가로선

**캡스톤 디자인 회의록**

2020년 4월 18일 토요일

**─**

# **참석자**

유성근, 김동규, 조혜령, 송찬훈, 신현승

# **회의 안건**

## 지난 회의 내용

1. 기존 웹 또는 앱 블록체인 서비스에서는 secret key를 어떻게 관리하는지,

secret key를 다운 받지 않고 유저를 인증할 수 있는 방법이 있는지 조사해오기

1. Commercial paper의 application structure 참고해서 하이퍼레저패브릭 구조

공부하고, 각자 투표 네트워크 구조 생각해오기

1. 각자 파트별로 javascript,node.js,ECMAscript 공부하기

## 새 내용

1. 네트워크 설계 채널 위주 간단한 초안 만들기 - 유성근 팀원
2. DB설계 초안(필요한 데이터 목록정도) 만들기 - 유성근 팀원
3. 각자 파트별로 javascript,node.js,ECMAscript 공부하기
4. 다음 회의일시 : 4월 21일 (화) 오후8시

* 네트워크 설계 공유 및 토의 (채널, organization, ca 등)
* DB설계 공유 및 토의(어떤게 원장에 들어갈건지, 프라이빗데이터에 들어갈건지 등 정하기)
* IBM evote 따라 해보기
* 체인코드 설계 시작?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **구분** | **금주계획** | **완료예정** | **금주실적** | **진척율(%)** | **차주계획** | **완료예정** |
| 네트워크 구조 설계 | 1. fabric-ca-client 정리하기  -암호학 수강한 팀원  2. fabric-ca-client 관련 책 찍어서 올리기 - 김동규 팀원  3. 기존 블록체인 서비스에서  sk 관리하는 방법 & sk 다운로드  없이 유저인증 할 수 있는 방법 조사해오기  4. 하이퍼레저패브릭 구조 공부 및 네트워크 구조 생각해오기 | 미정  미정  04.18  04.18 | ?  (cryptogen 사용할거니까  폐기?)  3. 키를 서버에 저장하는  delegated 모델로 결정  4. 유저인증 부분 토의 | 100%  30% | 1. 네트워크 설계 간단한 초안  (채널 위주) - 유성근 팀원 | 4.21 |
| DB 설계 |  |  |  |  | 1. DB설계 초안  (필요한 데이터 목록정도)   * 유성근 팀원 | 4.21 |
| 기타 | 1. 프로그래밍언어 공부  ( javascript,node.js,  ECMAscript 등 공부하기) | 미정 |  |  | 1. 프로그래밍언어 공부  ( javascript,node.js,  ECMAscript 등 공부하기) | 미정 |
| 특이사항 |  | | | | | |

# **메모**

* **기존 앱or웹 서비스에서 sk를 관리하는 방법(혜령팀원)** 
  + Amazon Web Services 인증정보관리

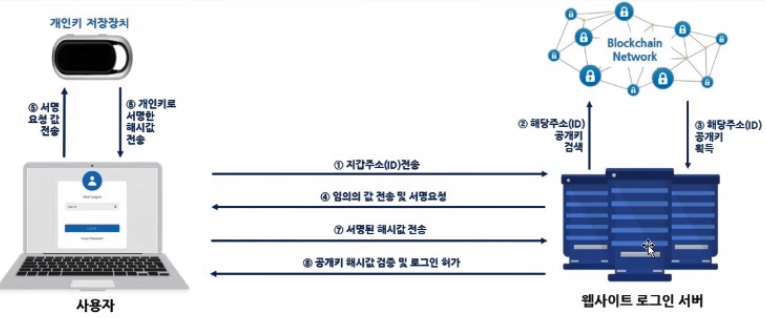
Access key(id)와 Secret Access Key(secret key)가 있음

아마존을 호출할때마다 간단하게 인증정보설정이 가능하도록 커멘드 제공

커맨드를 통해서 설정한 정보들은 사용자의 컴퓨터에 저장

* + HDAC technology 블록체인을 이용한 전자서명 기반의 사용자 인증처리 방법

<https://medium.com/hdackorea/%EB%B8%94%EB%A1%9D%EC%B2%B4%EC%9D%B8-%EA%B8%B0%EB%B0%98-%EB%A1%9C%EA%B7%B8%EC%9D%B8-%EB%B0%A9%EC%8B%9D-bb3ddb0273b7>



1. 사용자가 로그인을 위해 웹사이트에 지갑주소 전송
2. 해당 주소의 공개키를 검색
3. 공개 키 획득
4. 임의의 값 및 서명 요청(일종의 challenge?)
5. 개인 키 저장 장치로 유저가 서명하고, 서명한 값 해시값으로 변환해서 웹사이트로 전송
6. 공개키 해시값과 비교해서 로그인 허가

=> 사용자인증을 할때 이런 방향으로 하는 것도 나쁘지 않을듯.

