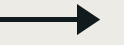


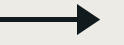
YOUUMIN PARK'S

Portfolio



발전 하는 개발자
박유민 입니다.

skills



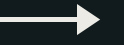
Python, Node js, Git, Docker, MySQL,
Rest API, Java,, Go, hyperledgerfabric,
linux

PROJECTS



1.UFO

2. KNUP



Project 1. UFO

2020.07-

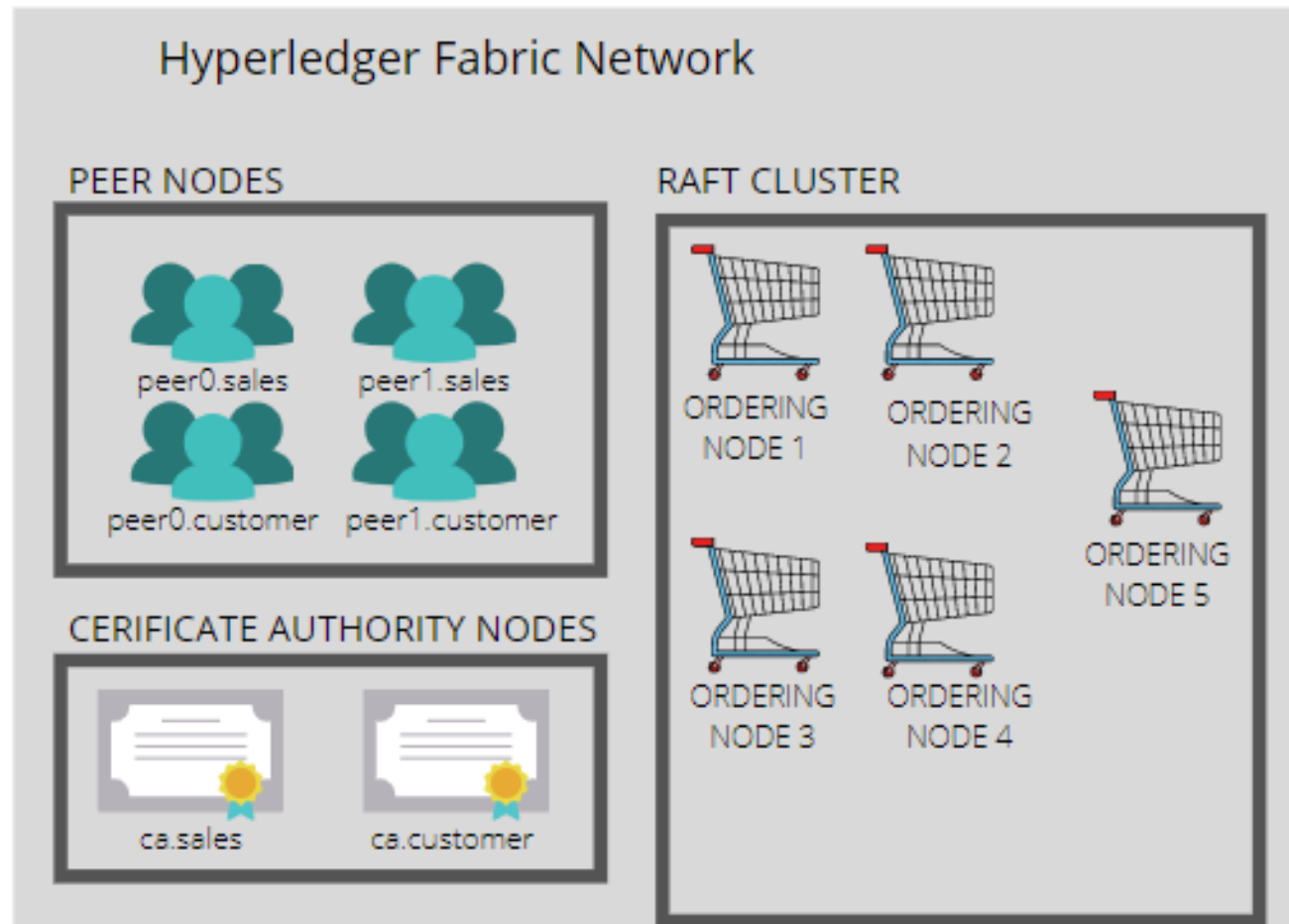
UFO?

