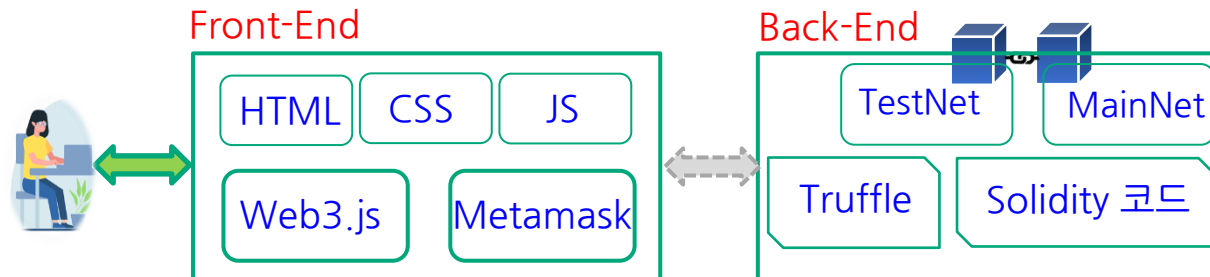

솔리더티 기반의 DApp 시연

솔리더티 기반의 DApp 시연

- ❑ Truffle Framework는 솔리더티 코드(스마트 컨트랙트)를 로컬 환경에서 보다 쉽게 컴파일하고 배포할 수 있는 프레임워크
- ❑ DApp의 구현을 위한 개발 환경
 - 스마트 컨트랙트를 작성하고 배포하는 백엔드 사이드
 - 사용자 인터페이스를 구현하는 프론트엔트 사이드
 - Web3.js: 이더리움 자바스크립트 API
 - Metatmask: 크롬 확장 프로그램으로 동작하는 이더리움 라이트 클라이언트



DApp 시연

❑ DApp 시연 시나리오

- ① Geth 실행
- ② 환경설정: 디렉터리(BankApp, solidity) 생성 및 Truffle 설치
- ③ truffle-config.js의 network 부분 수정
- ④ Bank.sol 파일 수정
- ⑤ 2_deploy_Bank.js 파일 생성
- ⑥ 웹 프로그래밍: web3 module 설치, Browserify 설치
- ⑦ DApp 실행 후 토큰 전송

Bank

From Account info: [0xA185E588db530e69fa9b8BBc592058E647fa997] Balance : 983

To Account info: [0x1d8864D12A933d5aA5615766753b52f9491477d7]

Value: [1] PassWord: [pass00] **Send**

Log

transfer successtransfer success

Count	To	Amount	Time
1	0x1d8864D12A933d5aA5615766753b52f9491477d7	12	Mon Oct 04 2021 23:33:07 GMT+0900 (한국 표준시)
2	0x1d8864D12A933d5aA5615766753b52f9491477d7	1	Tue Oct 05 2021 00:46:39 GMT+0900 (한국 표준시)

Geth 실행

```
cmd >> geth --networkid 1004 --datadir myGeth --nodiscover --port 30303 --http --http.port "8545" --http.addr "0.0.0.0" --http.corsdomain "*" --http.api "eth, net, web3, miner, debug, personal, rpc" --ws --ws.port 8546 --ws.origins "*" --ws.api "admin,eth,debug,miner,net,txpool,personal,web3" --allow-insecure-unlock --mine --miner.threads 1 --allow-insecure-unlock console
```

- ※ --networkid를 이용해 생성된 블록체인네트워크의 id를 지정해준다.
- ※ truffle을 사용해 contract를 배포와 외부 접근을 위해 http를 사용한다.
- ※ Smart contract에서 event를 발행 시켜 사용하려면 websocket을 사용해야 한다.
- ※ --allow-insecure-unlock를 사용해 계정 암호 해제 허용한다.
- ※ --mine을 사용해 geth가 시작할 때 채굴을 자동으로 시작해주는 옵션이며 miner.threads 1 를 사용해 thread의 수를 1로 지정한다.
- ※ console를 사용해 콘솔 사용한다.

