Exploring Smart Contract

Ethereum Smart Contract & Remix
Lecture goals

- Decentralised applications
- Smart contracts
- Solidity
  - Variables, types, visibility
  - Events
  - Modifiers
- Web3.js
- Remix IDE
- Exercise
Decentralised applications

- DApps
- Run on a blockchain
- P2P network
- Distributed ledger
- Smart contracts

Source: coindesk
Decentralised applications - Use cases

- Healthcare
- Voting
- Supply chain
- Banking
- Record keeping
- Auctions
- IoT
- Security layer

Image source: Asia blockchain review
Decentralised applications - Benefits

- Zero downtime
- Privacy
- Resistance to censorship
- Data integrity
- Trust-less
- Verifiable behavior

Source: https://ethereum.org/en/developers/docs/dapps/
Source: allerin blog
Decentralised applications - Drawbacks

- Maintenance
- Performance
- Network congestion
- User experience
- Centralisation
- Interoperability
- Cost
- Rules and regulations

Source: https://ethereum.org/en/developers/docs/dapps/
Source: FreightWaves
Smart contract

- Code
- Computer protocol
- Autonomous
- Hold funds/state
- Immutable
Cycle of smart contract execution over Ethereum blockchain

Source: Smart Contract: Attacks and Protections by Sayeed
Solidity

- Proposed by Gavin Wood
- Programming language
- Smart contracts
- Ethereum
- Ethereum virtual machine
pragma solidity >=0.7.0 <0.9.0;

/**
 * @title Storage
 * @dev Store & retrieve value in a variable
 */

contract Storage {
    uint256 number;

    /**
     * @dev Store value in variable
     * @param num value to store
     */
    function store(uint256 num) public {
        number = num;
    }

    /**
     * @dev Return value
     * @return value of 'number'
     */
    function retrieve() public view returns (uint256) {
        return number;
    }
}
Variables, types, visibility

- Integer
- String
- Bool
- Constant

Visibility: Public, Private, Internal, External
Events

- EVM logging facilities
- Call javascript callbacks
- Stores the arguments passed in transaction logs
- Inform external users
Modifiers

- Modify/control behavior of function
  - Add prerequisite
    - Owner of SC can execute
- Use special symbol "_;"
- Automatically checking a condition
  - Prior to executing a function

```solidity
string studentName;
string course;
uint creditPoints;
address admin;

constructor () {
    admin = msg.sender;
}

modifier onlyAdmin {
    require(msg.sender == admin);
}

event StudentDetails { 
    string studentName, 
    string course, 
    uint creditPoints
}

function setStudentDetails(string _studentName, string _course, uint _creditPoints) onlyAdmin public { 
    studentName = _studentName;
    course = _course;
    creditPoints = _creditPoints;
    emit StudentDetails(_studentName, _course, _creditPoints);
}
Web3.js

- Collection of libraries
- Interact with decentralised apps

`npm install web3`
Remix - Ethereum IDE

https://remix.ethereum.org/
DApp development environment

- **Terminal**
- **Node Js** (node -v)
  - **NPM** (npm -v)
  - [https://nodejs.org/en/](https://nodejs.org/en/)
- **Ganache cli**
  - `npm install -g ganache-cli`
    - **Start server**
      - `ganache-cli`
- **Web3.js**
  - `npm init`
  - `npm install web3`
  - [https://web3js.readthedocs.io/en/v1.5.2/getting-started.html](https://web3js.readthedocs.io/en/v1.5.2/getting-started.html)
Lecture summary

- Decentralised applications
- Smart contracts
- Solidity
  - Variables, types, visibility
  - Events
  - Modifiers
- Web3.js
- Remix IDE
- Exercise