- 명칭
University Festival in One
- 개요
전국 대학 축제를 하나의 플랫폼에서 확인하고, 고유 화폐로 결제
- 소개
각 대학의 축제를 하나의 어플로 확인하고 각종 정보를 확인할 수 있으며,
블록체인 Hyperledger Fabric으로 거래 네트워크를 구성해 안전하고 신빙성
있는 거래환경 제공
- 개발환경
 - Nodejs**
->서버 구성
->Hyperledger Fabric의 Chaincode 개발 언어
 - Docker & Docker Compose**
->Hyperledger Fabric 네트워크를 도커 컨테이너로 구성
 - Swifti**
->OS mobile application 개발 언어
 - AWS**
->EC2, LoadBalancer를 이용하여 NodeJS 서버와 Hyperledger Fabric 네트워크를 구성
RDS MySQL을 통해 축제 관련 정보, 유저 데이터 관리
S3를 이용해 Hyperledger Fabric의 인증서 및 축제 관련 이미지 등 정적데이터 관리
IAM 정책을 통해 각종 액세스 관리

- 팀 내에서 맡은 역할
- 블록체인 HYPERLEDGER 담당자
- ChainCode 작성(node js)
- Fabric(raft)
- fabric sdk server 작성(node js)

UFO?

블록 체인 네트워크 : 우분투 18.04 환경에서 작업



- sales, customer 2개의 조직으로 구성
- 각 조직마다 2개의 peer 노드로 구성
- raft 구현을 위해 5개의 orderer
- Couchdb를 world state로 사용

개발과정 블로그

https://yuminee.github.io/2020/11/27/Hyperledger%20fabric/raft_algorithm2/

https://yuminee.github.io/2020/11/29/Hyperledger%20fabric/raft_algorithm3/

UFO?

체인코드

- Init() : 체인코드 instantiate을 위해 넣어주었다. Transaction Id 를 console.info로 보여준다.
- initWallet(id) : 사용자(id)가 가입을 하면 지갑을 생성하고, balance를 0으로 초기화 해준다.
- chargeMoney(id, amount) : 가입한 사용자의 id에 amount만큼의 돈을 충전한다.
- getBalance(id) : 사용자(id)로 현재 잔액을 조회한다.
- transferMoney(sender, receiver, amount) : sender는 receiver에게 amount만큼의 금액을 보낸다.
- deleteWallet(id) : id로 지갑정보를 지운다(회원 탈퇴)
- getWalletHistory(id) : getHistoryForKey를 이용하여 해당하는 id에 대한 트랜잭션 history를 return

개발과정 블로그

<https://yuminee.github.io/2020/12/01/Hyperledger%20fabric/ufochaincode1/>

UFO?

initWallet 실행시 지갑은

```
wallet{
  Id : id
  Token : "0"
  invoke : "initWallet"
}
```

으로 초기화 됨

위에서 invoke는 모바일에서 거래 기록을 요청하여 받아올때 어떤 체인코드 함수가 실행되어서 나온 결과인지를 알려주기위해 추가하였음

```
{
  TxId: 'f6e47b71268cdaba26459b2dec9bfbcb5f8cb1d520e2548eb5a1130a81f7059c6',
  Timestamp: { seconds: [Long], nanos: 93269744 },
  IsDelete: 'false',
  Value: { ID: '2015116581', Invoke: 'sender', Token: '4000' }
},
{
  TxId: 'd46fa9344b0c4362747f3cec1d401fd6d0b8c8006f38093f6a26f41124fa9d9e',
  Timestamp: { seconds: [Long], nanos: 744770167 },
  IsDelete: 'false',
  Value: { ID: '2015116581', Invoke: 'chargeMoney', Token: '5000' }
},
{
  TxId: '76749906f3574c1df1e6220416b3129ae708db0c6a9c897c4b43953579c97a09',
  Timestamp: { seconds: [Long], nanos: 465174180 },
  IsDelete: 'false',
  Value: { ID: '2015116581', Token: '0', Invoke: 'initWallet' }
}
```

getWalletHistory 실행시 화면

패브릭 sdk 서버

모바일 서버가 JSON 으로 아래의 주소로 보내면 해당하는 Chaincode invoke, query 발생

- initWallet
 - POST /initWallet

key	value	Description
id	String	받은 id 값으로 Token을 0으로 초기화 하여 지갑 생성

- getBalance
 - POST /getBalance

key	value	Description
id	String	받은 id 값의 Token return

- deleteWallet
 - POST /deleteWallet

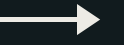
key	value	Description
id	String	받은 id값의 wallet 지움