관리자: 명령 프롬프트

```
C:\WINDOWS\system32>geth --networkid 1004 --datadir myGeth --nodiscover --port 30303 --http --http.port "8545" --http.addr "0.0.0.0" --http.corsdomain "*" --http.api "eth, net, web3, miner, debug, personal, rpc" --ws --ws.port 8546 --ws.origins "" --ws.api --allow-insecure-unlock --mine --miner.threads 1 --allow-insecure-unlock console
```

디렉터리 생성 및 Truffle 설치

- 환경 설정: cd를 이용해 바탕화면으로 설정한다.

cmd >> mkdir BankApp

cmd >> cd BankApp

cmd >> mkdir solidity

cmd >> npm init -y //Node.js의 기본 패키지 관리자

cmd >> cd solidity

cmd >> npm install -g truffle

cmd >> truffle init //contracts 홀더 등이 만들어짐

관리자: 명령 프롬프트

```
C:\Users\₩\Desktop>mkdir BankApp
C:\Users\₩\Desktop>cd BankApp
C:\Users\₩\Desktop\BankApp>mkdir solidity
C:\Users\₩\Desktop\BankApp>npm init -y
Wrote to C:\Users\₩\Desktop\BankApp\package.json:
```

```
{
  "name": "BankApp",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

package.json

```
C:\Users\₩\Desktop\BankApp>cd solidity
C:\Users\₩\Desktop\BankApp\solidity>npm install -g truffle
```

관리자: 명령 프롬프트

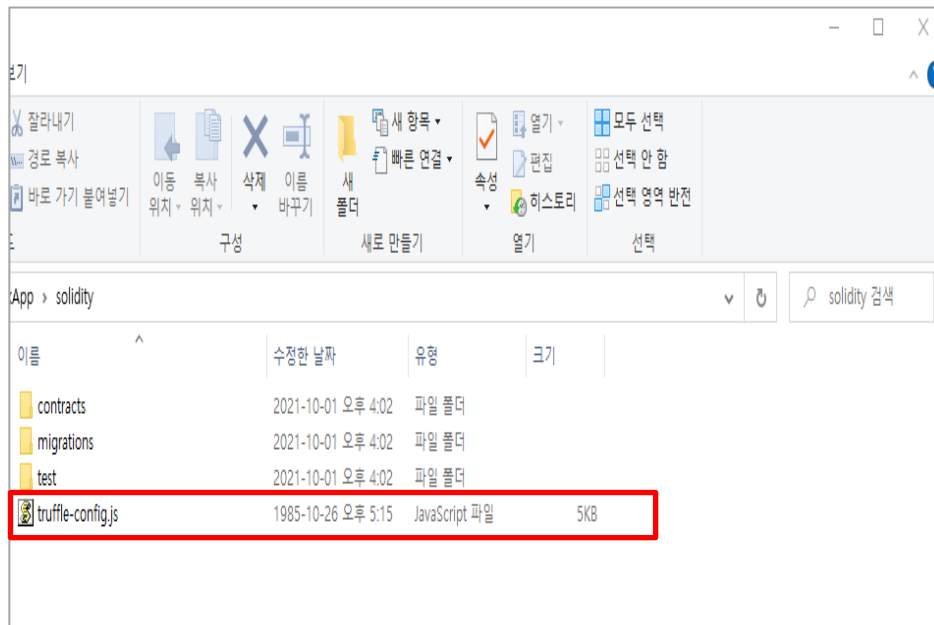
```
C:\Users\₩\Desktop\BankApp\solidity>truffle init
Starting init...
=====
> Copying project files to C:\Users\₩\Desktop\BankApp\solidity
Init successful, sweet!

