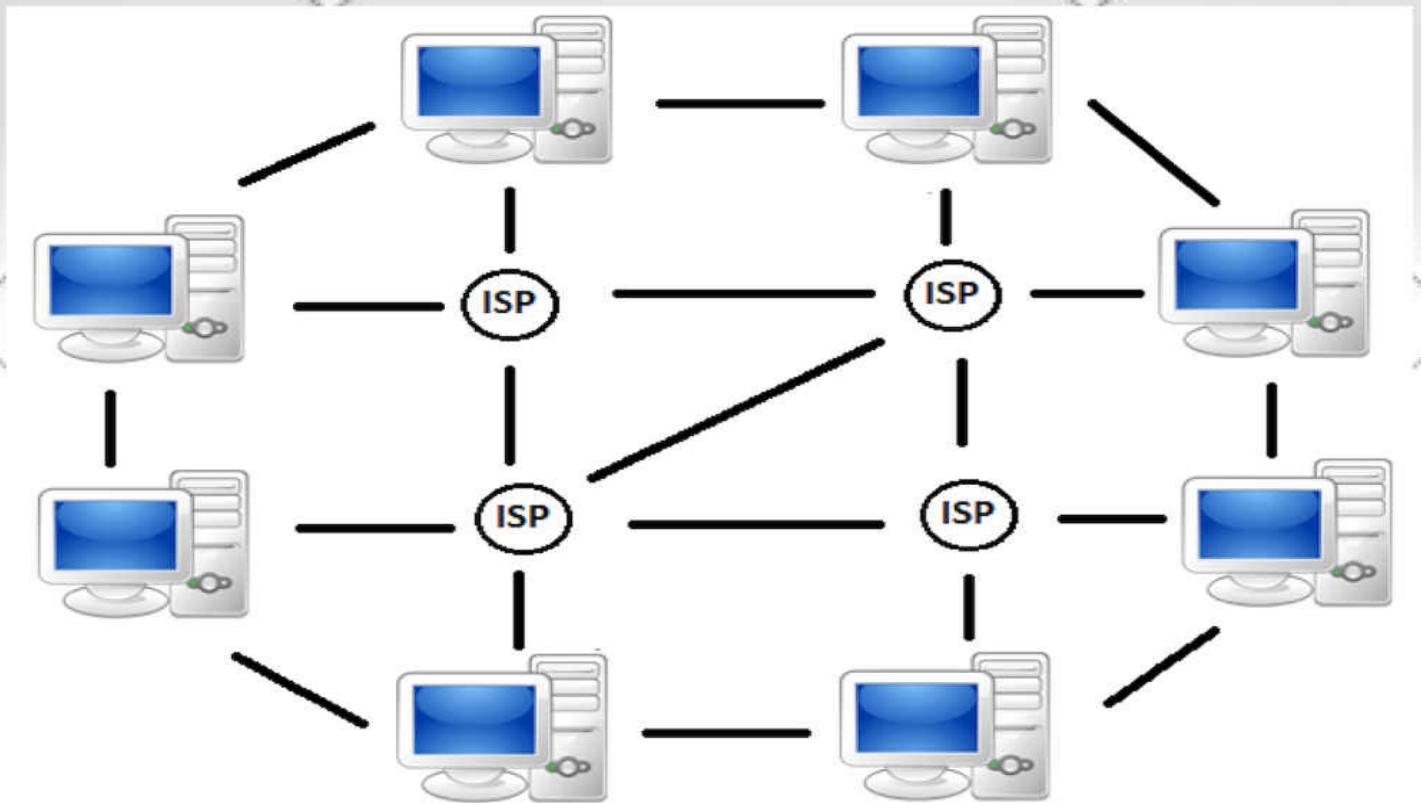


**BUILD YOUR FIRST DAPP**

# DApps

## Web 3.0



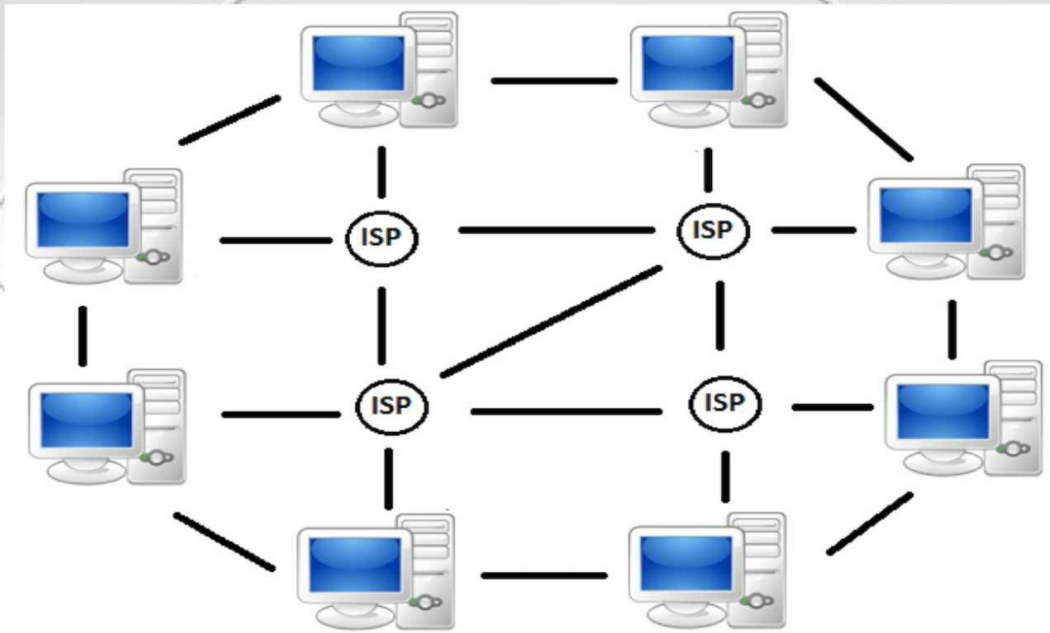
**THANIKASALAM K**

see more please visit: <https://homeofpdf.com>

BUILD YOUR FIRST DAPP

# DApps

Web 3.0



THANIKASALAM K

Copyright © 2018 by Thanikasalam K. All rights reserved.

No portion of this book may be reproduced mechanically, electronically, or by any other means, including photocopying, without written permission of Thanikasalam K. It is illegal to copy this book, post it to a website, or distribute it by any other means without permission from Thanikasalam K.

### **Limits of Liability and Disclaimer of Warranty**

The author and publisher shall not be liable for your misuse of this material. This book is strictly for informational and educational purposes.

### **Warning – Disclaimer**

The purpose of this book is to educate and entertain. The author and/or publisher do not guarantee that anyone following these techniques, suggestions, tips, ideas, or strategies will become successful. The author and/or publisher shall have neither liability nor responsibility to anyone with respect to any loss or damage caused, or alleged to be caused, directly or indirectly by the information contained in this book

