Chapter 4

Blockchains that support Smart Contracts, and DApp

Smart Contract

- The logic which is implemented on blockchain
- It is a piece of code which will be stored and executed on each node
- Its logic and behavior is transparent and permanent.
- You cannot change it the way you wanted after it is deployed.
- Smart contract acts like legal contract governed by the mechanism of blockchain
- Some use cases
    - Smart contract can hold cryptos, and it will be transferred to the right accounts according to a condition. E.g. Escrow services.
    - Smart contract can tell the public how the funds will be distributed. In ICO scenario.

What blockchains supporting Smart Contract

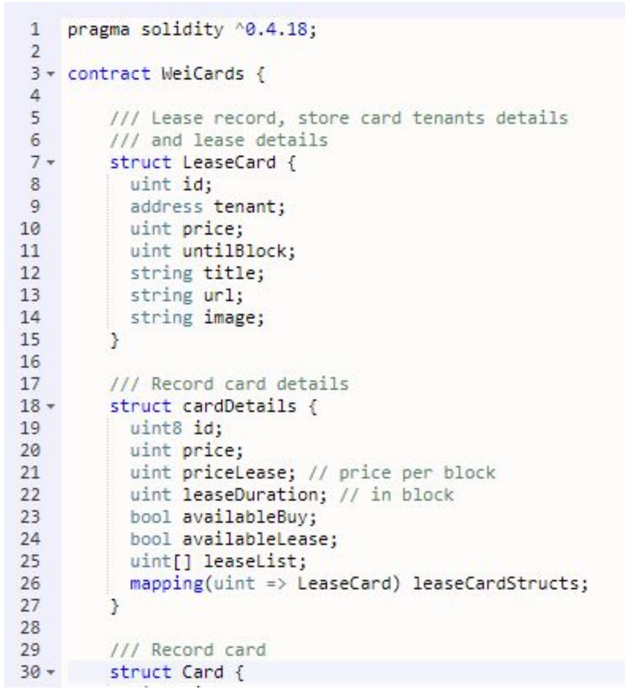
- Ethereum
  - Language Solidity
- EOS
  - C/C++
- NEO
  - C#, VB.Net, F#, Java, Kotlin, Python
- Cardano
  - PLUTUS (Haskell)
- Pl^G
  - To be found out :-)

Ethereum so far the most popular blockchain with smart contract support

- Ether or ETH are the crypto currency created in Ethereum.
- If Bitcoin is like “Gold”, Ether is like “USD”
- Some of the applications using Smart Contract
  - Initial Coin Offering aka ICO
    - Used to raise capital
    - Smart Contract shows the number of tokens and allocation of tokens.
  - CryptoKitties
    - Auction logic is transparent and run in Smart Contract.
    - Certain key logic is in Smart Contract to ensure authenticity
  - LocalEthereum
    - Escrow service, e.g. buyers make payment, Smart Contract holds payment, Sellers delivers Ethers, buyers asks Smart Contract to release the fund.

Ethereum - Solidity

Solidity language is used to write smart contract on Ethereum. It looks like
https://remix.ethereum.org/#optimize=false&version=soljson-v0.4.24+commit.e67f0147.js



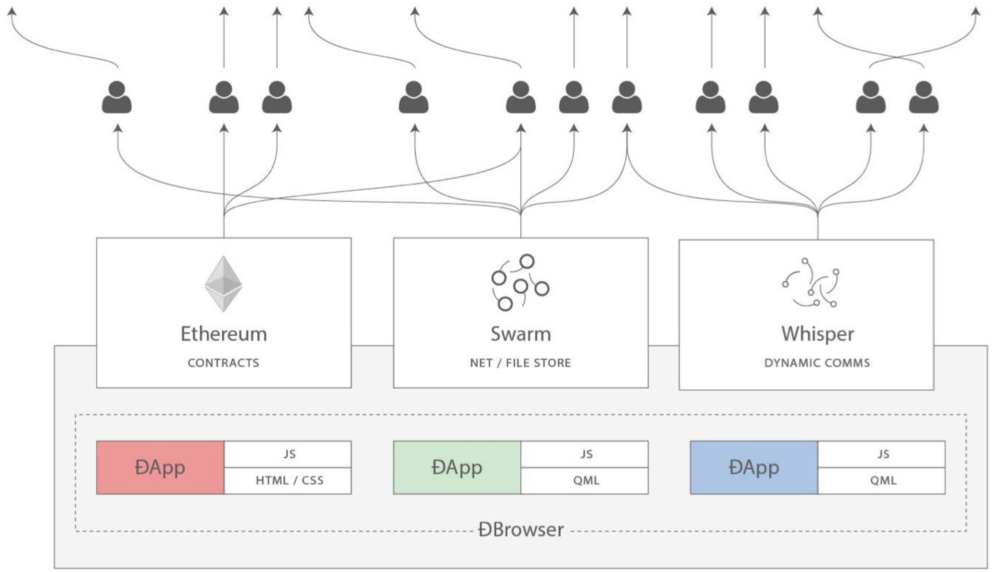
```
pragma solidity ^0.4.18;

contract WeiCards {

    /// Lease record, store card tenants details
    /// and lease details
    struct LeaseCard {
      uint id;
      address tenant;
      uint price;
      uint untilBlock;
      string title;
      string url;
      string image;
    }

    /// Record card details
    struct cardDetails {
      uint8 id;
      uint price;
      uint priceLease; // price per block
      uint leaseDuration; // in block
      bool availableBuy;
      bool availableLease;
      uint[] leaseList;
      mapping(uint => LeaseCard) leaseCardStructs;
    }

    /// Record card
    struct Card {
```

