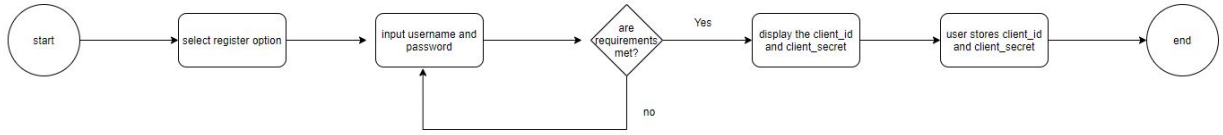
API Gateway design doc

Jason Nguyen #014232658

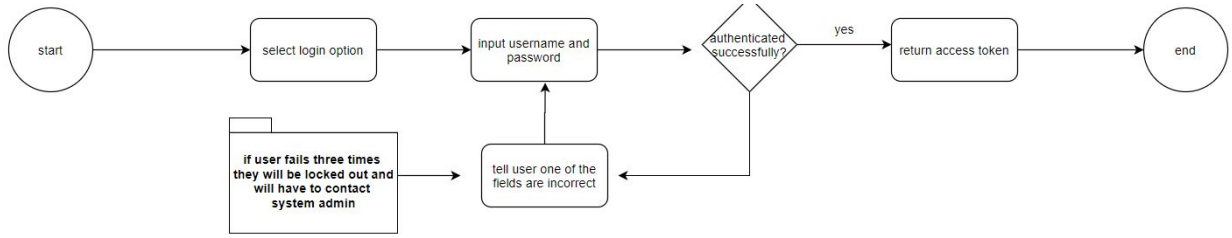
April 4, 2020

High Level

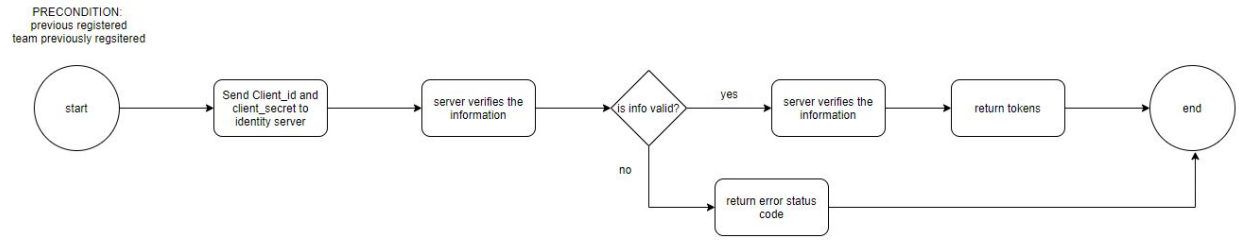
Team Registration

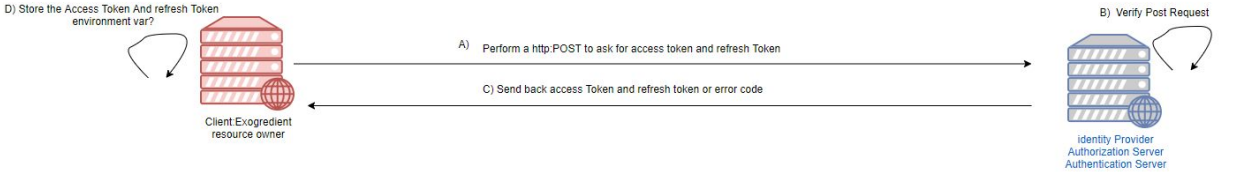


Team Login

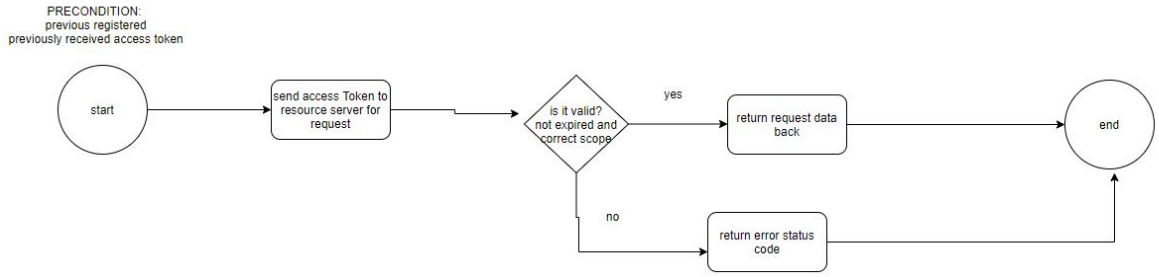


Get Access Token / Refresh Token (CCF)