First Edition: December 2018

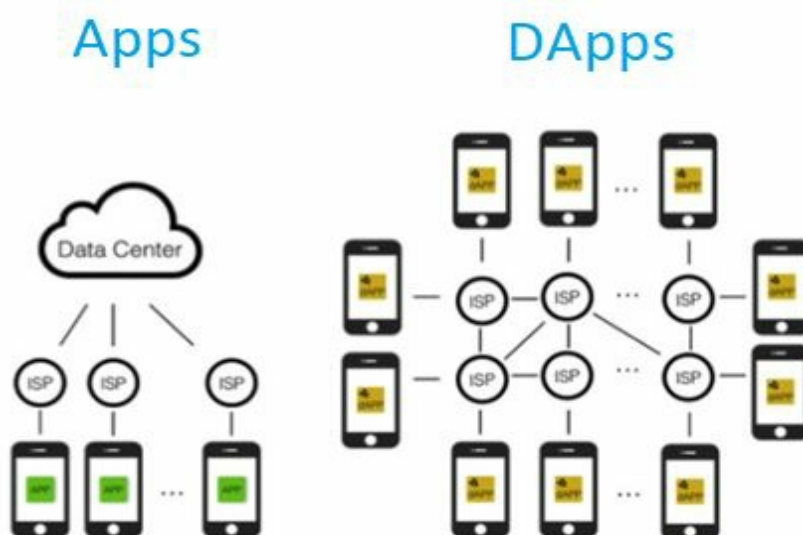
# Table of Contents

1. Introduction to DApps
2. Overview: Blockchain
3. Overview: Smart Contracts
4. Overview: Ethereum
5. Exercise: Ethereum Wallet – Ropsten Test Network
6. Relationship between DApps & Smart contracts
7. Overview: Solidity
8. Exercise: Deploy simple Smart Contract
9. Pre-requisition to build DApp
10. Exercise: Build your first DApp
11. Debug
12. Blockchain Security
13. Further Reading
14. Learning Activity
15. Summary

## 1. Introduction to DApps

This book helps you to build your first DApps (Decentralized Applications). Before we learn about DApps, it is important to know few technical terms like Blockchain, Ethereum, Smart contracts, Wallets, Tokens, Coin and Solidity programming language to better understanding. We will be covering these terms with this book in next chapters. At the end of reading this book you will be able to write your own DApps.

