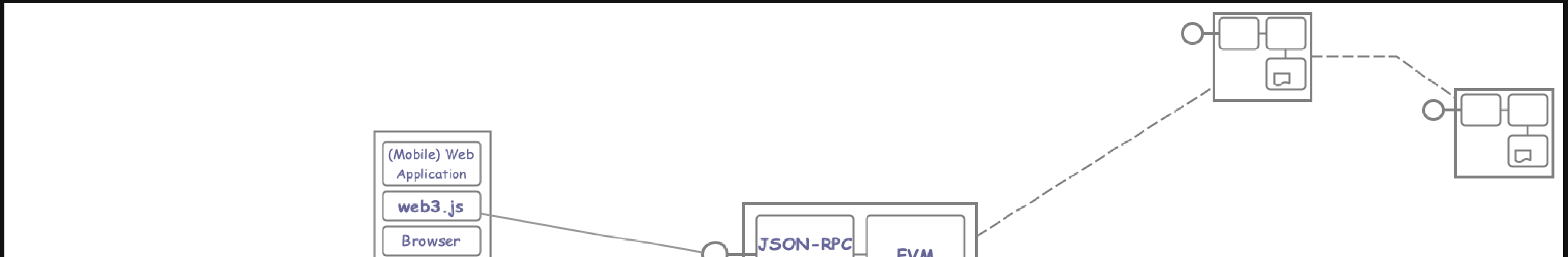
DApp 개발 기초

오상문

Feb. 2022

DApp Architecture



DApp 개발 Lifecyle

- 응용 분석/설계
- Smart Contract 구현
- Smart Contract **Audit**
- Smart Contract **Local**배포/단위테스트
- Smart Contract **Testnet**배포/단위테스트
- 응용 구현
- 응용 배포/단위테스트

DApp 개발 환경/도구

Category	Tool/Service	Remarks
Editing	Remix IDE	Web based
Build/Deploy	Truffle	JavaScript based
	Browine	Python based
Local Client	Ganache	Ganache CLI
Mainnet/Testnet Gateway	Infura	
Library	OpenZeppelin Contracts	
Block Explorer	Etherscan	Mainnet
	Etherscan/Rinkeby	
Wallet	MetaMask	

Truffle

- Installing Node.js

```
$ node --version
```

- Installing Truffle

```
$ npm ls -g truffle          # check whether or not Truffle in installed in global scope
$ npm uninstall -g truffle   # uninstalled currently installed Truffle
$ npm install -g truffle
```

- Creating Truffle Project

```
$ mkdir smart-contract-101 && cd smart-contract-101
$ truffle init
...
$ ls
/      ../     contracts/    migrations/   test/     truffle-config.js
$ cat truffle-config.js | less
...
```

Truffle Commands

```
$ truffle help
Truffle v5.4.22 - a development framework for Ethereum

Usage: truffle <command> [options]

Commands:
  build     Execute build pipeline (if configuration present)
  compile   Compile contract source files
  ...
  console   Run a console with contract abstractions and commands available
  ...
  deploy    (alias for migrate)
  ...
  networks  Show addresses for deployed contracts on each network
  ...
  test      Run JavaScript and Solidity tests
  ...
See more at http://trufflesuite.com/docs
```

```
$ truffle help migrate
...
$ truffle help test
...
```

- Resources
  - Truffle documentation
  - Truffle commands

Truffle Configuration

Prerequisite

- Setup **Node.js** project

```
$ npm init -y
$ npm install -D @truffle/hdwallet-provider
...
$ cat package.json
```

- Setup **Mnemonic**

  - macOS

```
$ env | grep BIP39_MNEMONIC
...
$ echo 'export BIP39_MNEMONIC="..."' >> ~/.profile
$ cat ~/.profile
...
```

  - Windows

```
> setx BIP39_MNEMONIC="..."
```

- Setup **Infura Project**

  - macOS

```
$ env | grep INFURA_PROJECT_ID
...
$ echo 'export INFURA_PROJECT_ID=...' >> ~/.profile
$ cat ~/.profile
...
```