Try our scaffold commands to get started:
  $ truffle create contract YourContractName # scaffold a contract
  $ truffle create test YourTestName        # scaffold a test

http://trufflesuite.com/docs
```

truffle-config.js 수정

- ❑ solidity 폴더에 있는 truffle-config.js를 열어 network 부분을 수정



```
// See this link for help: https://truffleframework.com/docs/truffle/reference/configuration.html#network-configuration
// if it's defined here and no other network is specified at the command line.
// You should run a client (like ganache-cli, geth or parity) in a separate terminal
// tab if you use this network and you must also set the 'host', 'port' and 'network_id'
// options below to some value.
//
development: {
  host: "127.0.0.1", // Localhost (default: none)
  port: 8545,       // Standard Ethereum port (default: none)
  network_id: "*",  // Any network (default: none)
  gas: 3000000
}

// Another network with more advanced options...
// advanced: {
//   port: 8777,      // Custom port
//   network_id: 1342, // Custom network
//   gas: 8500000,    // Gas sent with each transaction (default: ~6700000)
//   gasPrice: 20000000000, // 20 gwei (in wei) (default: 100 gwei)
```

Bank.sol 파일 수정(1/2)

❑ cmd >> truffle create contract Bank 입력 → contracts 폴더의 Bank.sol 파일 수정

```
C:\Windows\System32\cmd.exe
C:\Users\# Desktop\#새 폴더 (5)\# solidity>truffle create contract Bank
```

폴더 (5) > bank2 > solidity > contracts	
이름	수정된 날짜
Bank.sol	2021-10-02 오전 2:38

```
① pragma solidity >=0.5.0 <0.8.0; //solidity 0.5.0 버전을 기반으로 작성, 0.8.0까지 정상 동작
② contract Bank { //스마트 컨트랙트 정의
③     struct bankInfo { //struct 생성
④         address from;
⑤         address to;
⑥         uint256 amount;
⑦         string Time;
⑧     }
⑨     string[] public bankList;
⑩     mapping(string => bankInfo) ba;
⑪         // "키 - 값" 구조로 데이터를 저장할 때 활용되는 타입, 제공된 키(Key)를 가지고 값(value)을 얻어낼 수 있습니다.
⑫         // 변수명 ba에 key값을 부르면 해당 key값에 해당하는 정보를 불러옴
⑬         // 이더리움 기반 토큰의 핵심
⑭     event SetBankEvent( //이벤트 선언(특정 이벤트가 발생하면 블록체인 상에서 행동을 취함),
⑮         address indexed from, //프론트엔드에 전달할 데이터를 정의
⑯         address indexed to, uint256 indexed amount, string Time
⑰     );
```

Bank.sol 파일 수정(2/2)

```
① function setBank( //매개변수 안에 있는 값들을 받음
②     address _from, address _to, uint256 _amount,
③     string memory _Time    //임시 저장되는 변수, 컨트랙트 종료 시 없어짐.
④ ) public {
⑤     bankList.push(_Time); //배열 마지막에 원소 추가
⑥     ba[_Time] = bankInfo(_from, _to, _amount, _Time);
⑦     emit SetBankEvent(_from, _to, _amount, _Time); //트랜잭션 로그에 이벤트 데이터를 집어넣기 위해서 emit사용
⑧ }
⑨ }
```


Node.js 설치 확인

❑ Node.js 버전 확인

cmd >> node

C:\WINDOWS\system32\cmd.exe

Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\>node -v
v14.16.0

2_deploy_Bank.js 파일 생성

- ❑ solidity 폴더 안에 있는 migrations 디렉토리 안에 2_deploy_Bank.js 파일 생성

```
2_deploy_Bank.js - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

const Migrations = artifacts.require("Bank");

module.exports = function (deployer) {
  deployer.deploy(Migrations);
};
```

폴더 (5) > bank2 > solidity > migrations

이름	수정한 날짜	유형	크기
1_initial_migration.js	1985-10-26 오후 5:15	JavaScript 파일	1KB
2_deploy_Bank.js	2021-10-01 오후 8:51	JavaScript 파일	1KB

- ❑ solidity 폴더로 이동 후 → cmd >> truffle migrate --reset