*What is DApps?*



Open-source software that leverage on the Blockchain technology called Decentralized Applications (DApps). It is not like our traditional web system, client-server based. DApps are running at client's machines (peer to peer) without a central managing server. For example, traditional web-based applications like Facebook, Whatsapp, Uber and Gmail are having centralized servers to manage our information. DApps are extremely opposite to this model, information spread across all clients' machine which means decentralized with high security. There is no central authority to manage all these services, it is all managed by a group of

people, and they are called miners. You need to Paradigm shift your thought in order to learn DApps.

### *Why DApps?*

IT invention needs to be solved a real practical valuable problem. Then only it will be successful and followed by all of us. For DApps, it will solve the two main issues we have with the current centralized technology. There are trust and transparency. Centralized network like Facebook, Whatsapp, AWS and Gmail, they are holding your personal data, how they can be assured that, they are keeping your data with more secure without let anyone access your personal information. How we can trust them as they are preventing from hacking? How they can be assured about fraudulent activities? In Decentralized peer to peer network, you have no central authority to control your data, you are owner for your own data.

Other reasons for DApps are

- Security.
- Eliminate the middleman.
- No central point of failure
- Protected against hacking
- Reliability, there is no down time with peer to peer network, since data is spread across the network.
- High Speed. When you are downloading a file from centralized network, one server is serving your file to download, in Decentralized network, you are downloading a file from millions of nodes. So data transfer is faster in Decentralized network.
- Decentralized network cannot be blocked by any person.
- Cryptography algorithm, which creates a secure SHA256 hash.



- Profit sharing. All social media sites are benefiting with people content and not paying for the content creator. In Decentralized network, all are paid for their respective jobs.

### *How DApps works?*

The easiest way to understand about DApps is first need to understand how traditional websites operate.

In traditional web application, to render a page from the server uses HTML, CSS and JavaScript. It will need an API to get the details from the database. When you are login to Twitter, the page will call an API to fetch your personal details and display it on the front-end user interface page.

You can think of a traditional website like this:

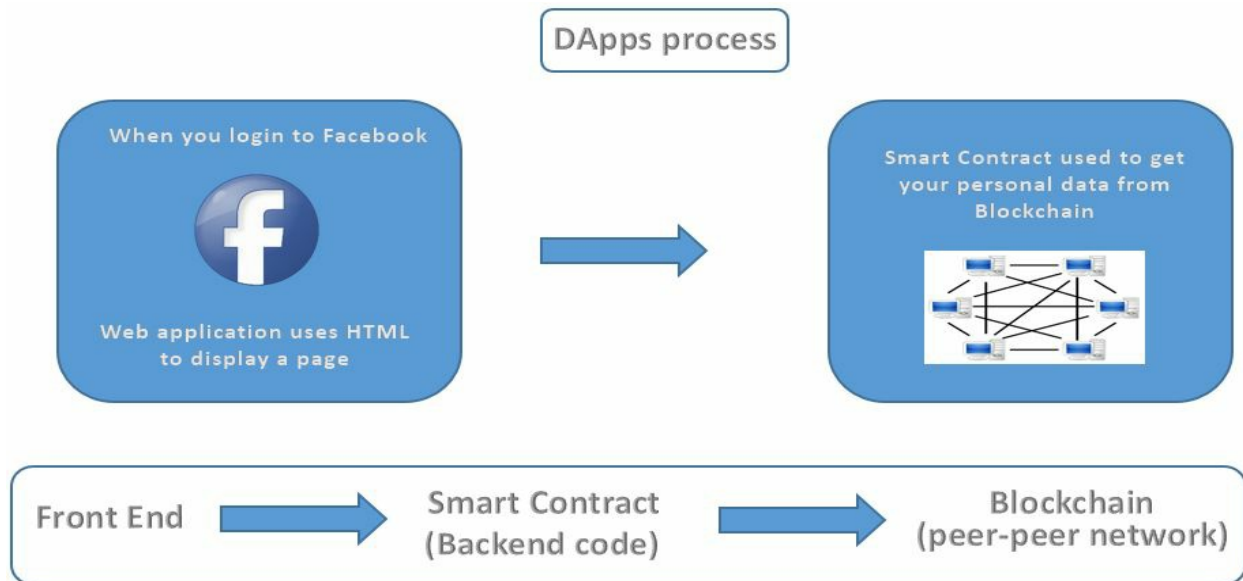


A DApps are very similar to a traditional web application. The front end uses the *exact same* technology to render the page. The one critical difference is that instead of an API connecting to a Database, you have a Smart Contract connecting to a blockchain. In DApps uses smart contracts to execute code and fetch information from the Blockchain. Blockchain replaces the database.