- chargeMoney
 - POST /chargeMoney

key	value	Description
id	String	충전할 Wallet의 id
amount	String	충전할 금액

- transferMoney
 - POST /transferMoney

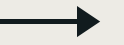
key	value	Description
sender	String	보내는 wallet Id
receiver	String	받은 wallet Id
amount	String	sender 가 보내는 Token의 amount



Project 2. KNUP

2020.03 - 2020.10

KNUP?

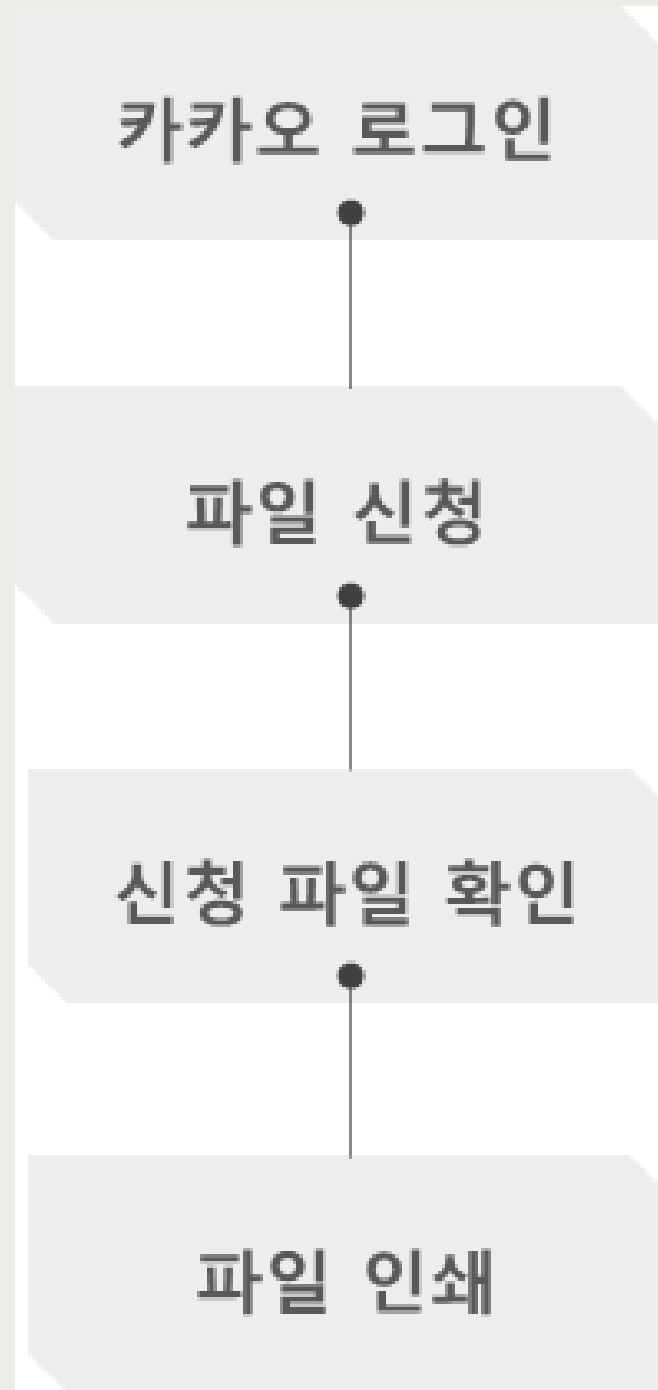


- 명칭
Kyunpook **N**ational **U**niversity **P**rint system
- 장르
Web Util Application
- 개요
3명(변상현,송동명,박유민)이서 팀을 이뤄 진행
카카오 로그인을 통하여 별도의 저장매체 없이 빠르게 인쇄
- 소개
교내의 복사실 및 학교 앞 편의점에서 인쇄를 할 수 있는 환경이 있지만, 별도의 저장매체 및 이메일을 이용하여야 하고, 한글 및 워드 등 별도의 프로그램을 통하여 열람 및 인쇄를 하여야 한다. KNUP은 이 모든 작업이 웹에서 이루어 지기에 저장매체를 휴대할 일이 없고, 웹에서 열람 및 인쇄를 해서 별도 프로그램 로딩시간을 단축할 수 있다.