  - Windows

```
> setx INFURA_PROJECT_ID=...
```

  - Resources
    - Getting Started With Infura

Truffle Configuration

- `truffle-config.js`

```js
const HDWalletProvider = require("@truffle/hdwallet-provider");
const mnemonic = process.env.BIP39_MNEMONIC;

module.exports = {
  networks: {
    development: {
      host: '127.0.0.1',
      port: 8545,
      network_id: 2016,
      gas: 3E8,
      gasPrice: 0,
      websockets: false
    },

    mainnet: {
      provider: () => new HDWalletProvider(
        mnemonic, "https://mainnet.infura.io/v3/" + process.env.INFURA_PROJECT_ID),
      network_id: '1'
    },

    rinkeby: {
      provider: () => new HDWalletProvider(
        mnemonic, "https://rinkeby.infura.io/v3/" + process.env.INFURA_PROJECT_ID),
      network_id: '4',
    }

  }
}
```

- Resources
  - Truffle configuration reference
  - `@truffle/hdwallet-provider`
  - BIP-32 : Hierarchical Deterministic Wallets(HD Wallets
  - BIP-39 : Mnemonic code for generating deterministic keys
  - Ethereum 201: HD Wallets

Testnets

Network(Chain)	Chain ID	Consensus	Avg. Block Time	Explorer
Mainnet	1	PoW	15 min.	Etherscan
Ropsten	3	PoW	30 sec.	Etherscan/Ropsten
Rinkeby	4	PoA	15 sec.	Etherscan/Rinkeby
Kovan	42	PoA	4 sec.	Etherscan/Kovan

- Resources
  - Chainlist
  - EIP-155: Simple replay attack protection

Faucets

Network	Faucet
Ropsten	https://faucet.ropsten.be/
Rinkeby	https://faucet.rinkeby.io/
	https://faucets.chain.link/rinkeby
Kovan	https://faucet.kovan.network/
	https://ethdrop.dev/

Truffle Console / Rinkeby

```
$ truffle console --network rinkeby
...
truffle(rinkeby)>
truffle(rinkeby)> web3.version
'1.5.3'
truffle(rinkeby)> web3.eth.net.getId()
4
truffle(rinkeby)> web3.eth.getBlockNumber().then(n => parseInt(n).toLocaleString())
'10,206,304'
truffle(rinkeby)> web3.eth.getBlock('latest')
...
truffle(rinkeby)> web3.eth.getBlock('latest').then(b => new Date(b.timestamp * 1000))
2022-02-21T10:46:03.000Z
truffle(rinkeby)> web3.eth.getBlock(0)
...
truffle(rinkeby)> web3.eth.getBlock(0).then(b => new Date(b.timestamp * 1000))
2017-04-12T14:59:06.000Z
truffle(rinkeby)> web3.eth.getCoinbase()
'0xb009cd53957c0d991cabe184e884258a1d7b77d9'
truffle(rinkeby)> web3.eth.getBalance(_).then(b => parseInt(b).toLocaleString())
'101,000,000,000,000,000'
truffle(rinkeby)> web3.eth.isMining()
false
ruffle(rinkeby)> web3.eth.net.getPeerCount()
100
truffle(rinkeby)> web3.eth.getAccounts()
...

truffle(rinkeby)> accounts
...
truffle(rinkeby)> web3.eth.getBalance(accounts[0])
'101000000000000000'
truffle(rinkeby)> web3.eth.getBalance(accounts[1])
'100000000000000000'
truffle(rinkeby)>
```