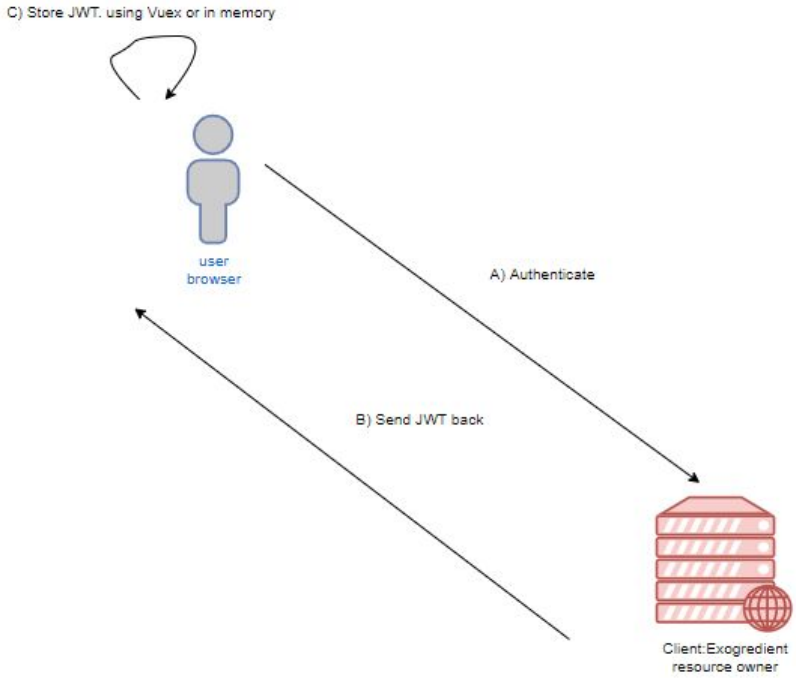




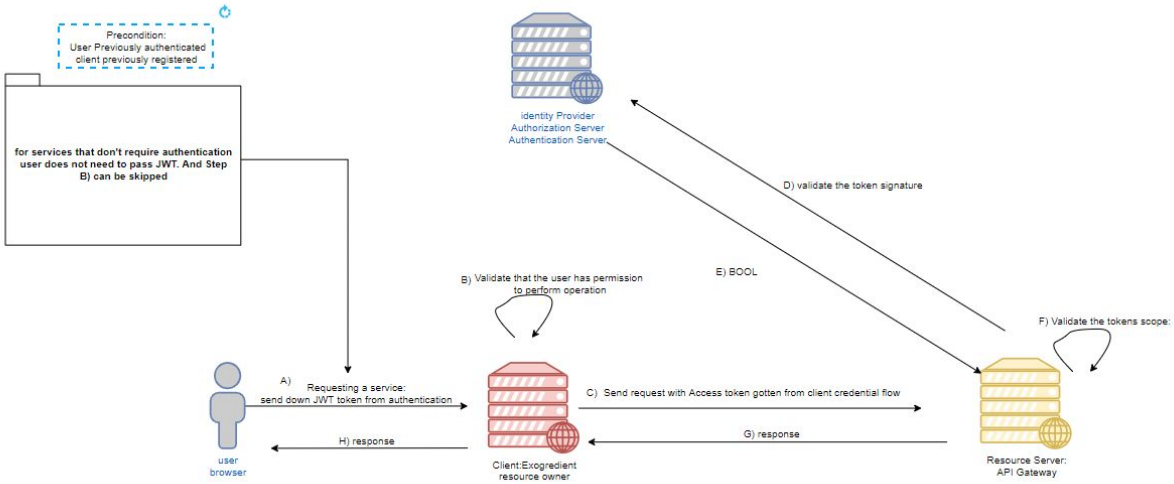
Use Access Token (Client Credential Flow)



