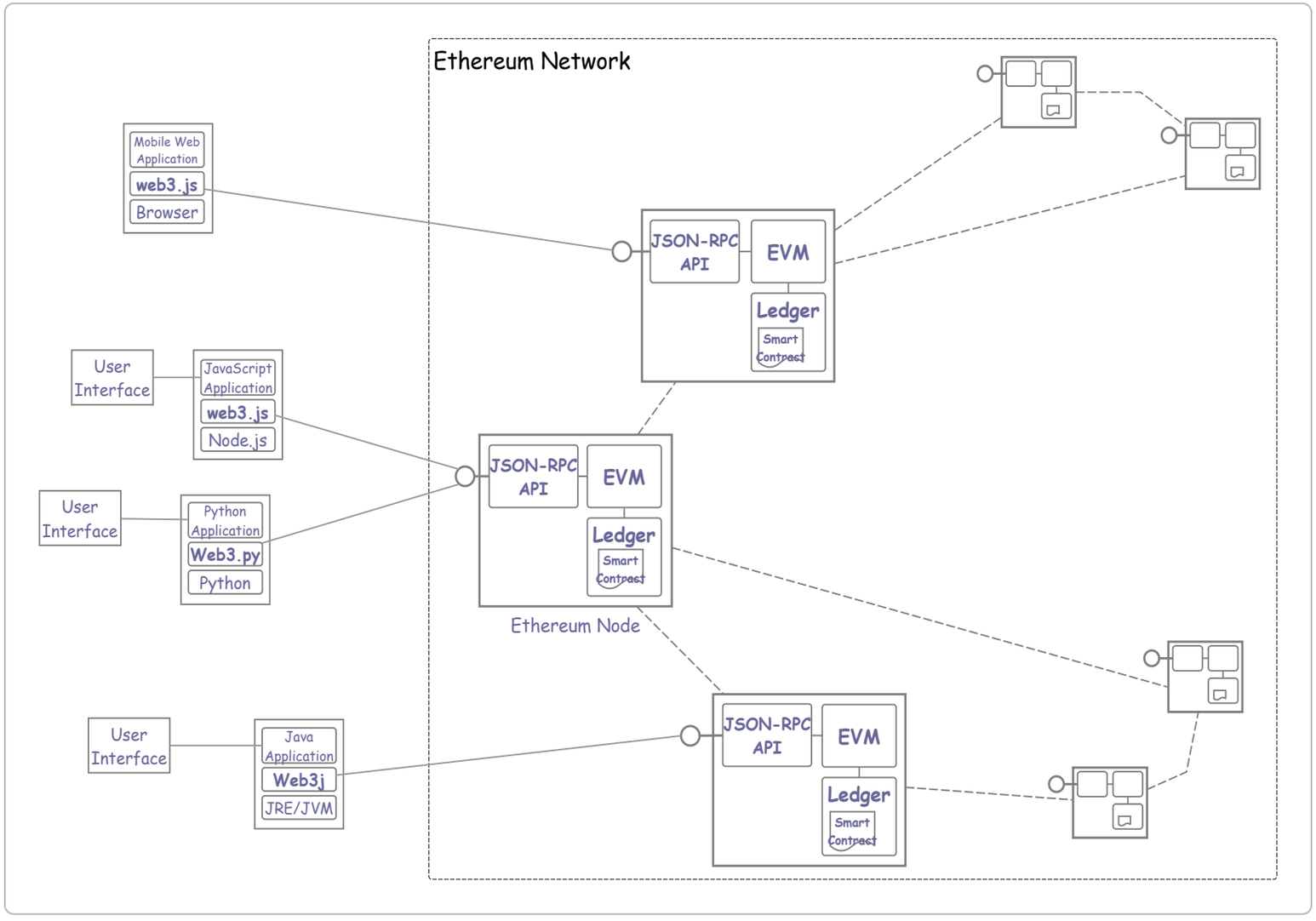
Smart Contract 개발 환경 구성

오상문

Sep. 2022

ToC

DApp Architecture



- 참고자료
  - JSON-RPC API
  - Web3.js API
  - Web3.py API
  - Introduction to the Ethereum Stack

Smart Contract 개발 환경

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

- Development Frameworks

사전준비사항 (Prerequisite)

- O/S : Windows, macOS, Linux
  - Windows의 경우 Git Bash 권장
    - 윈도우에서 Git Bash 설치하기

