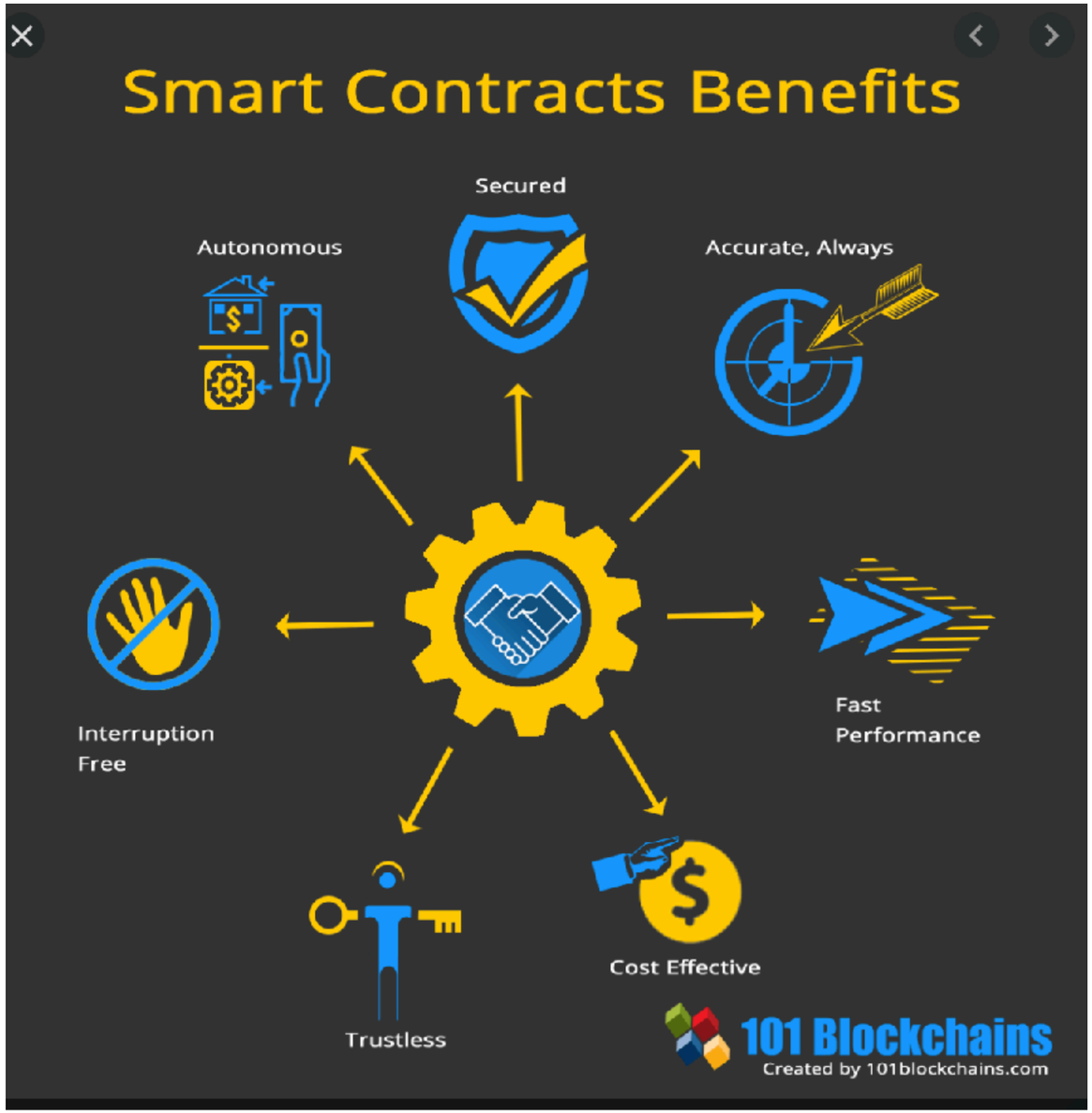
스마트 컨트랙트 설계서

클레이튼 & 솔리디티

2020.09.10

스마트 컨트랙트의 개요

- 스마트 계약 또는 스마트 컨트랙트는 계약 당사자가 사전에 협의한 내용을 미리 프로그래밍 하여 전자계약서 문서 안에 넣어두고, 이 계약 조건이 모두 충족 되면 자동으로 계약 내용이 실행되도록 하는 시스템이다.



