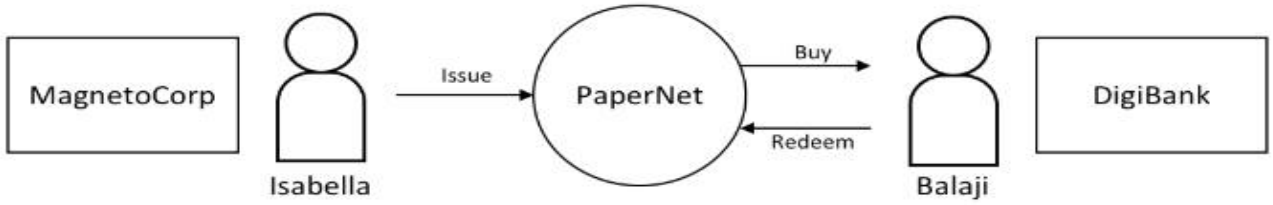
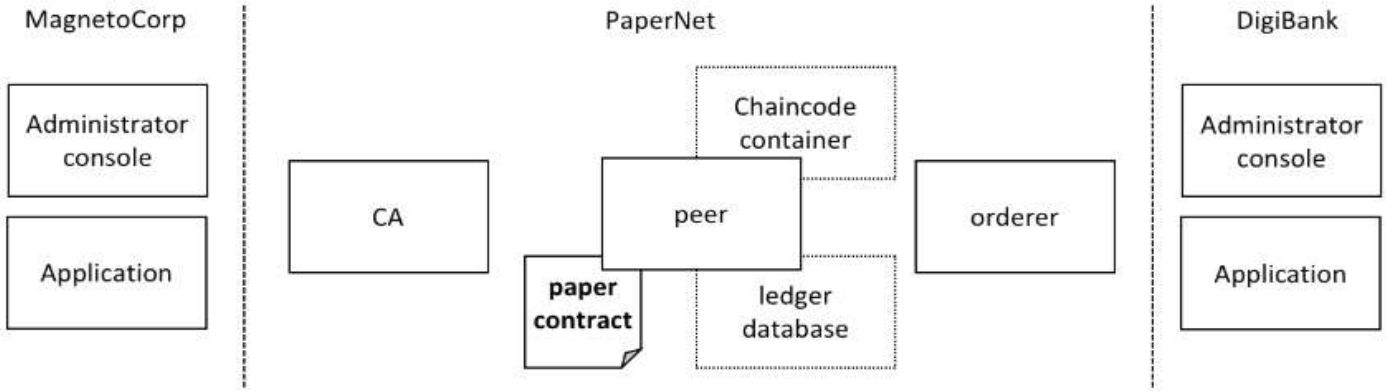
1. 시나리오

2개의 User(MagnetoCorp, DigiBank)가 있고 각각을 대표하는 Isabella, Balaji가 있음.
이들은 PaperNet을 통해 Issue, Buy, Redeem 3가지 행위를 할 수 있음.



2. PaperNet 실행



① basic-network에서

$ ./start.sh → peer, orderer, fabric-ca, couchdb 4개의 image container 생성
cli는 magetocorp와 digitbank가 다르기 때문에 각각 up 시켜줘야 함(docker-compose 이용)

② magnetocorp에서

$ cd fabric-samples/commercial-paper/organization/magnetocorp/configuration/cli

$ docker-compose –f docker-compose.yml up –d cliMagentoCorp

```
(magnetocorp admin)$ cd commercial-paper/organization/magnetocorp/configuration/cli/
(magnetocorp admin)$ docker-compose -f docker-compose.yml up -d cliMagnetoCorp

Pulling cliMagnetoCorp (hyperledger/fabric-tools:)...
latest: Pulling from hyperledger/fabric-tools
3b37166ec614: Already exists
(...)
Digest: sha256:058cff3b378c1f3ebe35d56deb7bf33171bf19b327d91b452991509b8e9c7870
Status: Downloaded newer image for hyperledger/fabric-tools:latest
Creating cliMagnetoCorp ... done
```

$ cd fabric-samples/commercial-paper/organization/magnetocorp/contract/lib
→ papercontract.js가 체인코드(issue, buy, redeem 관련 내용 포함)

$ docker exec –it cliMagentoCorp bash → 실행중인 docker container 중 –it로 command 창 실행

peer chaincode install –n papercontract –v 0 –p /opt/gopath/src/github.com/contract –l node

peer chaincode instantiate –n papercontract –v 0 –l node –c '{"Args":["org.papernet.commercialpaper:instantiate"]}' -C mychannel

→ –P 옵션을 통해 endorsement policy를 설정하는데 유효하던, 무효하던 모든 거리는 blockchain ledger에 기록되며, 이 중 유효한 tx만이 WS(couchDB)에 업데이트 됨.

$ cd fabric-samples/commercial-paper/organization/magnetocorp/application
→ addToWallet.js는 Isabella이 자신의 신원정보를 지갑에 넣는 것과 관련된 코드
→ issue.js는 신원을 사용해 00001이라는 paper를 발행하는 예시코드

$ npm install → js-yaml 등 다양한 패키지들을 사용하기 위해 설치(시간 약 1분 소요)
→ 만약 권환 관련 error 날 경우? sudo chown -R userID:userID '/home/userID/.npm/'
$ node addToWallet.js → 'done' 뜨며 isabella의 wallet(신원) 생성

```
project-b@projectb-VirtualBox:~/fabric-samples/commercial-paper/organization/magnetocorp$ tree identity/
identity/
└── user
    └── isabella
        └── wallet
            └── User1@org1.example.com
                ├── User1@org1.example.com
                ├── c75bd6911aca808941c3557ee7c97e90f3952e379497dc55eb903f31b50abc83-priv
                └── c75bd6911aca808941c3557ee7c97e90f3952e379497dc55eb903f31b50abc83-pub
```