- `Node.js` v14.x
  - Node.js 내려받기
  - `Node.js` Releases

- NVM or NVM for Windows
  - 2개 이상 다수 version 의 Node.js 를 단일 장비에 설치하고 필요에 따라 version을 바꾸어가며 사용

Software	O/S	Guide
`NVM`	Linux, macOS	nvm 소개 및 설치 방법
`NVM for Windows`	Windows	nvm-windows, node.js 및 npm 설치 (Microsoft)

Truffle 설치

Smart contract 개발 환경 - Compile, Build, Deploy, Test

- Node.js 버전 확인

```
$ node --version
v14.19.0
```

- Truffle 설치

```
$ npm ls -g truffle         # Truffle 설치 여부 확인
...
$ npm ls -g --depth 0       # Global scope으로 설치된 모든 pacakge 확인
...
$ truffle version           # 설치된 Truffle 버전 확인
...
$ npm uninstall -g truffle  # 현재 설치된 Truffle 제거(uninstall)
...
$ npm install -g truffle
...
```

- Truffle 설치 확인

```
$ truffle version
Truffle v5.5.27 (core: 5.5.27)
Ganache v7.4.0
Solidity v0.5.16 (solc-js)
Node v14.19.0
Web3.js v1.7.4
```

Truffle Command-line 명령어

`truffle help`, `truffle help <subcommand>`

- Truffle이 제공하는 모든 명령어는 `truffle help` 명령으로 확인 가능

```
$ truffle help
Truffle v5.5.4 - a development framework for Ethereum

Usage: truffle <command> [options]

Commands:
  build      Execute build pipeline (if configuration present)
  compile    Compile contract source files
  config     Set user-level configuration options
  console    Run a console with contract abstractions and commands available
  create     Helper to create new contracts, migrations and tests
  ...
  deploy     (alias for migrate)
  develop    Open a console with a local development blockchain
  exec       Execute a JS module within this Truffle environment
  help       List all commands or provide information about a specific command
  init       Initialize new and empty Ethereum project
  install    Install a package from the Ethereum Package Registry
  migrate    Run migrations to deploy contracts
  networks   Show addresses for deployed contracts on each network
  ...
  run        Run a third-party command
  test       Run JavaScript and Solidity tests
  unbox      Download a Truffle Box, a pre-built Truffle project
  version    Show version number and exit
  watch      Watch filesystem for changes and rebuild the project automatically
```

- 개별 하위 명령어에 대한 상세 구문과 도움말은 `truffle help <subcommand>` 명령으로 확인

```
$ truffle help deploy

truffle deploy [--reset] [-f <number>] [--compile-all] [--verbose-rpc] [--network <n
(alias for migrate)

--reset
    Run all migrations from the beginning, instead of running from the last complete
--f <number>
    Run contracts from a specific migration. The number refers to the prefix of the
--to <number>
    Run contracts to a specific migration. The number refers to the prefix of the mi
--compile-all
    Compile all contracts instead of intelligently choosing which contracts need to
--compile-none
    Do not compile any contracts before migrating.
--verbose-rpc
    Log communication between Truffle and the Ethereum client.
...
--dry-run
    Only perform a test or 'dry run' migration.
--skip-dry-run
    Do not run a test or 'dry run' migration.
...
--network <name>
    Specify the network to use. Network name must exist in the configuration.
...
```

Truffle Project 생성

`truffle init`

- Truffle 은 smart contract 개발을 위한 표준 디렉토리 layout 을 정의하고 있음.
- 프로젝트 base 디렉토리 아래에 용도별로 `contracts`, `migrations`, `test`, `build` 의 하위 디렉토리를 활용하고, Truffle 환경설정은 `truffle-config.js` 파일에 지정함.

```
$ mkdir first-contracts && cd first-contracts
...
first-contracts $ truffle init

Starting init...
================
...

first-contracts $ ls -R
.:
./  ../  .gitattributes  contracts/  migrations/  test/  truffle-config.j

./contracts:
./  ../  .gitkeep

./migrations:
./  ../  .gitkeep

./test:
./  ../  .gitkeep
first-contracts $ cat truffle-config.js
...
```

```
first-contracts $ npm init -y
...

first-contracts $ cat package.json
...
```