인터페이스

클레이튼 KIP-7

- 클레이튼 Caver-js 라이브러리 에서 제공하는 KIP-7의 기본 인터페이스.
- 개발자 사이의 코드 규약을 정한다.
- 여러 구현체에서 공통적인 부분을 추상화 한다.

```
// IKIP7
event Transfer(address indexed from, address indexed to, uint256 value);
event Approval(address indexed owner, address indexed spender, uint256 value);

function totalSupply() external view returns (uint256);
function balanceOf(address account) external view returns (uint256);
function transfer(address recipient, uint256 amount) external returns (bool);
function allowance(address owner, address spender) external view returns (uint256);
function approve(address spender, uint256 amount) external returns (bool);
function transferFrom(address sender, address recipient, uint256 amount) external returns (bool);
function safeTransfer(address recipient, uint256 amount, bytes data) external;
function safeTransfer(address recipient, uint256 amount) external;
function safeTransferFrom(address sender, address recipient, uint256 amount, bytes data) external;
function safeTransferFrom(address sender, address recipient, uint256 amount) external;

// IKIP7Metadata (optional)
function name() external view returns (string memory);
function symbol() external view returns (string memory);
function decimals() external view returns (uint8);

// IKIP7Mintable (optional)
function mint(address _to, uint256 _amount) external returns (bool);
function isMinter(address _account) external view returns (bool);
function addMinter(address _account) external;
function renounceMinter() external;

// IKIP7Burnable (optional)
function burn(uint256 _amount) external;
function burnFrom(address _account, uint256 _amount) external;

// IKIP7Pausable (optional)
event Paused(address _account);
event Unpaused(address _account);

function paused() external view returns (bool);
function pause() external;
function unpause() external;
function isPauser(address _account) external view returns (bool);
function addPauser(address _account) external;
```

인터페이스

스테이킹 컨트랙트

- Staking() 스테이킹시 같은 갯수로 반환
  - arguments : (value)
  - output : 이벤트 호출 (sender,value)
- Unsticking() 언스테이킹시 같은 갯수로 반환
  - arguments : (value)
  - output : 이벤트 호출 (sender,amount)
- TransferToStaker() onlyOwner 스테이커 에게 클레이를 전송하는 기능
  - arguments : to,amount
  - output : x

```
contract Klaymore is KIP7,KIP7Metadata,KIP7Pausable,MultiSigWallet {
    address public owner;
    uint256 private _totalSupply;

    mapping (address => uint256) private _balances;
    event CoinDeposit(address indexed _from, uint256 _value);
    event SwapRequest(address indexed _from, uint256 _value);

    modifier onlyOwner(){
    require(msg.sender == owner);
    _;
    }

    constructor(string memory name, string memory symbol, uint8 decimals) KIP7Metadata(name, s
    owner = msg.sender;
    }

    //스테이킹 기능, 스테이킹시 같은 갯수의 토큰을 반환
    function Staking() public payable {
        _balances[msg.sender] += _balances[msg.sender].add(msg.value);
        _totalSupply = _totalSupply.add(msg.value);
        _mint(msg.sender,msg.value);
        emit CoinDeposit(msg.sender, msg.value);
    }
    //언스테이킹 기능,
    function Unstaking(uint256 amount) public returns (bool) {
        require(amount <= _balances[msg.sender]);
        _balances[msg.sender] = _balances[msg.sender].sub(amount);
        _burn(msg.sender,amount);
        msg.sender.transfer(amount);
        emit SwapRequest(msg.sender,amount);
        return true;
    }
    //스테이커에게 전송하는 기능. 이 기능은 오로지 컨트렉트의 주인만이 할수있다.
    function TransferToStaker(address payable _to, uint256 amount) onlyOwner public {
        _to.transfer(amount);
    }
}
```

인터페이스

멀티 시그니처

- submitTransaction() 트랜잭션 제안하기
  - arguments : to, value, data
  - output : transaction ID
- confirmTransaction() 트랜잭션 승인하기
  - arguments : transaction ID
  - output : 이벤트 호출 (sender, transactionID)

```
/// @dev Allows an owner to submit and confirm a transaction.
/// @param destination Transaction target address.
/// @param value Transaction ether value.
/// @param data Transaction data payload.
/// @return Returns transaction ID.
function submitTransaction(address destination, uint value, byte
    public
    returns (uint transactionId)
{
    require(isOwner[msg.sender]);
    transactionId = addTransaction(destination, value, data);
    confirmTransaction(transactionId);
}

/// @dev Allows an owner to confirm a transaction.
/// @param transactionId Transaction ID.
function confirmTransaction(uint transactionId)
    public
    ownerExists(msg.sender)
    transactionExists(transactionId)
    notConfirmed(transactionId, msg.sender)
{
    confirmations[transactionId][msg.sender] = true;
    emit Confirmation(msg.sender, transactionId);
    executeTransaction(transactionId);
}
```

