = encrypt(json, secretkey);
    return token;
}
```

VAULTS

JWT JSON web token

A . B . C

Header Payload Signature

{ Header contains encryption Algo HS256 HMAC MD5 }
↳ Base 64 encoded

Payload contains the token details which are Base 64 encodes

Signature is encrypted value of (Header+Payload)

— Secret Key —

login(Email, password)

{

// Verify Email/Password

// Generate JSON → Username
→ Role
→ Expiry

B = base64Encode (json)

A = Base64Encode ({ "Algo" : "HS256" })

C = HS256.encrypt(A+B, secretkey)

[token = A.B.C]

return token;

}

ValidateToken (token)

{

X.Y.Z = token.split(".");

Algo = decode(X)

if (X+Y != Algo.decrypt(Z, secretkey))

{

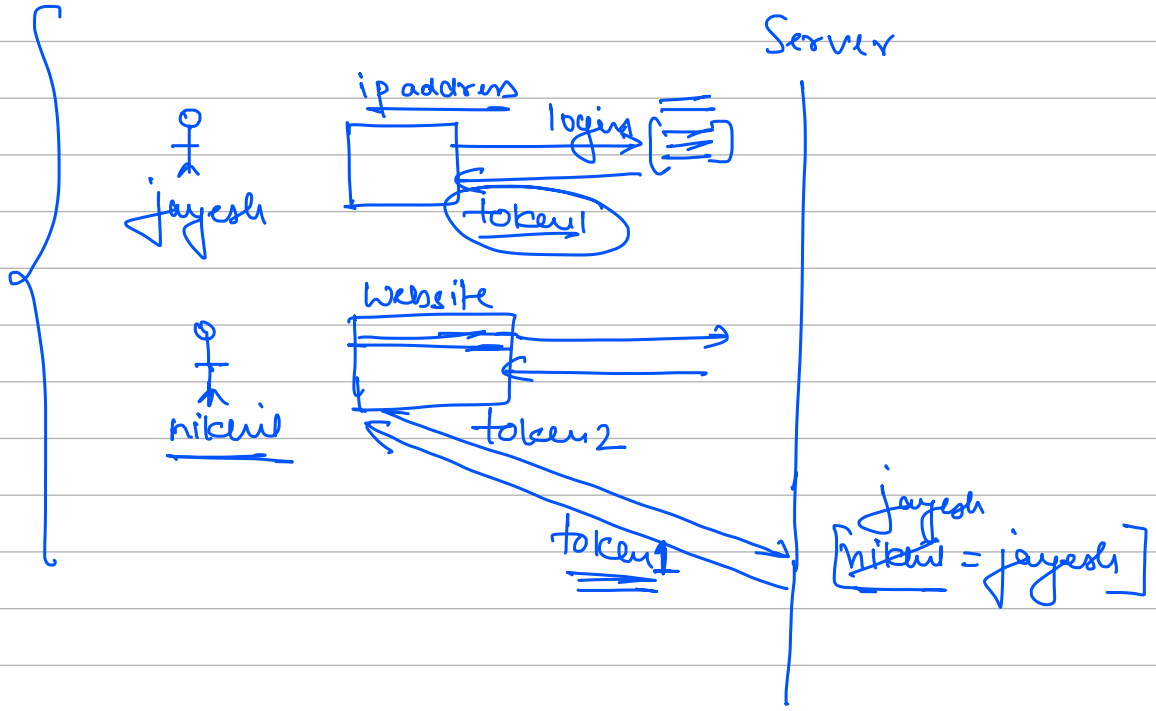
Token is NOT valid

}

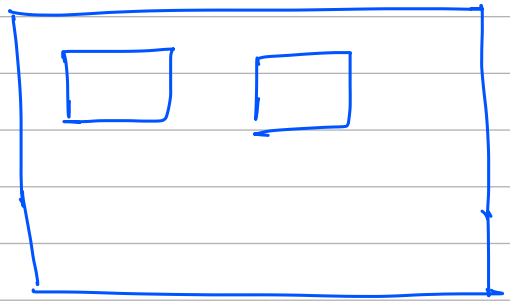
expiry time
username
Roles

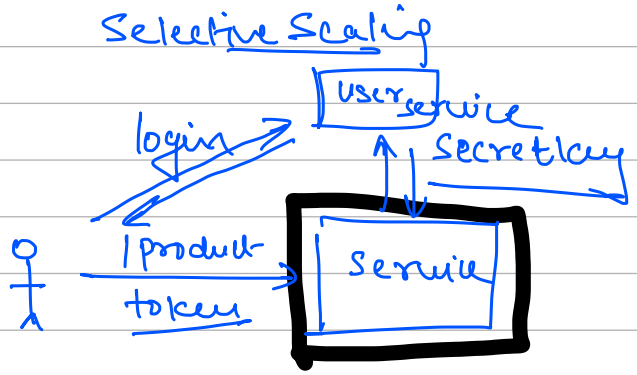
return true;

}



Monolith vs Microservice





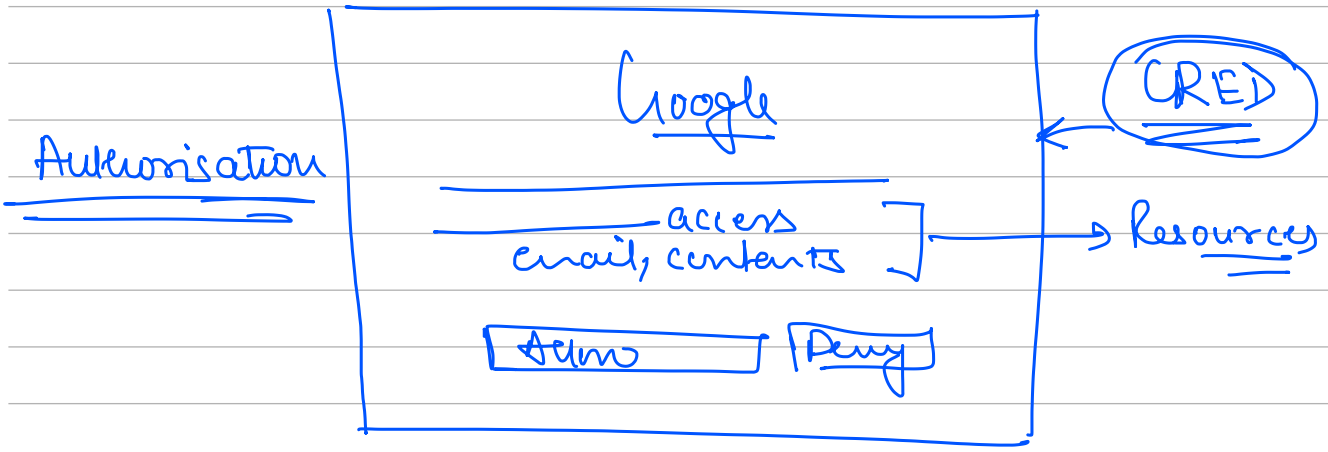
OAuth

login
signup

If every auth provider has their own impl
It is going to very difficult for us to integrate them in our api

→ Login Google