- 참고자료
  - Truffle Commands

Truffle Compile 명령

`truffle compile`

- Contract Source 파일 생성

```
first-contracts $ touch contracts/MetaCoin.sol  # 빈 source 파일 생
first-contracts $ vi contracts/MetaCoin.sol
...
first-contracts $ cat contracts/MetaCoin.sol
...
first-contracts $
```

- Contract Compile

```
first-contracts $ truffle compile --all
...
- Downloading compiler. Attempt #1.
...
> Compiled successfully using:
   - solc: 0.8.16+commit.07a7930e.Emscripten.clang
first-contracts $
```

- Contract Compile 결과 산출물

```
first-contracts $ ls build/contracts/
./  ../  ConvertLib.json  MetaCoin.json
first-contracts $
```

- Contract Source (원본)

```
// SPDX-License-Identifier: MIT
// from 'https://github.com/truffle-box/metacoin-box'
pragma solidity ^0.8.13;

library ConvertLib {
  function convert(uint amount, uint conversionRate)
      public pure returns (uint convertedAmount) {
    return amount * conversionRate;
  }
}

contract MetaCoin {
  mapping (address => uint) balances;

  event Transfer(address indexed _from,
      address indexed _to, uint256 _value);

  constructor() { balances[tx.origin] = 10000; }

  function sendCoin(address receiver, uint amount)
      public returns(bool sufficient) {
    if (balances[msg.sender] < amount) return false;
    balances[msg.sender] -= amount;
    balances[receiver] += amount;
    emit Transfer(msg.sender, receiver, amount);
    return true;
  }

  function getBalance(address addr) public view returns(uint) {
    return balances[addr];
  }

  function getBalanceInEth(address addr) public view returns(uint) {
    return ConvertLib.convert(getBalance(addr),2);
  }
}
```

Truffle Compile 산출물

bytecode, ABI

```
first-contracts $ node        # Node.js shell (REPL) 진입
Welcome to Node.js v14.19.0.
Type ".help" for more information.
```

```
>
> a = fs.readFileSync('./build/contracts/MetaCoin.json')
...
> b = JSON.parse(a.toString())
...
> b.bytecode
'0x6080604052348015610010576000080fd5b506127106000803273ffffffffffffffff...'

> console.dir(b.abi, {depth : null})
[
  { inputs: [], stateMutability: 'nonpayable', type: 'constructor' },
  ...
  {
    inputs: [
      { internalType: 'address', name: 'receiver', type: 'address' },
      { internalType: 'uint256', name: 'amount', type: 'uint256' }
    ],
    name: 'sendCoin',
    outputs: [ { internalType: 'bool', name: 'sufficient', type: 'bool' } ],
    stateMutability: 'nonpayable',
    type: 'function'
  },
  ...
]

> .exit
first-contracts $
```

Ganache

Local standalone Ethereum simulator for testing purpose

- 설치

```
first-contracts $ npm ls -g --depth 0 ganache    # 설치 여부 확인
...
first-contracts $ npm install -g ganache         # 설치
...
first-contracts $ ganache --version              # 버전 확인
```

- 사용법

```
first-contracts $ ganache --help | less
...
```

- 참고자료
  - Ganache Startup Options

Ganache 실행

```
first-contracts $ mnemonic="army van defense carry jealous true garbage claim echo media make crunch"
first-contracts $ ganache --database.dbPath run/ganache/data -m "$mnemonic" -a 10 -n false -e 10000
...
```

```
ganache v7.4.1 (@ganache/cli: 0.5.1, @ganache/core: 0.5.1)
Starting RPC server

Available Accounts
==================
(0) 0x2161DedC3Be05B7Bb5aa16154BcbD254E9e9eb68 (10000 ETH)
(1) 0x9595F373a4eAe74511561A52998cc6fB4F9C2bdD (10000 ETH)
(2) 0x67F439f1ba85f86e1e405810675c06bC4020596D (10000 ETH)
(3) 0xf319c1A07c173800a5A3532195A8804bd90d997E (10000 ETH)
...

Private Keys
==================
(0) 0x73bf21bf06769f98dabcfac16c2f74e852da823effed12794e56876ede02d45d
(1) 0x9b1dd6e4ee4f1895e9191e626bd61081cf7f4cfe63e16024faeac73aa829cfcb
(2) 0x1b129af25984b49d1be37ddcefaccf05eefc934fe56c2529caa77e85d161d3db
(3) 0xd0ba4cd486a6d63c02c1d83b187ae1169397710572e73211c97e6b769d283adb
...

HD Wallet
==================
Mnemonic:      army van defense carry jealous true garbage claim echo media make crunch
Base HD Path:  m/44'/60'/0'/0/{account_index}

...

RPC Listening on 127.0.0.1:8545
```

