# Blockchain-based Asset Management Application

An - Rossukhon

Growth Session #16 - July 12-13 2018

- Build an Asset Management application using Ethereum Smart contract (DAPP)

- Everyone can add asset

- Everyone can see the list of asset

- Everyone can see who is the owner, who is holding that asset

Blockchain

- A [blockchain](#), is a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.

- Decentralized platform.

Blockchain Explained!!!
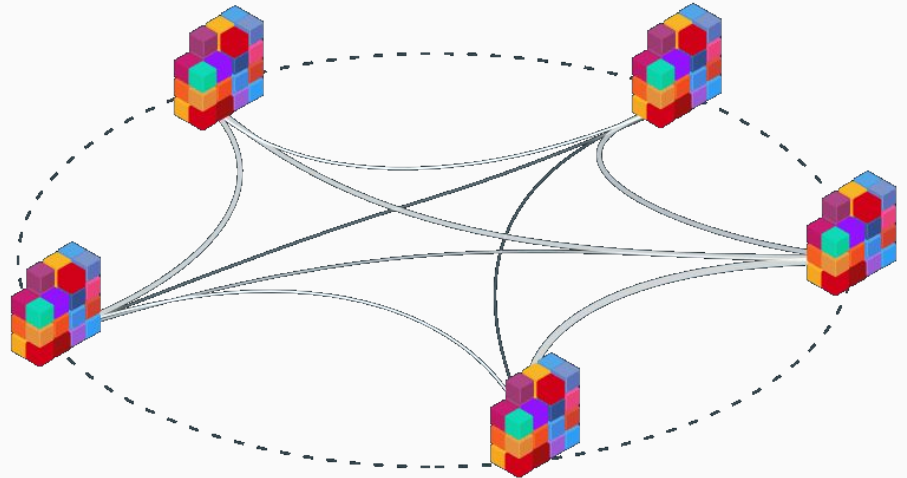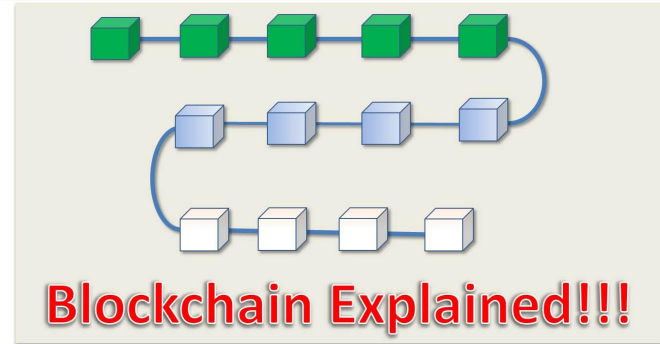
Ethereum

- Is an open source.

- Based on blockchain technology where all transactions are meant to be irreversible and unchangeable.

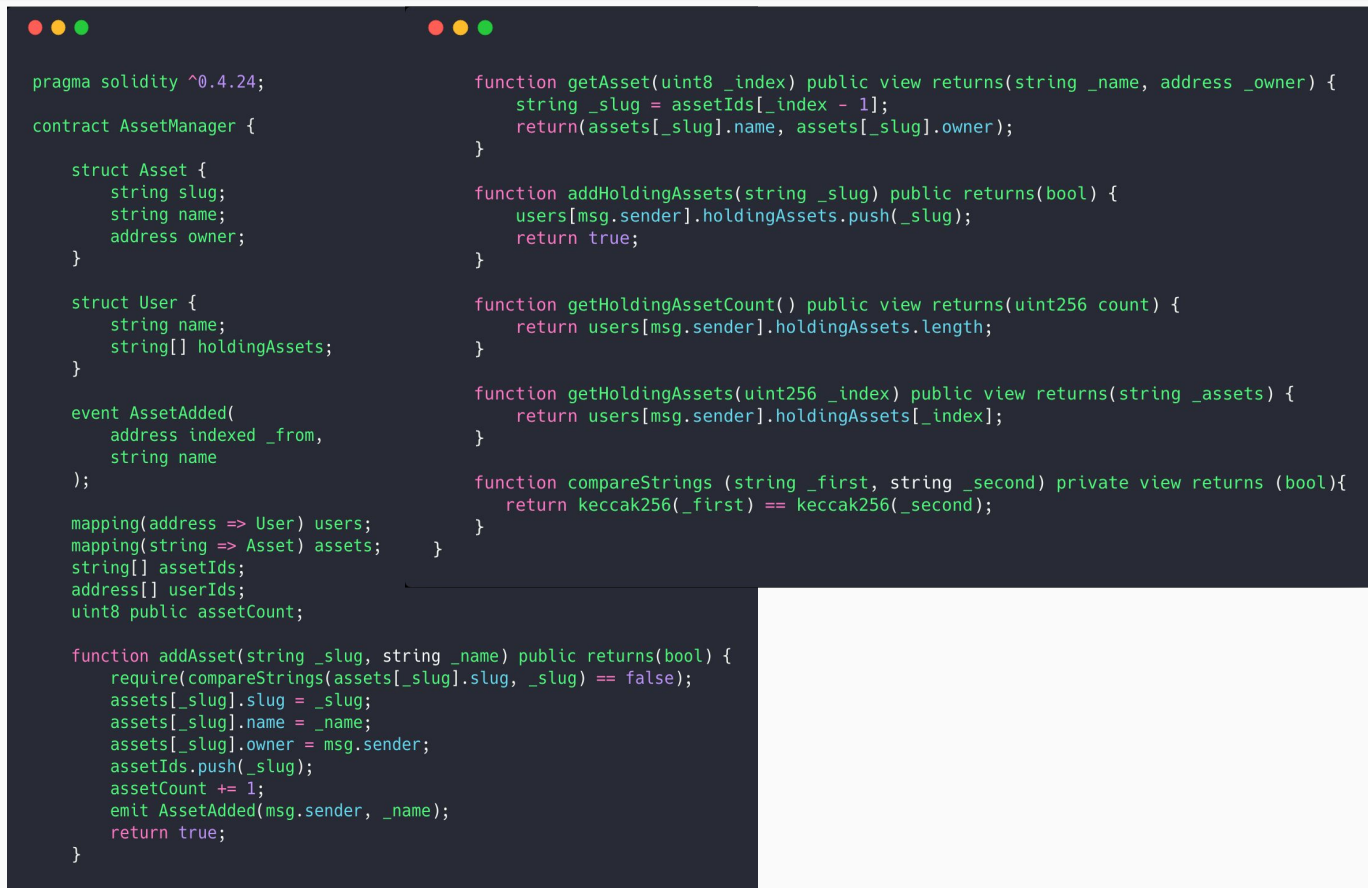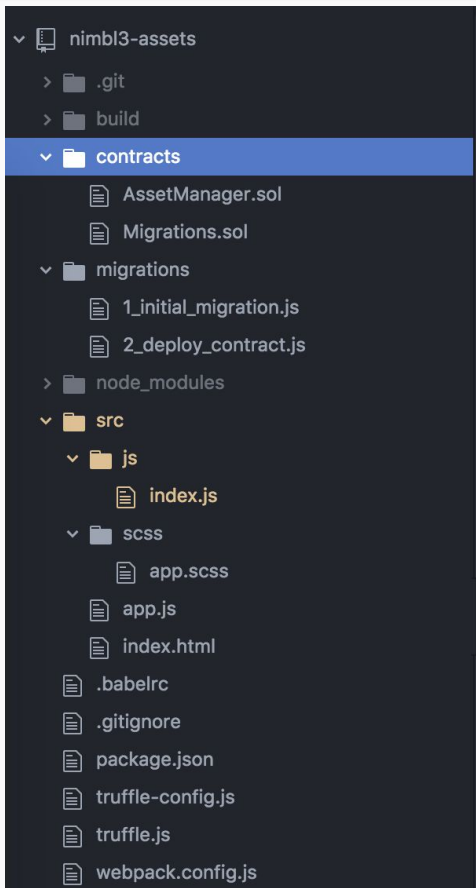- Is a decentralized platform that allows to run Smart Contracts.