```
C:\Windows\System32\cmd.exe

C:\Users\ Desktop (5) solidity>truffle migrate --reset
```

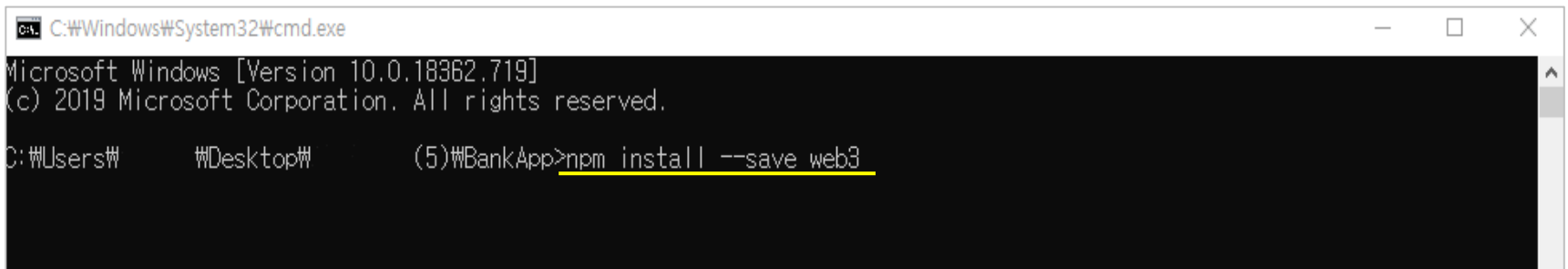
- ❑ 2_deploy_Bank.js의 contract address를 복사 → main.js 파일에 복사

```
2_deploy_Bank.js
=====
Replacing 'Bank'
-----
> transaction hash: 0x307643888c60ea5bc66285729f224bc1fa82eab36a6682b59357f7b9703c8e53
> Blocks: 0 Seconds: 12
> contract address: 0x416f0F9394159053946eA4C2a7AB45F8cC77dA84
```

Web Programming(1/3)

❑ web3 module 설치




BankApp폴더로 이동 후 web3 module을 설치하고 html, css, javascript 파일을 생성
cmd >> npm install --save web3



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18362.719]
(c) 2019 Microsoft Corporation. All rights reserved.

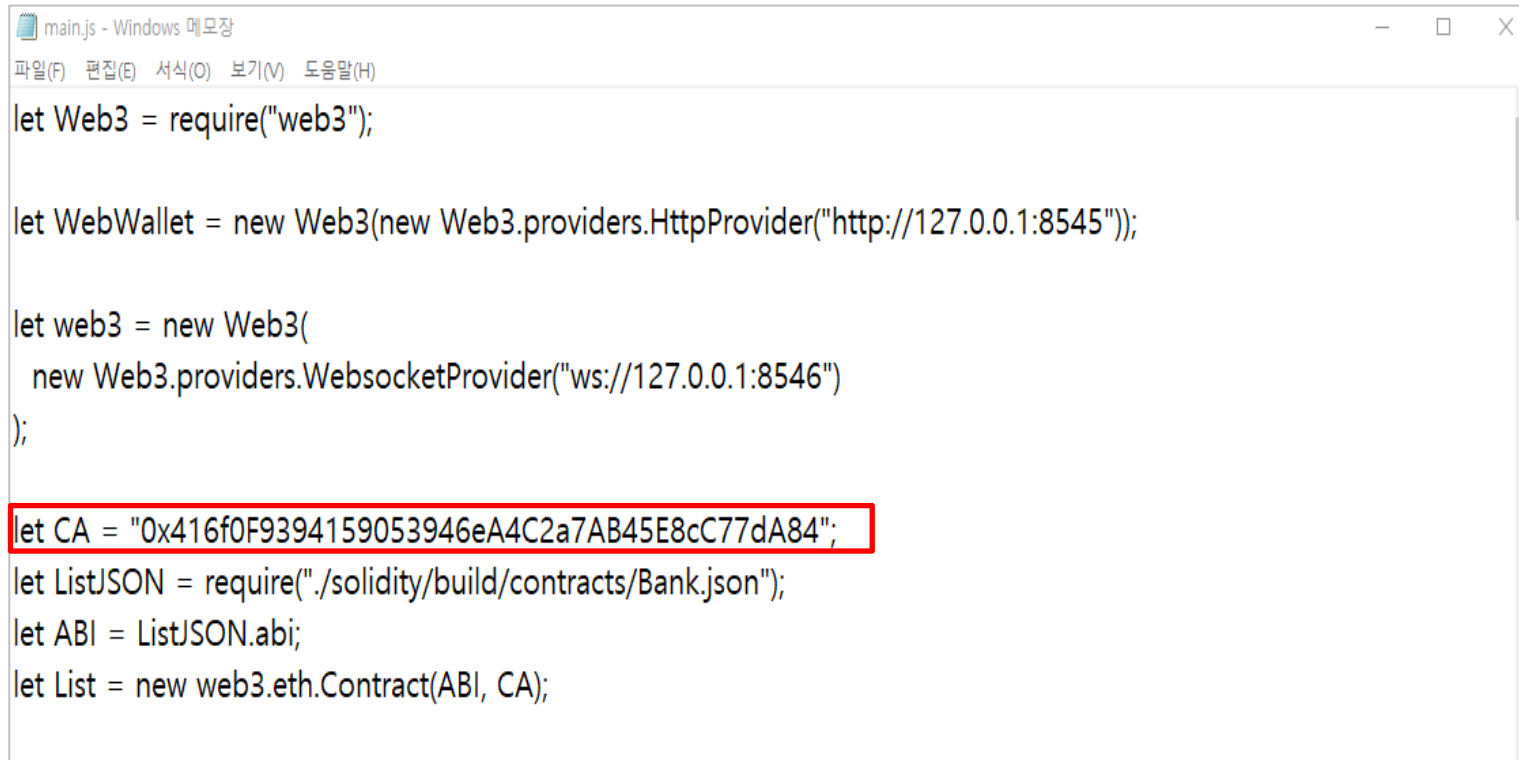
C:\Users\#\Desktop\#(5)\BankApp>npm install --save web3
```