$ node issue.js → 체인코드 중 issue 관련 샘플 진행(발급)

```
project-b@projectb-VirtualBox:~/fabric-samples/commercial-paper/organization/magnetocorp/application$ node issue.js
Connect to Fabric gateway.
Use network channel: mychannel.
Use org.papernet.commercialpaper smart contract.
Submit commercial paper issue transaction.
2019-03-19T11:59:27.633Z - info: [TransactionEventHandler]: _strategySuccess: strategy success for transaction "0b5ec57e8b57c04a9e07ad3bf851b9707ba630f62b17a8156befa227ede5920d"
Process issue transaction response.
MagnetoCorp commercial paper : 00001 successfully issued for value 5000000
Transaction complete.
Disconnect from Fabric gateway.
Issue program complete.
```

③ digibank에서
$ cd fabric-samples/commercial-paper/organization/digibank/configuration/cli
$ docker-compose –f docker-compose.yml up –d cliDigiBank
$ cd fabric-samples/commercial-paper/organization/digibank/application
→ addToWallet.js는 balaji가 자신의 신원정보를 지갑에 넣는 것과 관련된 코드
→ buy.js는 신원을 사용해 00001이라는 paper를 구매하는 예시코드
→ redeem.js는 신원을 사용해 00001이라는 paper를 반납하는 예시코드
→ chaincode가 많은 peer에 install되어 있을지라도, instantiate는 발행할 때 한 번만 필요하기 때문에 digibank에서는 위의 install/instantiate 절차 생략 가능.
$ npm install → js-yaml 등 다양한 패키지들을 사용하기 위해 설치(시간 약 3분 소요)
$ node addToWallet.js → 'done' 뜨며 balaji의 wallet(신원) 생성

```
project-b@projectb-VirtualBox:~/fabric-samples/commercial-paper/organization/digibank$ tree identity/
identity/
└── user
    └── balaji
        └── wallet
            └── Admin@org1.example.com
                ├── Admin@org1.example.com
                ├── cd96d5260ad4757551ed4a5a991e62130f8008a0bf996e4e4b84cd097a747fec-priv
                └── cd96d5260ad4757551ed4a5a991e62130f8008a0bf996e4e4b84cd097a747fec-pub
```

$ node buy.js → 체인코드 중 buy 관련 샘플 진행(구매)

```
project-b@projectb-VirtualBox:~/fabric-samples/commercial-paper/organization/digibank/application$ node buy.js
Connect to Fabric gateway.
Use network channel: mychannel.
Use org.papernet.commercialpaper smart contract.
Submit commercial paper buy transaction.
2019-03-19T12:33:48.758Z - info: [TransactionEventHandler]: _strategySuccess: strategy success for transaction "2e6340183dccd7c9179936cfd6afe3efda398c09f2f105f91ec60fa955f0752d"
Process buy transaction response.
MagnetoCorp commercial paper : 00001 successfully purchased by DigiBank
Transaction complete.
Disconnect from Fabric gateway.
Buy program complete.
```

$ node redeem.js → 체인코드 중 redeem 관련 샘플 진행(반납)

```
project-b@projectb-VirtualBox:~/fabric-samples/commercial-paper/organization/digibank/application$ node redeem.js
Connect to Fabric gateway.
Use network channel: mychannel.
Use org.papernet.commercialpaper smart contract.
Submit commercial paper redeem transaction.
2019-03-19T12:36:59.905Z - info: [TransactionEventHandler]: _strategySuccess: strategy success for transaction "a36bce7fdc2da87bf10effb8a89835a02c25000e6b87cd48df1b212f0d961e73"
Process redeem transaction response.
MagnetoCorp commercial paper : 00001 successfully redeemed with MagnetoCorp
Transaction complete.
Disconnect from Fabric gateway.
Redeem program complete.
```

3. 참고사항

① 1.4에 새로 추가된 개념? Gateway

- app을 대신해 network 상호 작용을 관리
- app이 gateway에 연결만 해주면 모든 후속 작업은 해당 게이트웨이의 구성을 사용해 관리
- 추상화(abstract)를 이용해 코딩

② Wallet(지갑) : 지갑에는 현금이나 토큰은 없고, 신분을 가진다.

Wallet

Towards the top of `issue.js`, you'll see two Fabric classes are brought into scope:

```
const { FileSystemWallet, Gateway } = require('fabric-network');
```

You can read about the `fabric-network` classes in the node SDK documentation, but for now, let's see how they are used to connect MagnetoCorp's application to PaperNet. The application uses the Fabric **Wallet** class as follows:

```
const wallet = new FileSystemWallet('../identity/user/isabella/wallet');
```

See how `wallet` locates a wallet in the local filesystem. The identity retrieved from the wallet is clearly for a user called Isabella, who is using the `issue` application. The wallet holds a set of identities – X.509 digital certificates – which can be used to access PaperNet or any other Fabric network. If you run the tutorial, and look in this directory, you'll see the identity credentials for Isabella.

Think of a wallet holding the digital equivalents of your government ID, driving license or ATM card. The X.509 digital certificates within it will associate the holder with a organization, thereby entitling them to rights in a network channel. For example, `Isabella` might be an administrator in MagnetoCorp, and this could give her more privileges than a different user – `Balaji` from DigiBank. Moreover, a smart contract can retrieve this identity during smart contract processing using the transaction context.

Note also that wallets don't hold any form of cash or tokens – they hold identities.

③ nodejs 버전 업그레이드 방법 참고 사이트

- https://d2fault.github.io/2018/04/30/20180430-install-and-upgrade-nodejs-or-npm/