JSON-RPC and Web3.js

JSON-RPC	Description	web3.js
`eth_chainId`	the chain ID of the current connected node	`web3.eth.getChainId()`
`eth_blockNumber`	the number of most recent block	`web3.eth.getBlockNumber()`
`eth_getBlockByNumber`	information about a block by block number.	`web3.eth.getBlock()`
`web3_clientVersion`	the current client version	`web3.eth.getNodeInfo()`
`eth_coinbase`	the coinbase address to which mining rewards will go	`web3.eth.getCoinbase()`
`eth_accounts`	a list of addresses owned by client	`web3.eth.getAccounts()`
`eth_getBalance`	the balance of the account of given address	`web3.eth.getBalance()`
`eth_getTransactionCount`	the number of transactions sent from an address. (nonce)	`web3.eth.getTransactionCount()`
`eth_getCode`	code at a given address	`web3.eth.getCode()`
`eth_signTransaction`	signs a transaction can be submitted to the network at a later time.	`web3.eth.signTransaction()`
`eth_sendTransaction`	creates new message call transaction or a contract creation	`web3.eth.sendTransaction()`
`eth_sendRawTransaction`	creates new message call transaction or a contract creation for signed transactions.	`web3.eth.sendSignedTransaction()`
`eth_call`	executes a new message call immediately without creating a transaction on the block chain.	`web3.eth.call()`

- Resources
  - JSON-RPC API
  - Web3.js API
  - Web3.py API

JSON-RPC Samples

```
$ curl -sSX POST --data '{"jsonrpc":"2.0","method":"eth_chainId","params":[],"id":21}' \
  https://rinkeby.infura.io/v3/${INFURA_PROJECT_ID} | jq .
{
  "jsonrpc": "2.0",
  "id": 21,
  "result": "0x4"
}
$ curl -sSX POST --data '{"jsonrpc":"2.0","method":"eth_blockNumber","params":[],"id":22}' \
  https://rinkeby.infura.io/v3/${INFURA_PROJECT_ID} | jq .
{
  "jsonrpc": "2.0",
  "id": 22,
  "result": "0x9bced3"
}
$ curl -sSX POST --data '{"jsonrpc":"2.0","method":"eth_getBlockByNumber","params":["latest", false],"id":23}' \
  https://rinkeby.infura.io/v3/${INFURA_PROJECT_ID} | jq .
...
$ curl -sSX POST --data '{"jsonrpc":"2.0","method":"eth_getBlockByNumber","params":["0x0", false],"id":24}' \
  https://rinkeby.infura.io/v3/${INFURA_PROJECT_ID} | jq .
...
$ curl -sSX POST --data '{"jsonrpc":"2.0","method":"eth_accounts","params":[],"id":25}' \
  https://rinkeby.infura.io/v3/${INFURA_PROJECT_ID} | jq .
{
  "jsonrpc": "2.0",
  "id": 25,
  "result": []
}
$
```

- Resources
  - `jq` : `sed` for JSON data

Ganache (Ganache CLI)

Local standalone client mainly for testing

- Installing

```
$ npm install -g ganache
...
$ ganache --help
```

- Launching

```
$ ganache --chain.networkId 2016 \
  --chain.chainId 2016 \
  --server.host 127.0.0.1 \
  --server.port 8545 \
  --miner.defaultGasPrice 25000000000 \
  --miner.defaultTransactionGasLimit 400000000 \
  --miner.blockTime 0 \
  --wallet.totalAccounts 15 \
  --wallet.defaultBalance 10000 \
  --wallet.unlockedAccounts 0 1 2 3 4 \
  --database.dbPath run/ganache/data
```

- Resources
  - Ganache startup options

- Playing

```
$ truffle config get networks
{
  dashboard: {
    network_id: '*',
    networkCheckTimeout: 120000,
    url: 'http://localhost:24012/rpc',
    skipDryRun: true
  },
  development: {
    host: '127.0.0.1',
    port: 8545,
    network_id: 2016,
    gas: 300000000,
    gasPrice: 0,
    websockets: false
  },
  mainnet: { provider: [Function: provider], network_id: '1' },
  rinkeby: { provider: [Function: provider], network_id: '4' }
}
$ truffle console
truffle(development)>

...

truffle(development)> .exit
$
```

Remix IDE

Best Solidity editor ever.