css 파일, main 파일, index 파일을 BankApp에 위치시킨다.

 CSS.css	2021-10-01 오후 8:46	CSS 스타일시트 ...	1KB
 index.html	2021-10-01 오후 8:46	Chrome HTML D...	2KB
 main.js	2021-10-02 오전 3:25	JavaScript 파일	5KB

Web Programming(2/3)

- ❑ 앞서 복사해둔 contract address 값을 main.js 파일에 수정



```
main.js - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

let Web3 = require("web3");

let WebWallet = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:8545"));

let web3 = new Web3(
  new Web3.providers.WebsocketProvider("ws://127.0.0.1:8546")
);

let CA = "0x416f0F9394159053946eA4C2a7AB45E8cC77dA84";
let ListJSON = require("../solidity/build/contracts/Bank.json");
let ABI = ListJSON.abi;
let List = new web3.eth.Contract(ABI, CA);
```

Web Programming(3/3)

❑ Browserify 설치

브라우저에서 require()를 실행할 수 있도록 하기 위해 browserify모듈을 사용
cmd >> npm install -g browserify

```
Microsoft Windows [Version 10.0.18362.719]  
(c) 2019 Microsoft Corporation. All rights reserved.  
C:\Users\₩₩\Desktop\₩₩새 폴더 (5)\₩₩ >npm install -g browserify
```

cmd >> browserify main.js -o bundle.js

```
Microsoft Windows [Version 10.0.18362.719]  
(c) 2019 Microsoft Corporation. All rights reserved.  
C:\Users\₩₩\Desktop\₩₩새 폴더 (5)\₩₩ >browserify main.js -o bundle.js
```

DApp 실행-토큰 전송

- index 파일 실행 → 보낼 토큰의 수와 비밀번호를 입력 후 Send 버튼을 누르면 교환이 이루어지며 테이블에 거래정보가 출력된다.

새 탭

Bank

+

< > ↺ ① 파일 | C:/Users/ /Desktop/새%20폴더%20(5)%20(3)/bank2/index.html

🔍 ☆ 📌 ⋮

Bank

From

Account info:

Value: Password:

To

Account info :

Log

transfer successtransfer success