설치및 배포

Truffle & Klaytn

- Truffle
  - pragma solidity ^0.5.0
  - truffle version : v5.1.x
  - solo version : 0.5.6
  - node version : 10.22.0
- Klaytn
  - Caver-js version :1.5.1
  - network : baobab(Testnet)

```
gim-yeong-il@gim-yeong-il-ui-MacBookAir ~/Desktop/front_poc  master ● ?  truffle complie

Compiling your contracts...
===========================
> Compiling ./contracts/HeartLink.sol
> Compiling ./contracts/Migrations.sol
> Compiling caver-js/packages/caver-kct/src/contract/access/Roles.sol
> Compiling caver-js/packages/caver-kct/src/contract/access/roles/PauserRole.sol
> Compiling caver-js/packages/caver-kct/src/contract/introspection/IKIP13.sol
> Compiling caver-js/packages/caver-kct/src/contract/introspection/KIP13.sol
> Compiling caver-js/packages/caver-kct/src/contract/lifecycle/Pausable.sol
> Compiling caver-js/packages/caver-kct/src/contract/math/SafeMath.sol
> Compiling caver-js/packages/caver-kct/src/contract/token/KIP7/IKIP7.sol
> Compiling caver-js/packages/caver-kct/src/contract/token/KIP7/IKIP7Receiver.sol
> Compiling caver-js/packages/caver-kct/src/contract/token/KIP7/KIP7.sol
> Compiling caver-js/packages/caver-kct/src/contract/token/KIP7/KIP7Metadata.sol
> Compiling caver-js/packages/caver-kct/src/contract/token/KIP7/KIP7Pausable.sol
> Compiling caver-js/packages/caver-kct/src/contract/utils/Address.sol
> Artifacts written to /Users/gim-yeong-il/Desktop/front_poc/build/contracts
> Compiled successfully using:
   - solc: 0.5.6+commit.b259423e.Emscripten.clang
```

```
const owners = ["0x84662F4A520F45A75B9AfD665C716eaE86D96E8b","0x276159d8986dBEEE8bFd5C79cb582AA24EB43662"]; // 다수의 오너들
const required = 2; //트랜잭션 날릴때 승인을 해야하는 숫자 (트랜잭션 제안자가 제안을 함과 동시에 승인 +1)
const name = "HeartLink";
const symbol = "HLT";
const decimals = 18;
const amount = 100000000000000000000;

// truffle로 배포할 때 얻을 수 있는 데이터들을 deployedABI와 deployedAddress 파일들에 저장한다.
module.exports = function (deployer) {
  deployer.deploy(HeartLink,owners,required, name, symbol, decimals).then(() => {
    if (HeartLink._json) {
      fs.writeFile(
        "deployedABI",
        JSON.stringify(HeartLink._json.abi),
        // fs에서 writeFile함수는 두개의 인자를 받는데 여기서는 첫번째 인자 파일을 읽고 거기에 HeartLink컨트랙트의 abi를 json형식으로 받고 문자열로 넣는다.
        (err) => {
          if (err) throw err;
          console.log("파일에 ABI 입력 성공");
        }
      );
      fs.writeFile("deployedAddress", HeartLink.address, (err) => {
        if (err) throw err;
        console.log("파일에 주소 입력 성공");
      });
    }
  });
};
```