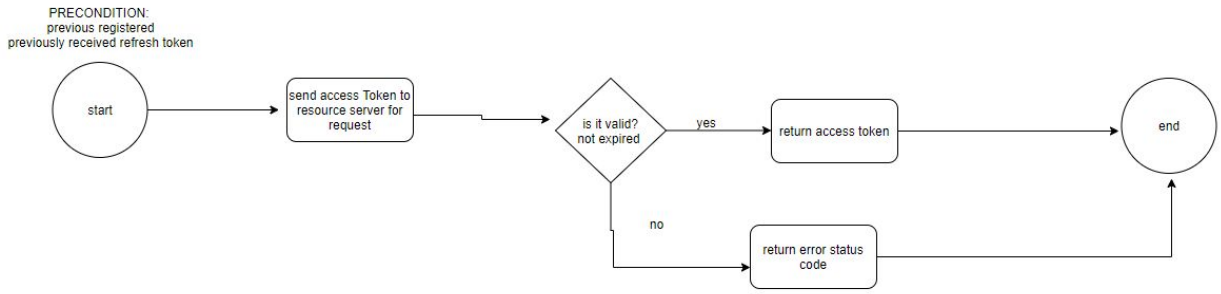
User login



Using access token with user authentication (Client Credential Flow)



Use Refresh Token (Client Credential Flow)



Low Level

Structure of JWT Access token
Structure of JWT:

- **Header**

  {
  “alg”: “RS256”,
  “typ”: “JWT”
  }

- **Payload**

  {
  “Iss”: “https://webapi.com”,
  “client_id” : “1231231231213”,
  "iat": 14700011113,
  "exp": 14700111113,
  “Scope”: “1231231231213”,
  }

*note: iss: issuer, client_id: gotten from registration, iat: issued at, exp: expired. Scope is equal to client_id
*note: Scope is equal to client id because the services can either be public or private to that client. And we can validate whether they are allowed to use it by this.

- **Signature**

  {
  RSASIGN(HEADER + “.” + Payload, private key)
  }

Structure of JWT refresh token

Structure of JWT:

- **Header**

  {

  “alg”: “RS256”,

  “typ”: “JWT”

  }

- **Payload**

  {

  “Iss”: “https://webapi.com”,

  “Client_id: ….

  "iat": 14700011113,

  "exp": 14700111113,

  “Scope”: “refresh”

  }

*note: iss: issuer, client_id: gotten from registration, iat: issued at, exp: expired.

- **Signature**

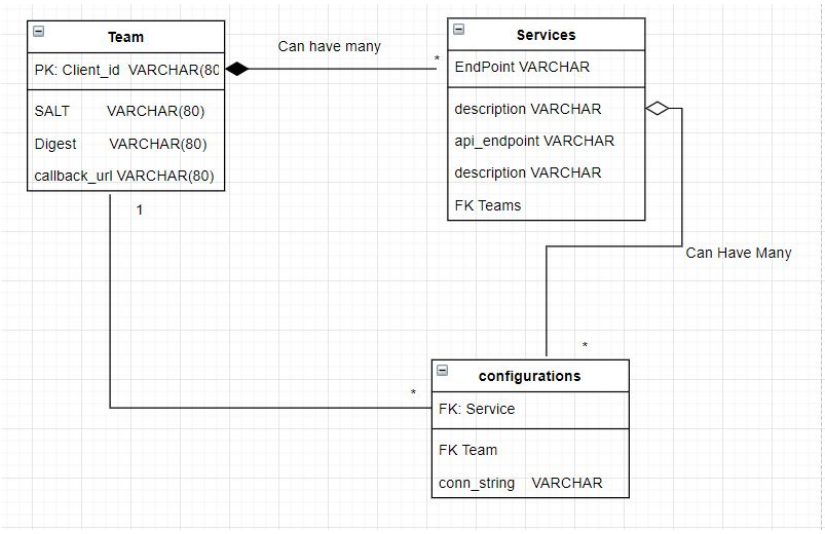
  {

  RSASIGN(HEADER + “.” + Payload, private key)

  }

Data Store

** simple initial implementation. Does not take routing into account


Team
PK: Client_id VARCHAR(80
SALT VARCHAR(80)
Digest VARCHAR(80)
callback_url VARCHAR(80)
Can have many
*
Services
EndPoint VARCHAR
description VARCHAR
api_endpoint VARCHAR
description VARCHAR
FK Teams
1
Can Have Many
*
*
configurations
FK: Service
FK Team
conn_string VARCHAR