Ganache 접속

- Truffle 환경설정 파일 생성

```
first-contracts $ touch truffle-config.js
first-contracts $ vi truffle-config.js
...
```

- `truffle-config.js` 내용

```
module.exports = {
  networks: {
    development: {
      network_id: "*",
      host: '127.0.0.1',
      port: 8545,
      gasPrice: 0
    }
  }
}
```

- Ganache 접속

```
first-contracts $ cat truffle-config.js
...

first-contracts $ truffle console
truffle(development)>
```

- 참고자료
  - Truffle Configuration
  - `truffle console`

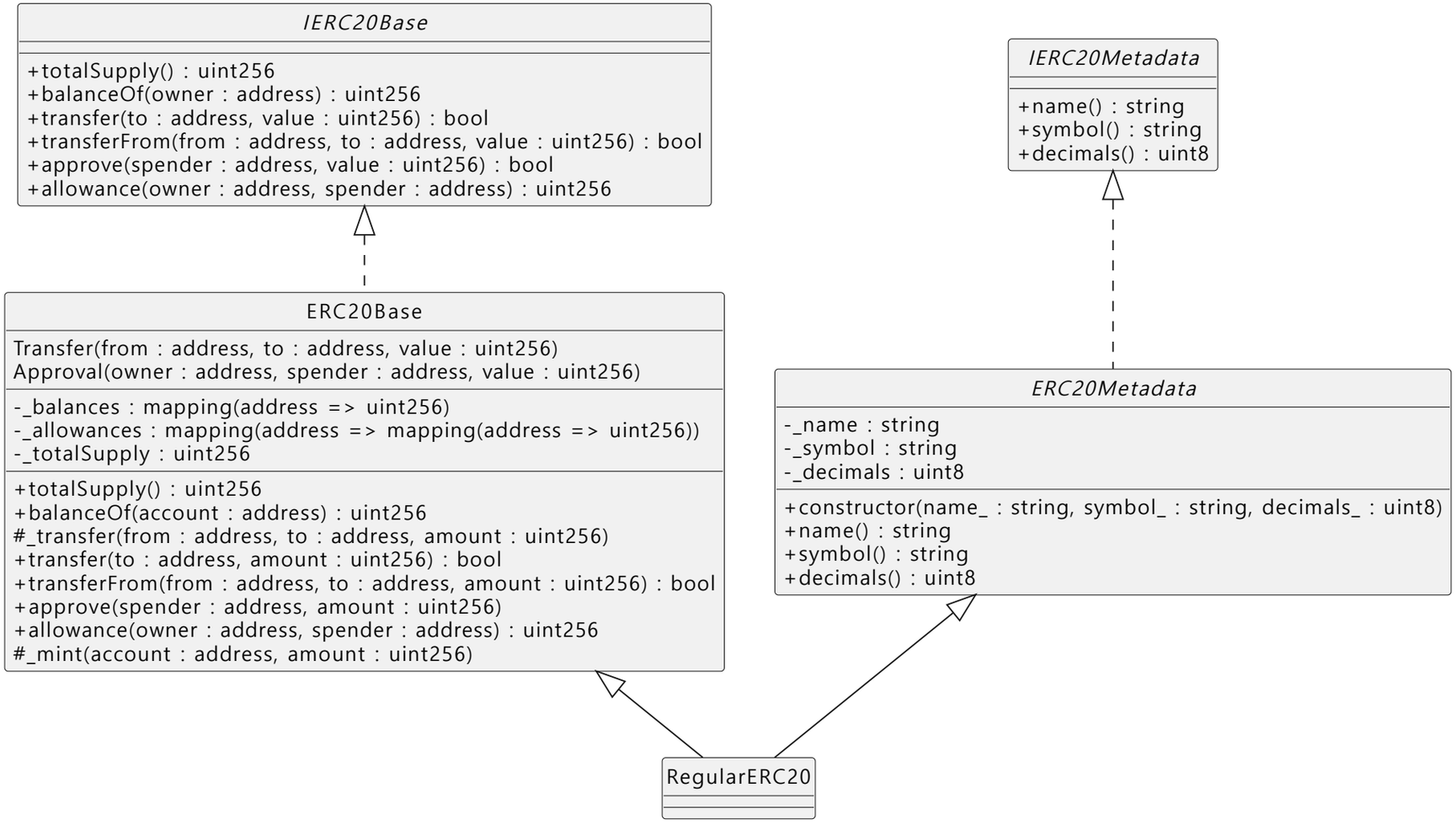
Web3.js on Ganache

```
truffle(development)> web3.eth.getNodeInfo()
'Ganache/v7.4.1/EthereumJS TestRPC/v7.4.1/ethereum-js'
truffle(development)> web3.eth.net.getPeerCount()
0
truffle(development)> web3.eth.getBlockNumber()
0
truffle(development)> web3.eth.getAccounts()
...
truffle(development)> accounts
...
truffle(development)> web3.eth.getBalance(accounts[0])
'100000000000000000000'
truffle(development)> web3.eth.getBalance(accounts[1])
'100000000000000000000'
truffle(development)> web3.eth.sign("transaction", accounts[0])
'0x79d7432c76be89f11225a1308e25435c20fab83d25af81172fd775e902a5492c5f03312b218d7ab733bf081dac8df25099ee580715df7155777a657d659baa491c'
truffle(development)> web3.eth.sign("transaction", accounts[1])
'0x67c969709886749fed8d38751e56ef4a8aeb5c877120a29ad763fcfc46e0d1825bb935594c861c30f6ac954c4c0e81597f5af4859cbb0966609c619377bb54311c'
truffle(development)> web3.eth.sendTransaction({from: accounts[0], data: "0x0"})
...
truffle(development)> web3.eth.sendTransaction({from: accounts[1], data: "0x0"})
...
truffle(development)> web3.eth.getBlockNumber()
...
truffle(development)> web3.eth.getBlock('latest')
...
truffle(development)> web3.eth.getBalance(accounts[0])
'999999...'
truffle(development)> web3.eth.getBalance(accounts[2])
'100000000000000000000'
```