설치및 배포

Truffle & Klaytn

- Truffle
  - pragma solidity ^0.5.0
  - truffle version : v5.1.x
  - solo version : 0.5.6
  - node version : 10.22.0
- Klaytn
  - Caver-js version :1.5.1
  - network : baobab(Testnet)

```
gim-yeong-il@gim-yeong-il-ui-MacBookAir ~/Desktop/front_poc master ● ? truffle deploy --network baobab

Compiling your contracts...
===========================
> Everything is up to date, there is nothing to compile.



Starting migrations...
======================
> Network name:    'baobab'
> Network id:      1001
> Block gas limit: 0 (0x0)


1_initial_migration.js
======================

   Deploying 'Migrations'
   ----------------------
   > transaction hash:    0xc33d5fd742ef974d76f98808a3f3fb7e79e48c94926fd20826d285959b6b0bd6
   > Blocks: 0            Seconds: 0
   > contract address:    0x474f7FC910DDD301E787633c9922d30A35aBB5F0
   > block number:        38168365
   > block timestamp:     1599723794
   > account:             0x64662F4A520F45A75B9AfD665C716eaE66D96E8b
   > balance:             5.392635748999999998
   > gas used:            219207 (0x35847)
   > gas price:           25 gwei
   > value sent:          0 ETH
   > total cost:          0.005480175 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:         0.005480175 ETH


2_deploy_contract.js
====================

   Deploying 'HeartLink'
   ---------------------
   > transaction hash:    0x118172735fe44c9ff6fbf2d10151d10bf0862ea71b22b8919390976c22d6df2d
```

컨트랙트 상속 구조도

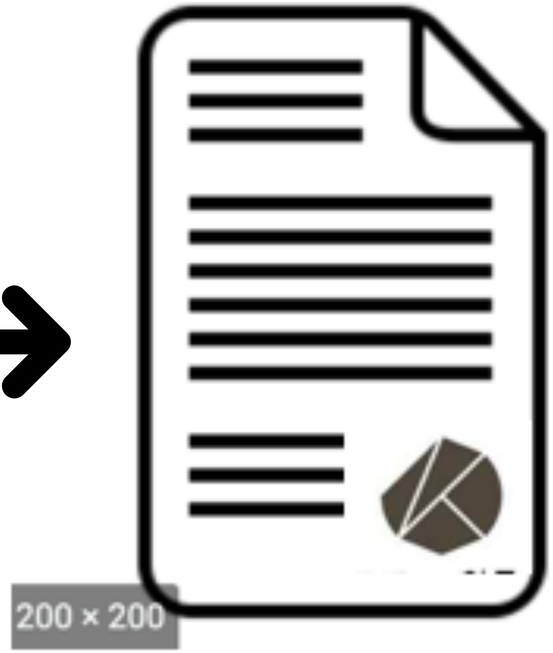
Klaytn interface & object

Address.sol
IKIP-13.sol
Pausable.sol
IKIP-7Receiver.sol
IKIP-7.sol
Safemath.sol
Roles.sol
PausableRole.sol