Ethereum - Underneath of the Smart Contract

- **Contracts**: decentralized logic
  - Runs on Ethereum Virtual Machine (EVM)
- **Swarm**: decentralized storage
  - The "database"
- **Whisper**: decentralized messaging
  - The "connection"

"Both Swarm and Whisper are complementary technologies contributing to the vision of Ethereum as a "world computer". When imagining Ethereum as a metaphor for a shared computer, it should be noted that computation alone is not enough.

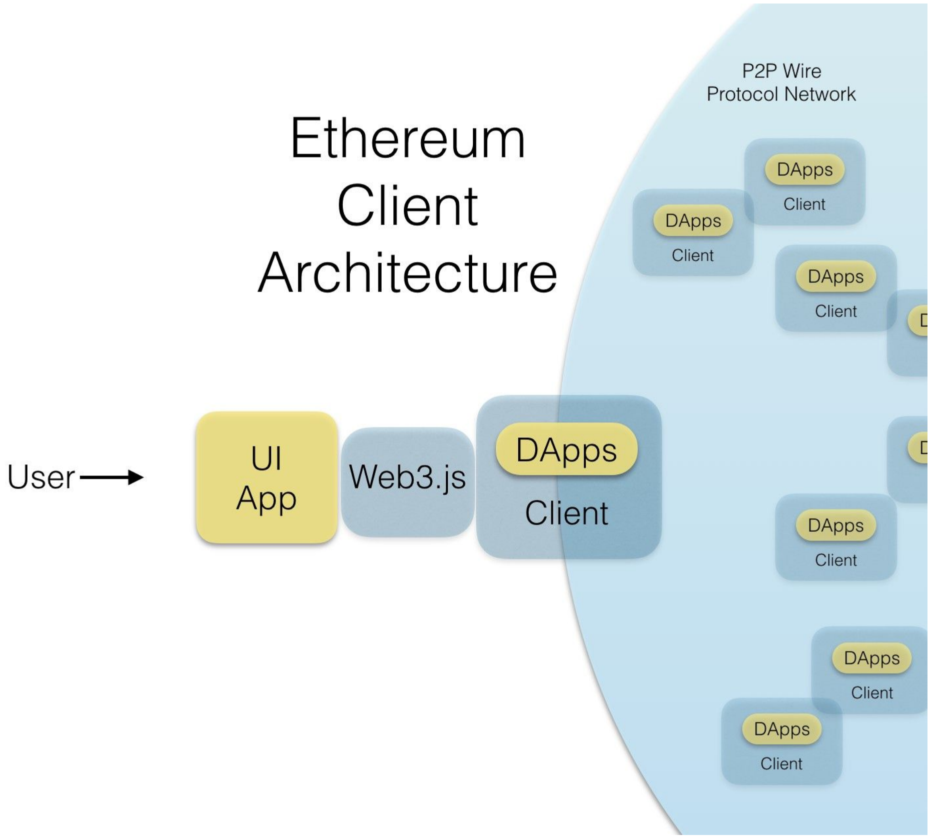
For a computer to be fully useful, it also needs storage to "remember" things and bandwidth to "communicate" them. "



DApp, Smart Contract

- DApp stands for Decentralized Application
  - It is web application utilising blockchain features.
  - It uses Smart Contract features for some of the core logic.
    E.g. Storying tokens
    - Save tokens into Smart Contract instead of recording it into someone database.
    - Smart contract is safer as it is on blockchain.
  - DApp front-end can use Web3.js library to connect to Smart Contact.
- Why DApp
  - It is same as why Blockchain and why decentralized (refer to chapter 1)
  - Certain core logic on blockchain make a lot more sense
    - E.g. Escrow or token distribution.

DApp Architecture



Next Chapter

- You will learn more about Solidity
- You will learn how to use Web3.js to communicate with Smart Contract on Ethereum
- How about jump ahead by trying https://cryptozombies.io/