You can think of DApps like this:



Compare to traditional web application, DApps use Smart contract instead of API and Blockchain instead of Database.



The best example for DApps are Bitcon and Ethereum.

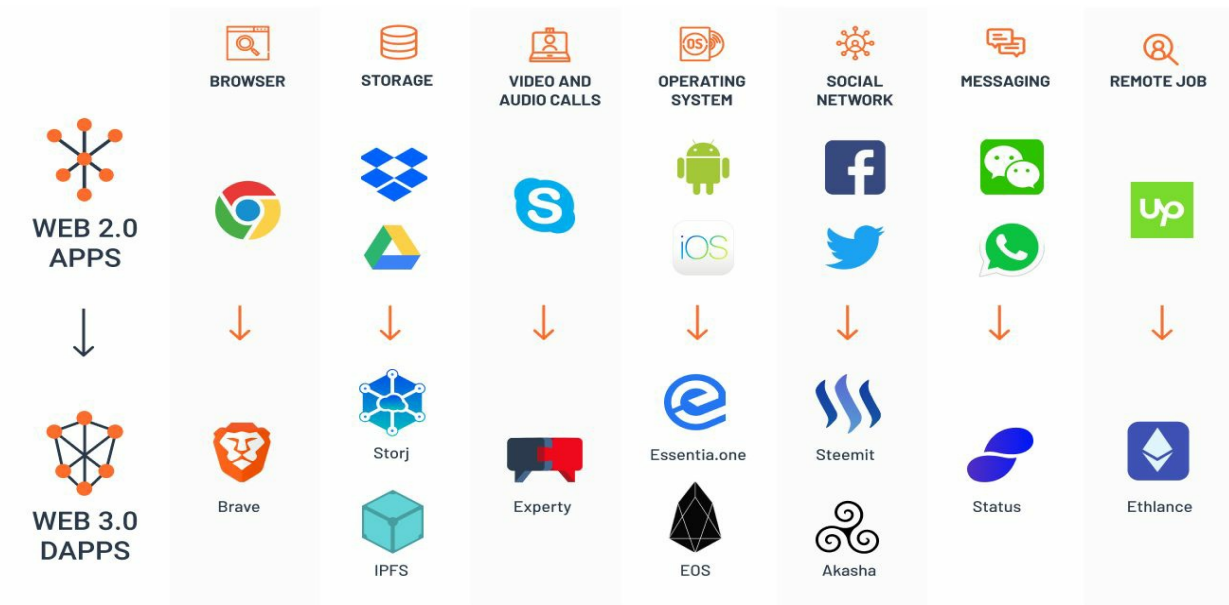
### *Web2.0 and Web3.0*

DApps are called Web3.0. When you are accessing a website in your browser, you really cannot see the difference whether it is a web2.0 or web3.0. Web3.0 websites are loading information from Blockchain network. Here is the list of Web 2.0 and Web3.0 apps for your easy reference.

Example:

<https://steemit.com/>. Steemit is a blogging and social networking website owned by Steemit Inc that uses the Steem blockchain to reward publishers and curators.

<https://www.cryptokitties.co/> . CryptoKitties is one of the world's first games to be built on blockchain technology



Source: Google

Functions	Web 2.0	Web3.0 (DApps)
<b>Scalable computation</b>	Amazon EC2	Ethereum, Truebit
<b>File storage</b>	Amazon S3	IPFS/Filecoin, Storj
<b>External data</b>	3 <sup>rd</sup> Party APIs	Oracle (Augur) ( <a href="https://www.augur.net/">https://www.augur.net/</a> )
<b>Monetization</b>	Ads, selling goods	Token model
<b>Payments</b>	Credit Card, Paypal	Bitcoin, Ethereum, state channels

In Blockchain, everyone can be viewing the available data in the network and you can ask a question, how secure DApps is? DApps are running in Blockchain technology. It has a high level of security using SHA256 cryptographic algorithm. Now what is Blockchain? Let's see in next chapter.