Solidity

- ***Programming language for EVM***
  - Compiled into bytecode and executed as a number of EVM opcodes.
  - Ethereum Yellowpaper
- ***Object-Oriented***
  - `contract` / `class`
  - `interface`, `abstract contract`,
  - multiple inheritance, polymorphism(function overriding)
- ***Statically typed***
  - Compile-time type safety
  - `bool`, `uint<M>` (uinsined int), `int<M>` (signed int), `address`, `bytes<N>` (fixed-sized byte array)
  - `bytes` (dynamic-sized byte array), `string` (unicode string)
  - `T[M]`, `T[]`, `mapping[K => V]` (hash table)
  - `enum`, `struct`, `function`, `contract`
- ***C++, Java, JavaScript like Syntax***
  - Curly-brace block (`{` ... `}`)
  - Constrol statements
    - `if` / `else if` / `else`
    - `for`, `do`, `while`, `break`, `continue`
  - Exception handling statement : `try` / `catch`
  - Operators
    - arithmetic : `+` `-` `*` `/` `%` `**` `++` `--`
    - comparison : `==` `!=` `<` `>` `<=` `=>`
    - logical : `&&` `||` `!`,
    - bitwise : `&` `|` `^` `~` `<<` `>>`
    - assignment : `=` `+=` `-=` `*=` `/=` `%=` `<<=` `=>>`
    - tenary : `<condition> ? <if-true> : <if-false>`
  - Visibility : `external`, `public`, `internal`, `private`

Sample Contracts Model



Sample Contracts

- `interface IERC20Metadata`

```
pragma solidity ^0.8.0;

interface IERC20Metadata {

    function name() external view returns (string memory);
    function symbol() external view returns (string memory);
    function decimals() external view returns (uint8);
}
```

- `contract ERC20Metadta`

```
pragma solidity ^0.8.0;
import "./IERC20Metadata.sol";

abstract contract ERC20Metadata is IERC20Metadata {
    string private _name;
    string private _symbol;
    uint8 private _decimals;

    constructor(string memory name_, string memory symbol_, uint8 decimals_){
        _name = name_;
        _symbol = symbol_;
        _decimals = decimals_;
    }

    function name() public view override returns (string memory){
        return _name;
    }

    function symbol() public view virtual override returns (string memory){
        return _symbol;
    }

    function decimals() public view virtual override returns (uint8){
        return _decimals;
    }
}
```