Klaytn Token Library

KIP-7.sol
KIP-7Pausable.sol
KIP-7Metadata.sol

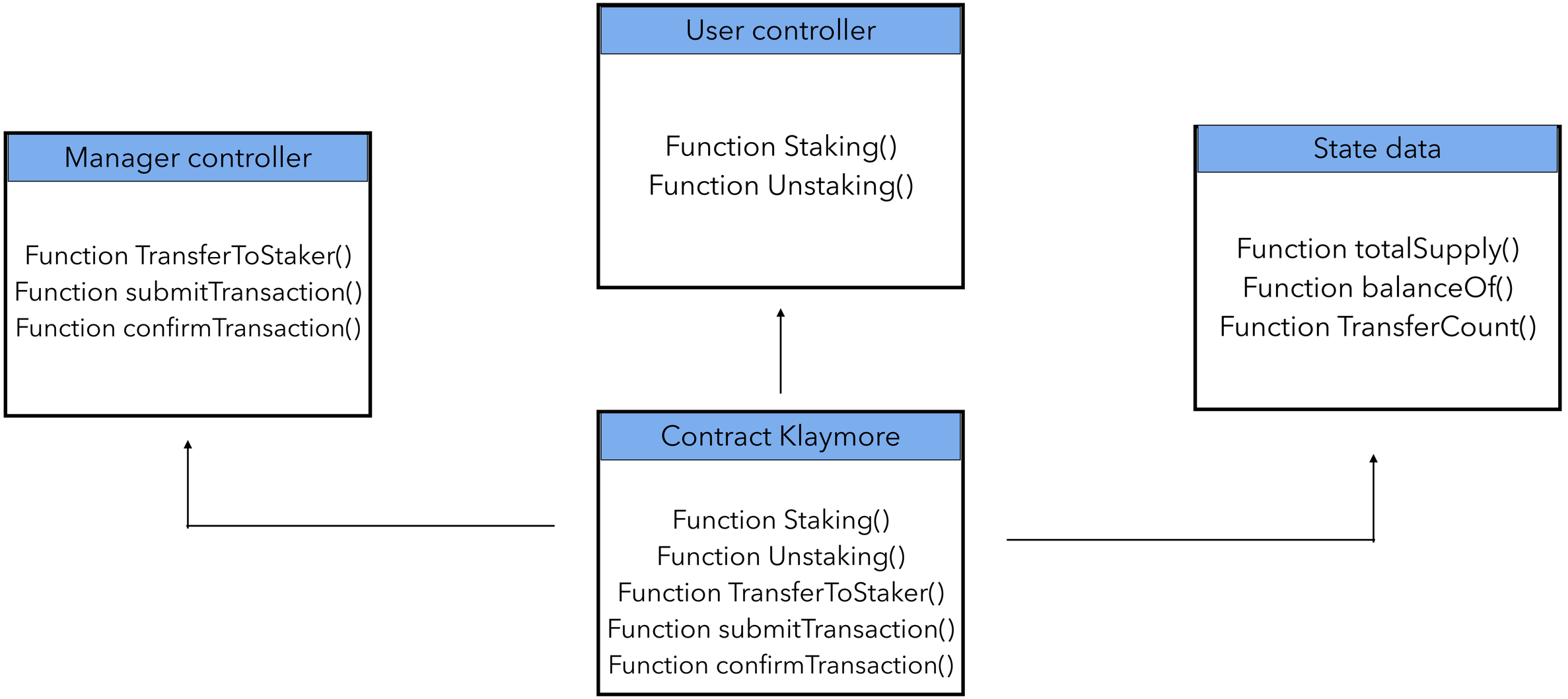
Klaymore.sol



Ethereum Consensys

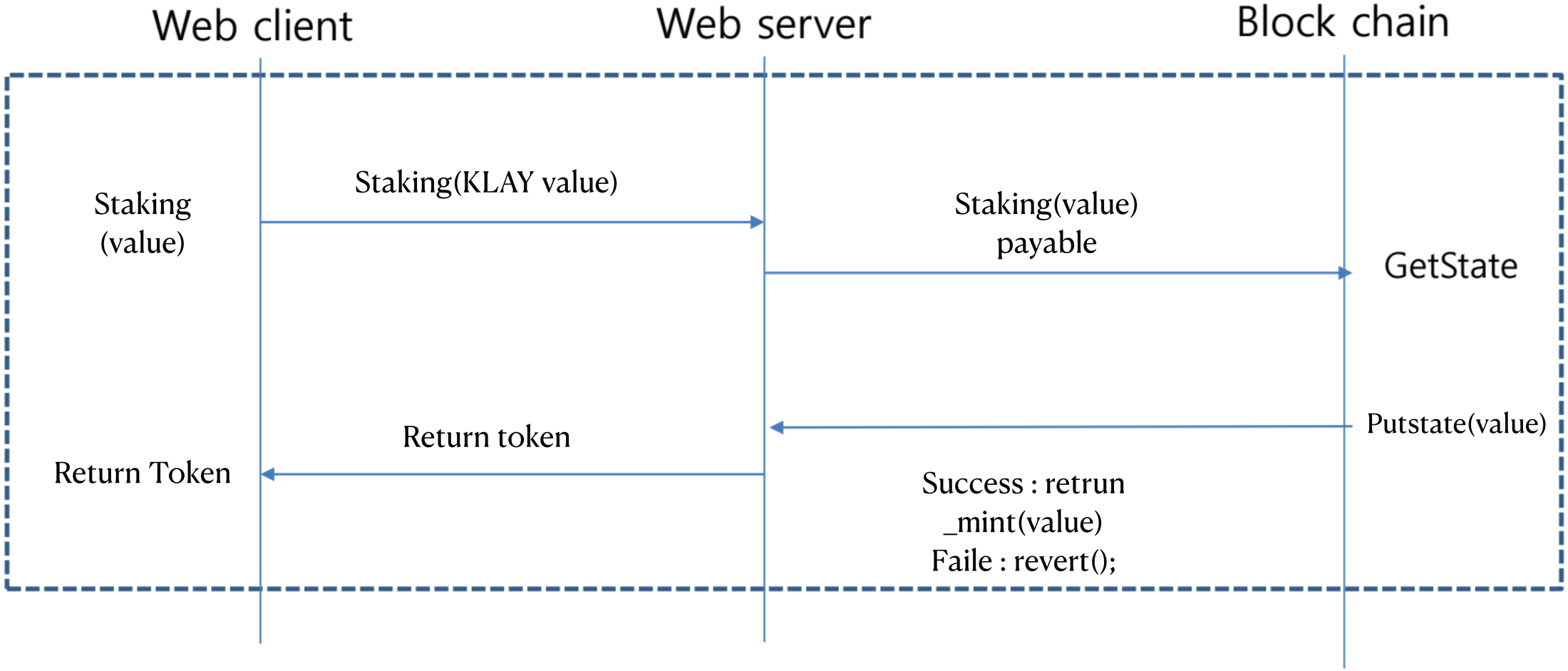
MultisigWallet.sol

클래스다이어그램


User controller
Function Staking()
Function Unstaking()
Manager controller
Function TransferToStaker()
Function submitTransaction()
Function confirmTransaction()