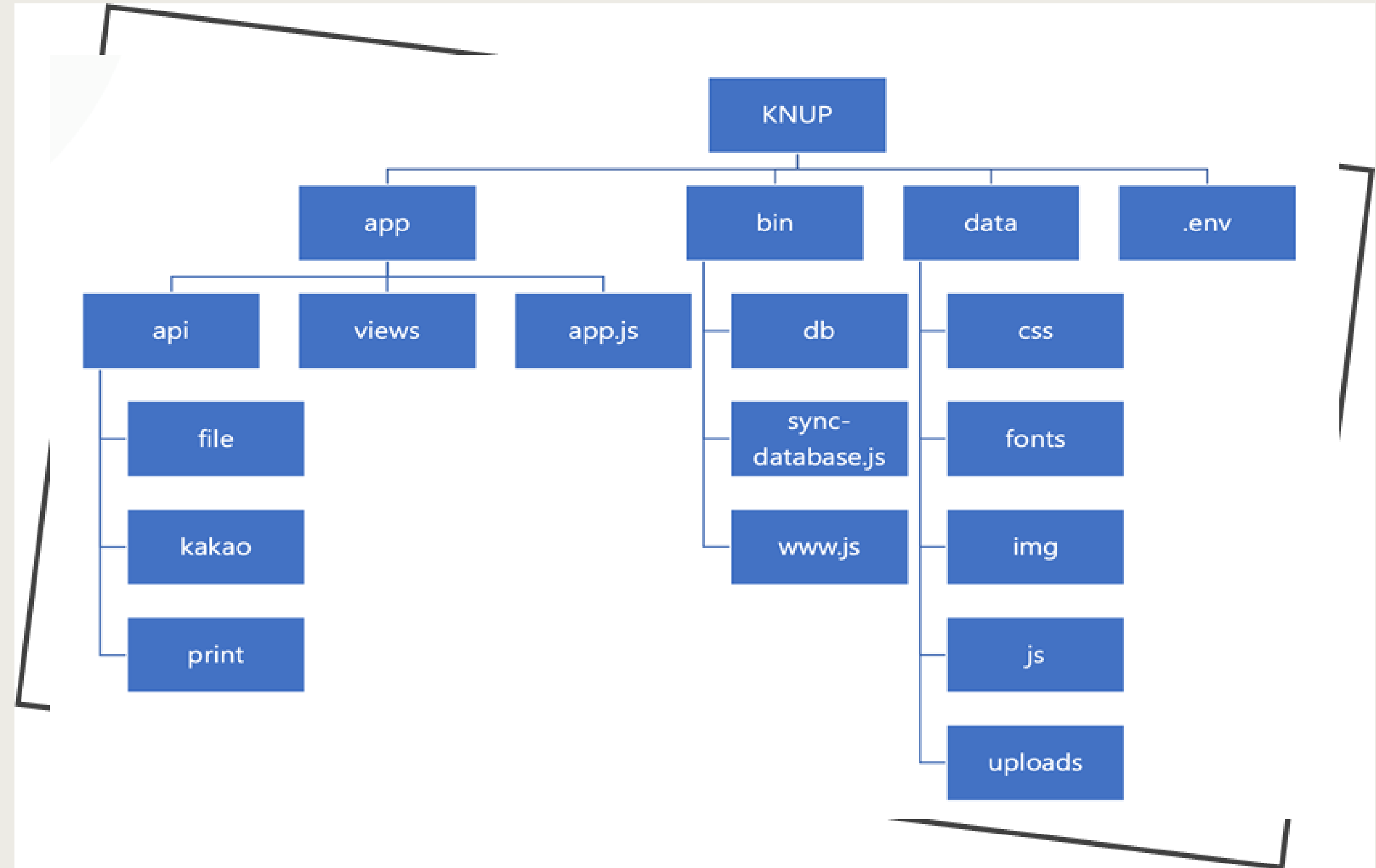
- 팀 내에서 맡은 역할
DB 담당자 (mysql)
Node js 서버
UI

GITHUB : <https://github.com/yuminee/KNUP>
동영상 : <https://youtu.be/6-P44pvRiuo>

주요 기능



디렉터리 계층 구조



감사 합니다.