→ Linkedin
→
→
→ OAuth

Industry wide Standard for Authentication

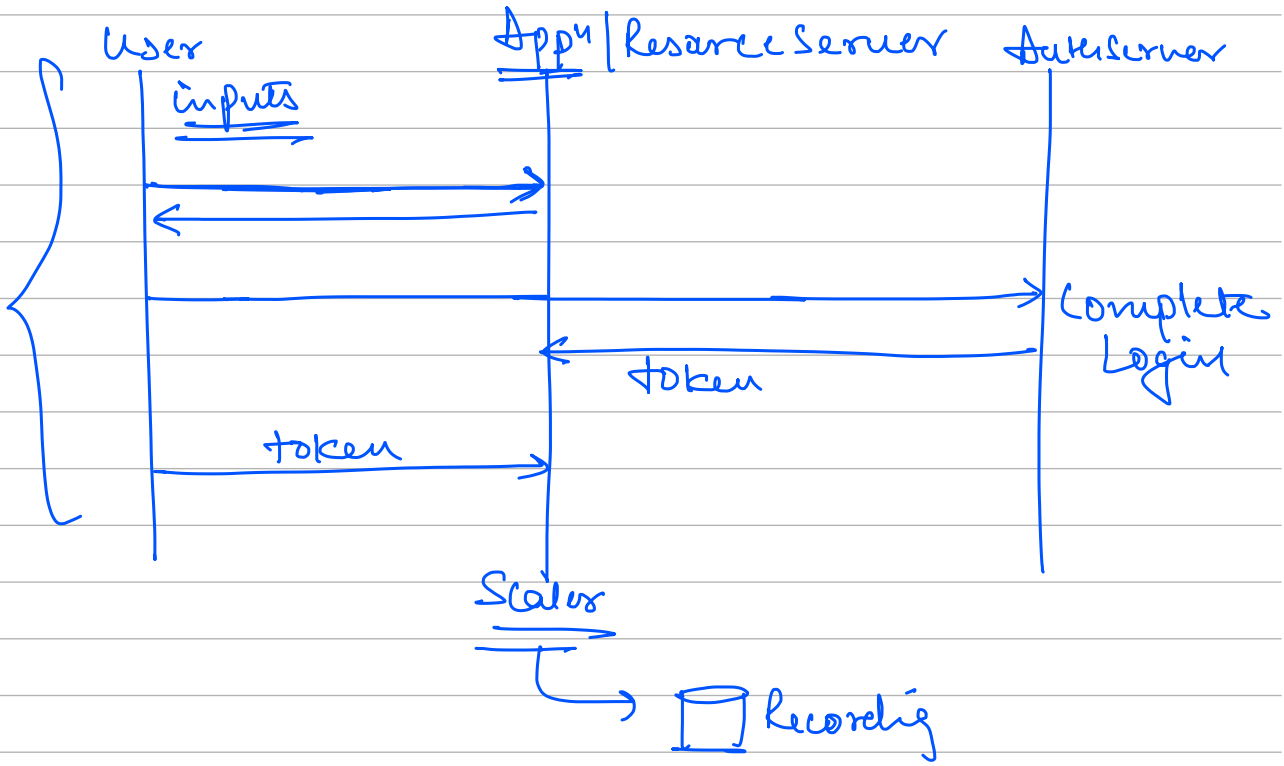


How OAuth works

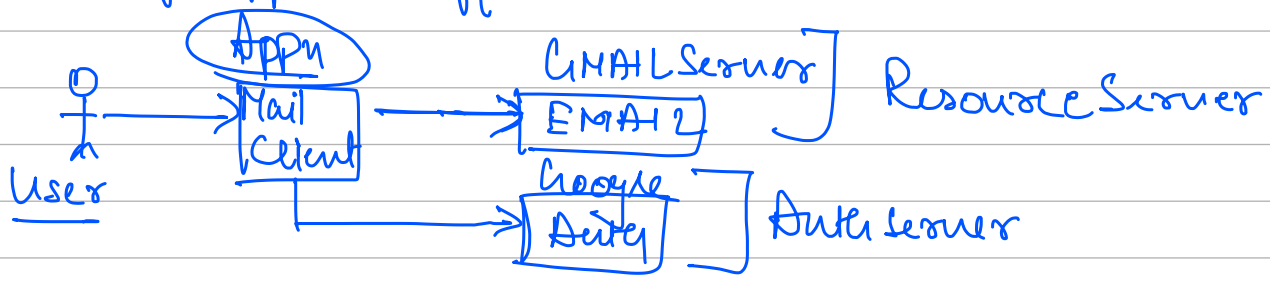
4 Components

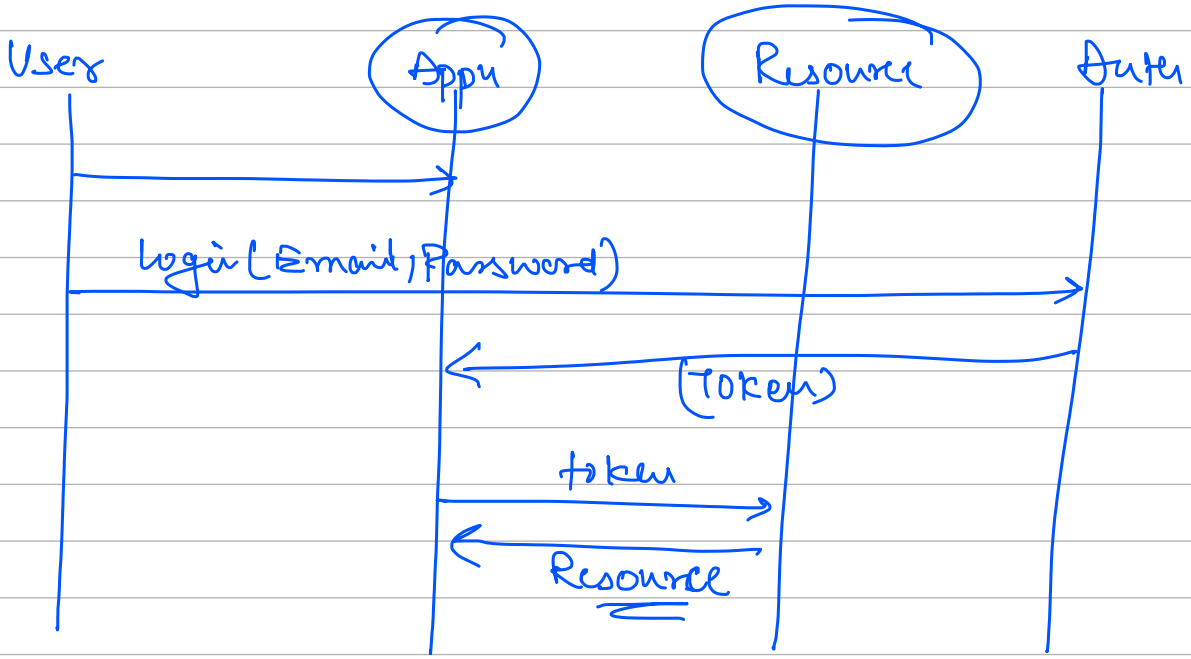
1. User : Person who wants access the data
2. Resource Server : Service that actually has the info required by user
3. Application : Service on which client wants to access the info.
4. Authorisation Server : Service that going to verify user and provide its Roles

① When App & Resource Server are Same.