State data
Function totalSupply()
Function balanceOf()
Function TransferCount()
Contract Klaymore
Function Staking()
Function Unstaking()
Function TransferToStaker()
Function submitTransaction()
Function confirmTransaction()

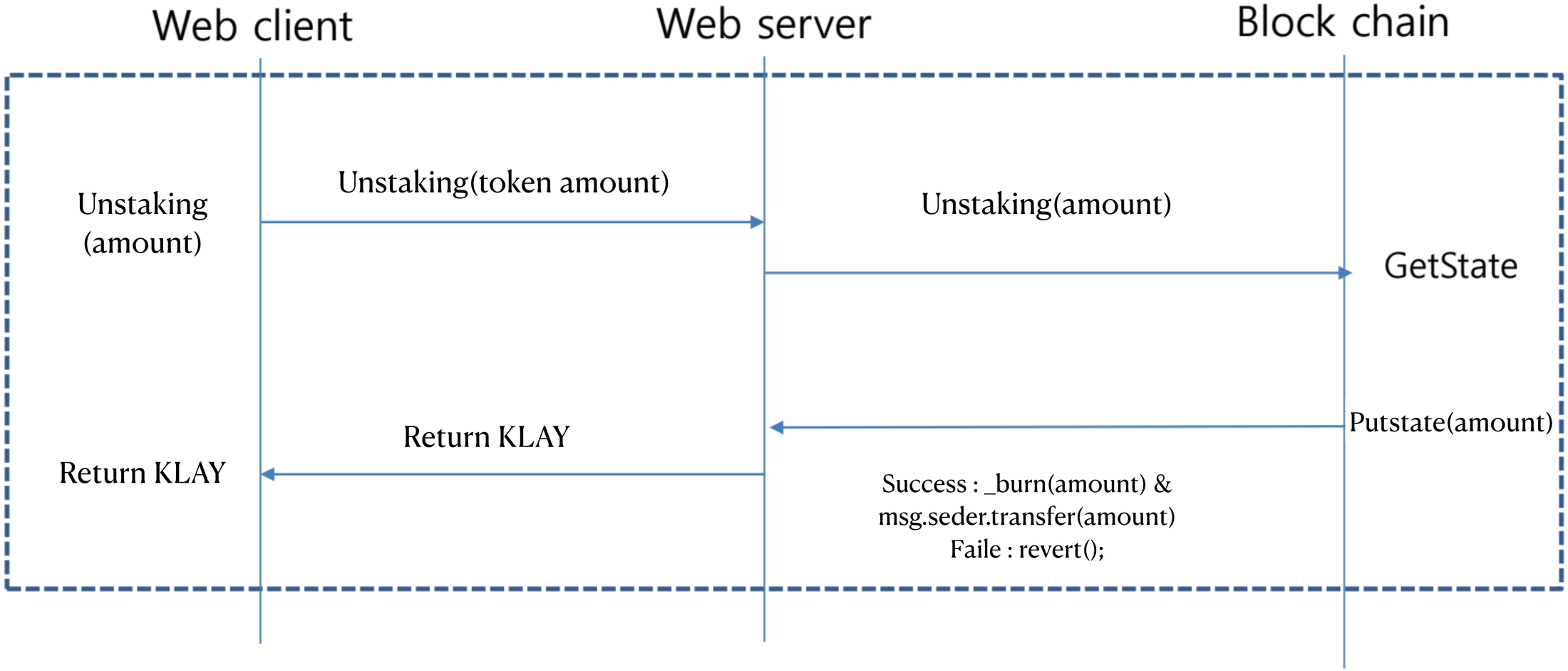
스마트 컨트랙트 연동 플로우 차트

Staking



스마트 컨트랙트 연동 플로우 차트

Unstaking



스마트 컨트랙트 연동 플로우 차트

TransferToStaker