그런데 사용자가 usb를 사용하거나 어플리케이션을 통해 저장할수있게

해야함 - 개인키 저장장치를 어떻게 처리하느냐가 관건

* + DID

개인이 지정한 데이터 저장소에 개인정보를 저장한 후 원하는 정보를 선택해 제출할 수 있는 블록체인 기반 신원증명 솔루션 ( 개인 장치 필요함)

=> 하이퍼레저 인디와 연계할 수 있을까...?

=> 문제는 이런 decentralized 모델에서는 sdk가 지원이 안되므로 우리가 다

개발해야됨....따라서 사실상 졸업프로젝트에 적용하기는 힘들고 향후 발전방향으로서

삼을 수 있을 것 같음.

* **하이퍼레저패브릭에서 주로 사용하는 유저관리 모델**
  + <https://github.com/johndonggyu/HLF-Study/tree/master/Curriculum> 강의목록

10번, 11번 참고

* + 유저관리에는 3가지 모델이 있다.

1. Delegated

Balance-Transfer 모델

서버가 모든 유저의 키를 관리

회원 관리하는 서버가 유저id(jwt token)과 인증서를 매칭해서 트랜잭션 생성

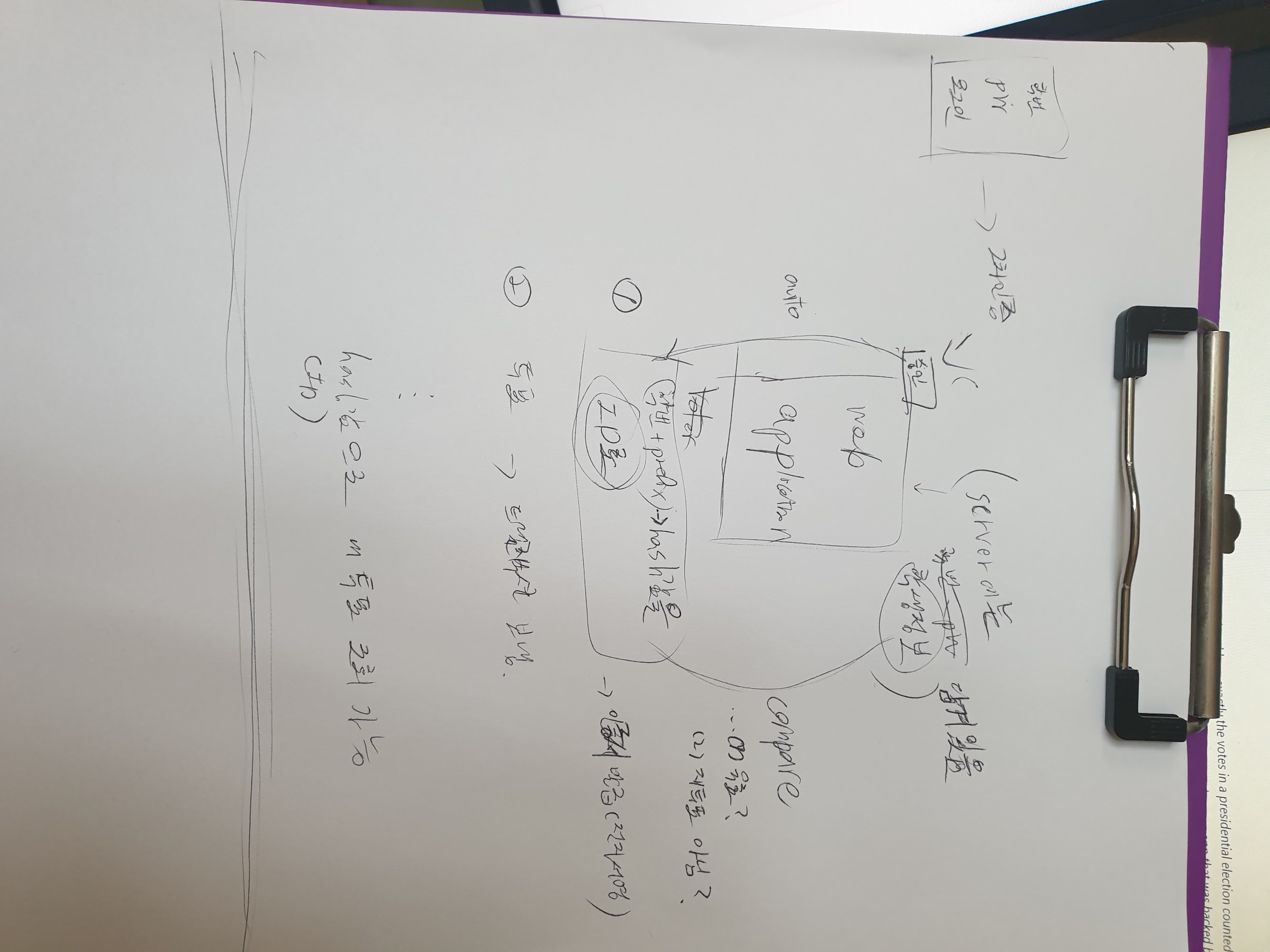
Fabric CA를 통해 유저 생성(sign up)