- Intalling `remixd`

```
$ npm install @remix-project/remixd
...

$ npx remixd --help
```

- Launching `remixd`

```
$ npx remixd --shared-folder ./ \
  --remix-ide https://remix.ethereum.org
...
$
```

- Open https://remix.ethereum.org
  - Click **Connect to Localhost** under the **File** section in the main pannel.
  - In the **File Explorers** on the left pannel, Click mouse right button on ***`contracts`*** directory, select **New Folder** menu to create ***`cryptopunks`*** directory under it.
  - Click mouse right button on ***`contracts/cryptopunks`*** directory select **New Folder** menu to create ***`CryptoPunksMarket.sol`*** file under it.
  - Copy the source from *`https://github.com/larvalabs/cryptopunks/blob/master/contracts/CryptoPunksMarket.sol`* and paste it into the ***`contracts/cryptopunks/CryptoPunksMarket.sol`*** file.
  - Change left pannel to **Solidity Compiler** by clicking in the leftmost bar
  - In the **Solidity Compiler** on the left pannel, click **Compile CryptoPunksMarket.sol** button to compile the contract source.

Sample Contract (1/4)

https://github.com/larvalabs/cryptopunks/blob/master/contracts/CryptoPunksMarket.sol

```
pragma solidity ^0.4.8;
contract CryptoPunksMarket {

    // You can use this hash to verify the image file containing all the punks
    string public imageHash = "ac39af4793119ee46bbff351d8cb6b5f23da60222126add4268e261199a2921b";

    address owner;

    string public standard = 'CryptoPunks';
    string public name;
    string public symbol;
    uint8 public decimals;
    uint256 public totalSupply;

    uint public nextPunkIndexToAssign = 0;

    bool public allPunksAssigned = false;
    uint public punksRemainingToAssign = 0;

    //mapping (address => uint) public addressToPunkIndex;
    mapping (uint => address) public punkIndexToAddress;

    /* This creates an array with all balances */
    mapping (address => uint256) public balanceOf;

    ...
```

- Resources
  - Structure of Contract
    - State variables
    - Events
    - Functions
  - Data types
    - Boolean
    - Integer
    - Address
    - Fixed-sized byte array
    - Dynamically-sized byte array
    - String
    - Enum
    - Fixed-sized array
    - Dynamically-sided array
    - Mapping
  - Visibility

Sample Contract (2/4)

https://github.com/larvalabs/cryptopunks/blob/master/contracts/CryptoPunksMarket.sol

```
...

struct Offer {
    bool isForSale;
    uint punkIndex;
    address seller;
    uint minValue;          // in ether
    address onlySellTo;     // specify to sell only to a specific person
}

struct Bid {
    bool hasBid;
    uint punkIndex;
    address bidder;
    uint value;
}

// A record of punks that are offered for sale at a specific minimum value, and perhaps to a specific
mapping (uint => Offer) public punksOfferedForSale;

// A record of the highest punk bid
mapping (uint => Bid) public punkBids;

mapping (address => uint) public pendingWithdrawals;

...
```

- Resources
  - Structs
  - Mapping Types

Sample Contract (3/4)

https://github.com/larvalabs/cryptopunks/blob/master/contracts/CryptoPunksMarket.sol

```
    event Assign(address indexed to, uint256 punkIndex);
    event Transfer(address indexed from, address indexed to, uint256 value);
    event PunkTransfer(address indexed from, address indexed to, uint256 punkIndex);
    event PunkOffered(uint indexed punkIndex, uint minValue, address indexed toAddress);
    event PunkBidEntered(uint indexed punkIndex, uint value, address indexed fromAddress);
    event PunkBidWithdrawn(uint indexed punkIndex, uint value, address indexed fromAddress);
    event PunkBought(uint indexed punkIndex, uint value, address indexed fromAddress, address indexed toAd
    event PunkNoLongerForSale(uint indexed punkIndex);

    /* Initializes contract with initial supply tokens to the creator of the contract */
    function CryptoPunksMarket() payable {
        //        balanceOf[msg.sender] = initialSupply;              // Give the creator all initial toke
        owner = msg.sender;
        totalSupply = 10000;                        // Update total supply
        punksRemainingToAssign = totalSupply;
        name = "CRYPTOPUNKS";                                    // Set the name for display purposes
        symbol = "Ͼ";                               // Set the symbol for display purposes
        decimals = 0;                                       // Amount of decimals for display purposes
    }

    function setInitialOwner(address to, uint punkIndex) {
        if (msg.sender != owner) throw;
        if (allPunksAssigned) throw;
        if (punkIndex >= 10000) throw;
        ...
    }

    ...
```