Smart contract

- An application that runs on Ethereum platform

- We can add business logic

- Using Solidity language

Technologies

- Solidity - Smart contract

- Truffle framework - Manage contract, migration, deploy and test contracts.

- web3.js, Javascript, Webpack - Client app

- Metamask - Connect between Browser and Blockchain network

Achievements

```solidity
pragma solidity ^0.4.24;

contract AssetManager {

    struct Asset {
        string slug;
        string name;
        address owner;
    }

    struct User {
        string name;
        string[] holdingAssets;
    }

    event AssetAdded(
        address indexed _from,
        string name
    );

    mapping(address => User) users;
    mapping(string => Asset) assets;
    string[] assetIds;
    address[] userIds;
    uint8 public assetCount;

    function addAsset(string _slug, string _name) public returns(bool) {
        require(compareStrings(assets[_slug].slug, _slug) == false);
        assets[_slug].slug = _slug;
        assets[_slug].name = _name;
        assets[_slug].owner = msg.sender;
        assetIds.push(_slug);
        assetCount += 1;
        emit AssetAdded(msg.sender, _name);
        return true;
    }

    function getAsset(uint8 _index) public view returns(string _name, address _owner) {
        string _slug = assetIds[_index - 1];
        return(assets[_slug].name, assets[_slug].owner);
    }

    function addHoldingAssets(string _slug) public returns(bool) {
        users[msg.sender].holdingAssets.push(_slug);
        return true;
    }

    function getHoldingAssetCount() public view returns(uint256 count) {
        return users[msg.sender].holdingAssets.length;
    }

    function getHoldingAssets(uint256 _index) public view returns(string _assets) {
        return users[msg.sender].holdingAssets[_index];
    }

    function compareStrings (string _first, string _second) private view returns (bool){
        return keccak256(_first) == keccak256(_second);
    }
}
```

Achievements

Screen
Owner: 0xbdd8b3437569d23a88f419e7266fa5a905c59be6

Keyboard
Owner: 0xbdd8b3437569d23a88f419e7266fa5a905c59be6

Mouse
Owner: 0xbdd8b3437569d23a88f419e7266fa5a905c59be6

MacBook
Owner: 0xbdd8b3437569d23a88f419e7266fa5a905c59be6

New Asset

Slug

slug

Name

name

Add

Feels like building a standard Web application:

- Data lives on Ethereum

- Backend is a Smart contract

- UI/Frontend is the Client application

Not FREE 🗌
Every transaction
requires Gas

Smart Contracts:

- Easy: normal programming language and easy to add logic by code

- Hard: really hard to change or update data/business logic after the contract has been deployed onto the Ethereum network

Building a Smart contract is like building Hardware products

Next Steps

- Complete other features

- Deploy to Heroku and test Ethereum network e.g. Rinkeby

Thanks!

Contact Nimbl3

hello@nimbl3.com

399 Sukhumvit Road, Interchange 21
Klongtoey nua, Wattana
Bangkok 10110

28C Stanley St,
Singapore 068737

20th Floor, Central Tower
28 Queen's Road
Central, Hong Kong

nimbl3.com