② If App is different from resource server





UserService

→ AuthServer

- → login()
- → SignUp()
- → validateToken()
- → forgetPwd()

Model

Enum ✗
Class ✓

User

User
- id
- name
- email
- phNo
- hashedPassword
- List<Roles>

Role
- id
- value

User : Role ⇒ M:M
1 M
M 1

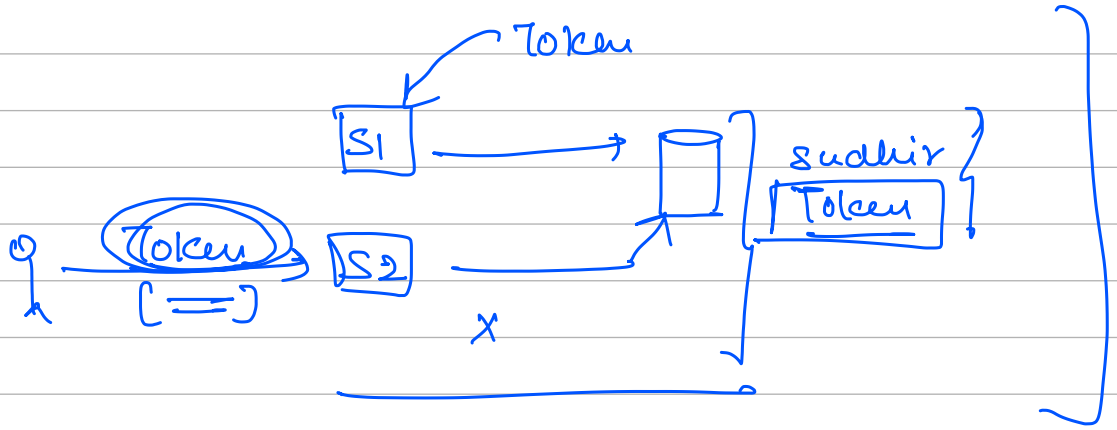
History
Audit

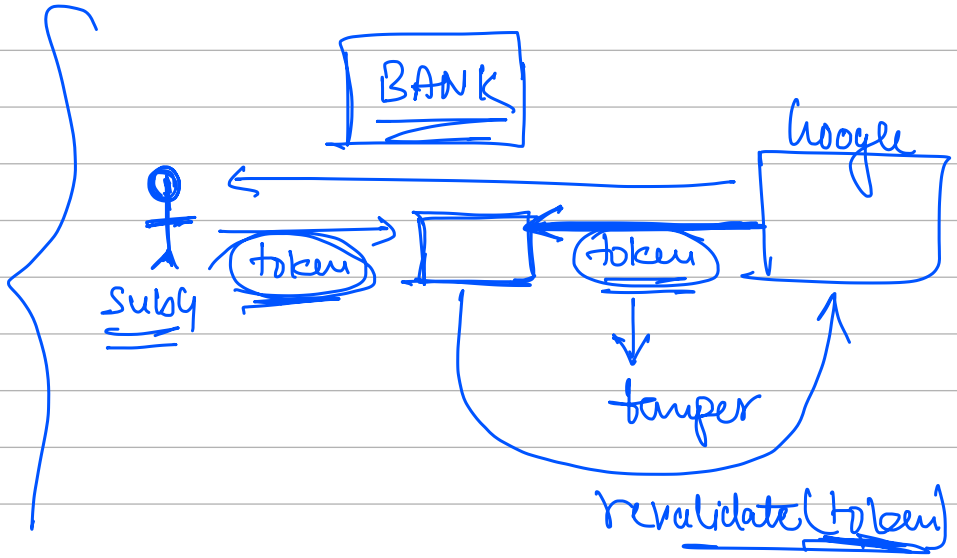
Token
- id
- value
- User
- expiry
- isdeleted

UserId	RoleId

HARD | SOFT Delete

Token : User ⇒ M:1
1 → 1
M ← 1


Token
S1
sudhir
Token
Token
S2
X


BANK
Google
Subu
token
token
tamper
revalidate(token)