1. Decentralized
2. Chaincode level user

=> 현실적으로 구현 가능한 것은 Delegated 모델일듯

* **우리가 사용할 모델 토의내용**

1. **성근팀원의 네트워크구조**



- 학생이 학번으로 인증 후, 2차인증해서 웹서버로 들어옴

2차인증 하자마자 ca에서 공개키와 비밀키를 하나 만들어줌

웹서버안에는 학교 통합인증시스템에서 받아온 학생정보가 담겨있고

사용자는 학번에 뭔가를 붙인 값이 해시된것을 id로 가짐

이 id가 투표하는데 인증할수도 있고 자기가 투표한것을 나중에 불러올수도 있는 하나의 꼬리표가 되는것

학번을 그대로 쓰면 다른사람이 도용할 위험이 있으므로 해시값 취함

(ibm evote의 구조와 유사)

- 레지스터유저 언제 일어남?

유저처리는 웹에서 하는거고 레지스터유저는 웹어플리케이션을 불러올때만 함. 서버 시작할때 1번만

즉, 여러 사용자들이 웹어플리케이션을 이용하는데 (이미 ca로부터 신뢰를 받은)웹 어플리케이션은 유저 1 하나의 아이덴티티를 모든 사용자에게 연결시켜주는것

1. **찬훈 팀원 의견**

- 키를 서버가 모두 가지고 있는것은 위험! 왜 키를 개인이 계속 가지고 있어야하느냐?

예전에 투표를 할때 38%였는데 한참동안 그게 유지되다가 마지막에 갑자기 40%가 넘어가는 일이 있었음. 그 후로 제 1순위 잠재 어태커가 선관위이고 투표율을 위해서 손을 댈수도 있겠다고 생각함.

따라서 서버를 통해 구현한다면 서버를 누가 관리하느냐가 관건이고, 선관위는 서버를 관리해서는 안됨(안전한 제 3자가 맡아야)

//그런데,

ㄱ. 서버 털었다고해서 공격이 가능한가?

- 중복투표안되니까 저장된 키 정보가 있다고 하더라도 다시 할 수 없음. 아이디에 해시함수 등 여러 조치를 취하면 아이디 추적도 할 수 없음.

만약 sk에 무작위 해시값넣어가지고 공격하면?

- 인돌싱정책(관리자만 아는 랜덤값을 아이디에 추가하는 등)으로 거를수있음

공격자가 동일한 샘플을 사용하면 공격에 성공할 수 있지 않을까?

- pk랑 sk쌍을 미리 만들지 않고 2차인증을 하는 순간 ca가 인증서를 발급을 하는 동시에 msp에 저장을 한다면 인증서가 필요하므로 공격할 수 없음 (인증서까지 탈취한다면 서버인척은 할 수 있을듯)

=> 어느정도 서버에서도 기본적인 공격 방지를 위해 보안 처리가 필요할듯

서버관리는 개발자인 우리가 하거나 정보통신팀이 맡을 수도 있음

ㄴ. 선관위는 어태커인가?

총학생회 시행세칙을 보면 선관위 위원장이 대의원회 의장이 된다는 조항이 있지만, 대의원회는 학생회 견제기구임(국회같은느낌, 학생회는 행정부)

게다가 대의원회랑 학생회는 겸직할 수 없음. 따라서 선관위를 어태커로 보기에는 무리가 있음 //

1. **동규팀원 의견**

우리는 3가지 모델을 생각해 볼 수 있음

ㄱ. Delegated 모델

- 웹서버를 구축할 때, 애초에 웹서버에 admin, user1 만 생성하여, 이후 모든 웹서버에 로그인한 사용자들은 user1에 해당하는 identity

를 사용하여 쿼리, 업데이트 등을 보낸다. (관리 편하지 않을까?)

// 별도의 user번호 없이 웹서버에 로그인한 모든 사용자가 user1을 사용한다면, 공격자 역시 용이하게 user1의 identity를 사용할수 있지 않을까?

// 동시에 10명 100명 접속하면 서버 트래픽 때문에 안되지 않을까?

* async(비동기)로 하면 괜찮지 않을까?