Sample Contract (4/4)

https://github.com/larvalabs/cryptopunks/blob/master/contracts/CryptoPunksMarket.sol

```
  ...

  function setInitialOwners(address[] addresses, uint[] indices) {
    ...
  }

  function allInitialOwnersAssigned() {
    ...
  }

  function getPunk(uint punkIndex) {
    ...
  }

  function transferPunk(address to, uint punkIndex) {
    ...
  }

  function punkNoLongerForSale(uint punkIndex) {
    ...
  }

  function offerPunkForSale(uint punkIndex, uint minSalePriceInWei) {
    ...
  }


  ...
```

```
  ...

  function offerPunkForSaleToAddress(uint punkIndex,
                uint minSalePriceInWei, address toAddress) {
    ...
  }

  function buyPunk(uint punkIndex) payable {
    ...
  }

  function withdraw() {
    ...
  }

  function enterBidForPunk(uint punkIndex) payable {
    ...
  }

  function acceptBidForPunk(uint punkIndex, uint minPrice) {
    ...
  }

  function withdrawBidForPunk(uint punkIndex) {
    ...
  }

}
```

Solidity Features

- ✓ C, Java, JavaScript like syntax

  Curly-brace block

- ✓ Statically typed

  Compile-time type safety

- ✓ Imperative and object-oriented

  Support ``` `interface` ```, ``` `abstract contract` ```, multiple ``` `inheritance` ```

- ✓ Radicall growing

  Breaking change in every major version upgrade from `v0.5.0`(Nov 2018) to `v0.8.0`(Dec 2020)

- ✓ Runs on EVM[Ethereum Virtual Machine]

  Compiled into bytecode and executed as a number of EVM opcodes.

- Resources
  - https://github.com/ethereum/solidity
  - Inheritance
  - Interfaces
  - Abstract Contracts
  - EVM
  - EVM Opcodes
  - Ethereum Yello Paper

Solidity Types (1/2)

Type	Keyword	Operators	Fields/Methods	Literal
Boolean	`bool`	`!`, `&&`, `||`, `==`, `!=`		
Unsigned Integer	`uint8`, `uint16`, `uint24`, …, `uint248`, `uint256`, `uint`	`<`, `<=`, `==`, `>=`, `>`, `&`, `|`, `^`, `~`, `<<`, `>>`, `+`, `-`, `*`, `/`, `%`, `**`		`100`, `0x2eff`, `300_000_000`, `2e10`, `2.1e10`
Signed Integer	`int8`, `int16`, `int24`, …, `int248`, `int256`, `int`	`<`, `<=`, `==`, `>=`, `>`, `&`, `|`, `^`, `~`, `<<`, `>>`, `+`, `-`, `*`, `/`, `%`, `**`		
Address	`address`		`balance`, `code`, `call()`, `delegatecall()`, `staticcall()`	`0xdCad3a6d3569DF655070DEd06cb7A1b2Ccd1D3AF`
Address Payable	`address payable`		`balance`, `code`, `call()`, `delegatecall()`, `staticcall()`, `transfer()`, `send()`	
Fixed-sized Byte Array	`byte1`, `byte2`, `byte3`, … `byte31`, `byte32`	`<`, `<=`, `==`, `>=`, `>`, `&`, `|`, `^`, `~`, `<<`, `>>`, `x[k]`	`length`	

Solidity Types (2/2)

Type	Keyword	Operators	Fields/Methods	Literal
(Dynamically-sized) Byte Array	`bytes`	`x[k]`	`push()`, `push(x)`, `pop()`, `bytes(string)`, `concat()`	
String	`string`		`concat()`	`'foo'`, `"foo"`, `'foo\nbar'`, `'foo\\bar'`
Array	`T[n]`, `T[]`, `T[n][m]`, `T[n][]`, `T[][m]`, `T[][]`	`a[k]`, `a[m:n]`	`length`, `push()`, `push(x)`, `pop()`	[1, 2, 3]
Mapping	`mapping(key-type => value-type)`	`m[key]`		
Struct	`struct T { … }`			

Deployment and …