ㄴ. Delegated 모델

- 처음 로그인 할때에만 웹서버 자체에서 user# identity를 생성하도록 하기. 이렇게 하면 웹서버가 총괄해서 identity를 보관하게 된다

(위에 처음 로그인 시라고 표현한 이유는, 우리는 통합로그인에 의지하고 따로 회원가입을 만들지 않기 때무에 그렇게 표현함)

//이 모델을 사용할 경우 키젠연산이 많이 이루어지므로 오버헤드가 발생할 우려가 있음 따라서 크립토젠을 사용해 키를 미리 만들어서 키쌍 하나씩 주는게 좋을듯. 원장 내에서의 익명성 보장 가능할듯

ㄷ. Decentralized 모델

- 기존의 p2p처럼 개개인이 개개인의 디바이스에 아이덴티티를 가지고 있도록 한다. 이렇게 될 경우에 첫 로그인 시 개인의 디바이스간에 CSR을 수행할 수 있도록, 우리가 CSR인터페이스를 지원해줘야함) 웹 앱을 구축한다고 가정한다면 priv를 다운로드 함은 물론 (pub key나

certificate 은 웹으라우저에서 보관기능 지원해줌) 이후 transaction 보낼때 priv key를 첨부하여 보낼 수 있도록 따로 웹 UI가 필요할 것 같다

(여기에다가는 당연히 TLS를 적용해야함)

ㄷ은 현실적으로 힘듬.

ㄱ,ㄴ은 장단점이 있으므로 둘 중에 뭘 할지 생각해봐야 함..

ㄱ의 경우 사용자는 네트워크 구조상 하나의 트랜잭션이라고 보면됨. 따라서 각각의 사용자에 대해서는 msp불필요할듯 (커머셜페이퍼의 balance transfer)

// 모두 같은 유저를 쓰면 사용자마다 투표 가능한 선거 다른 부분 어떻게 해결? - 단과대별로 유저 만들면 될듯 (학번이 단과대별로 묶여있으므로 정책도 처리하기 쉬울듯)

// 다 같은 유저 쓰면 내 투표 확인은 어떻게 함??

- 아이디값을 넣어서 쿼리를 해서 원장을 불러오거나, 아니면 페브릭에서는 아예 모든 값 다 던져주고 웹에서 아이디 통해서 그것만 보여주는 걸로 할수도 있을듯

ㄴ에서는 각각 유저#라는 이름을 달고 들어오는거 그러려면 각각 msp가 필요

=> 현재 결론 : ㄱ모델로 구현하되 유저를 단과대별로 생성, 그리고 성능을 위해 키 생성은

동적으로 하지 않고 cryptogen을 활용해 미리 생성

* **기타 회의 내용**
  + **msp는 어디에 저장되는가?**

피어, 오더러, 유저 msp는 로컬 스토리지에 저장된다고 한다.

msp 라는 폴더를 가지고 있는 노드는 피어랑 오더러 뿐인데

유저가 msp가지고 있다는 것은 무슨말인가?

- 피어와 오더러의 msp 폴더 안에있는 유저 계정에 msp가 있다는 뜻.

그렇다면 네트워크 msp와 채널 msp는 어디에 저장하는가?

로컬 폴더에 저장되는것이 아니라 블록에 저장됨

configtxgen로 네크워크 처음 만들때 블록에다가 채널과 네트워크 관련 설정정보 (msp포함)를 저장하는것.

* + **CSR**(Certificate Signing Request)

fabcar예제의 enrollAdmin.js과 registerUser.js에서 공통적으로 사용되는게 csr

이 과정은 여러가지 과정으로 이루어져있는데

1. 프라이빗키, 퍼블릭키 생성
2. ca에게 퍼블릭키 전송
3. ca가 그 퍼블릭키를 받고 인코딩된 인증서를 앱에게 보내줌
4. 앱은 월렛폴더를 만들고 프라이빗키, 퍼블릭키, 인증서 저장
   * jwt token(bearer토큰) - 동규팀원이 조사해본다고 함
   * **다음주 회의시간에 할것**

db설계 : 원장구조, 프라이빗데이터 구조, 서버db구조

- 투표시스템을 잘 알고 있는 팀장님이 화요일까지 하나의 대략적인 DB설계(필요한 데이터 목록 정도) 해오면 이중에서 어떤게 원장에 들어갈건지 프라이빗데이터에 들어갈건지 등을 만나서 토의해봐야함.

네트워크 설계

- 역시 투표시스템을 잘 알고있는 팀장님이 채널위주로 대략적인 네트워크 설계를 해오면 organization, ca 등도 토의해봐야함.

IBM evote 따라해보기

-<https://developer.ibm.com/patterns/how-to-create-a-secure-e-voting-application-on-hyperledger-fabric/>

다 끝내고 화요일부터는 체인코드 설계 들어갈 수 있도록...