## 2. Overview: Blockchain

We need to be familiar with the buzz word "Blockchain" while we are studying about DApps. In 20<sup>th</sup> Century the most innovative invention was "Internet", invented in 1990 by Tim Berners Lee. Currently, many researchers enthusiastically say that the "Blockchain" technology, the

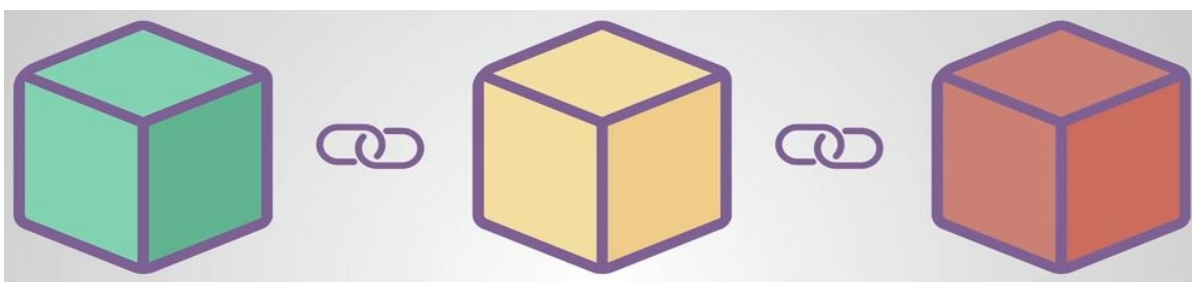
decentralized “digital ledger” system is an important invention in 21<sup>st</sup> century and it is the future.

Blockchain became most popular after invented the first crypto currency Bitcon in 2009 by Satoshi Nakamoto - is a pseudonym. Blockchain changes the software model from centralize to decentralize with its unique rich features: trust, immutable and secured.

### *What is Blockchain?*

*“Past cannot be changed, forgotten, edited, or erased. It can only be accepted” -  
Unknown*

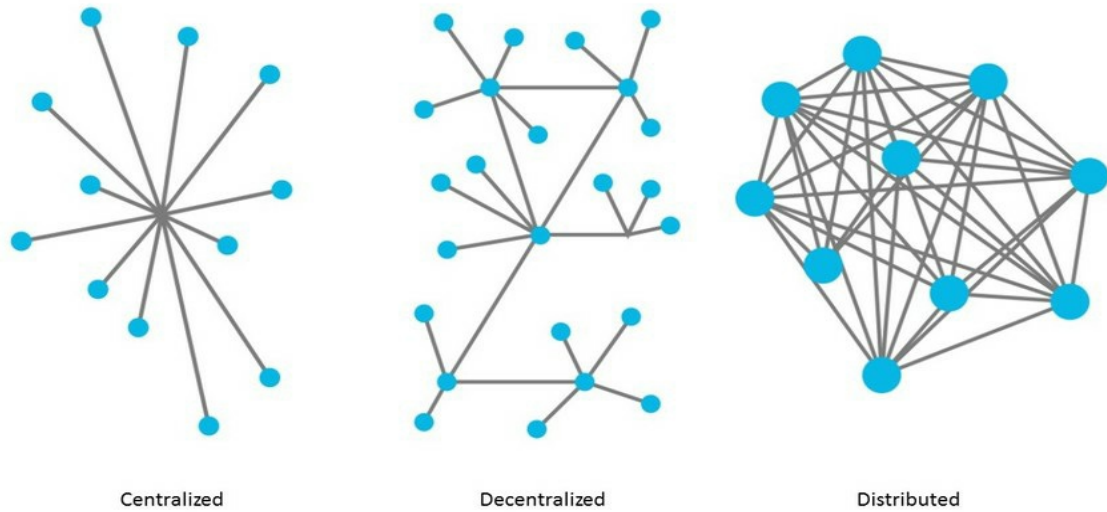
Hope you agree with the above unknown quote and it exactly suite for blockchain technology. Once record pushed in Blockchain, that past record cannot be changed, edited or deleted. Only you can add a new record in Blockchain.



Blockchain is a globally shared database and a distributed digital ledger that allows transactions to occur without a middleman such as government, bank, Credit Card Company, finance company etc. In Blockchain series of records (Blocks) are linked and stored in cryptography method. Chain of blocks are created and stored in decentralized network. These blocks are not stored in centralized server nor controlled by an individual, it stored in decentralized system. There is no central database in blockchain technology. No one or group of people has the power to change or tamper the data in block chain technology.

## Features

A decentralized system is a subset of a distributed system. In Decentralized system decision made by aggregate result from many nodes and there is no central point to made decision. In Distributed system the processing is shared with many nodes, but the decision may still be centralized.



- Decentralized
- Solve double spending problem
- Greater Transparency & Resilience
- Enhanced Security
- Improved traceability
- Increased efficiency and speed (share workloads among peers instead of from a central server)
- Reduced costs
- Once blocks send to Blockchain cannot be altered. (immutable)
- All blocks are stored in distributed system (peer to peer network).
- Reduces fraud

Internet                      Vs                      Blockchain

Centralized                      Distributed

OS: Windows, Linux, iOS

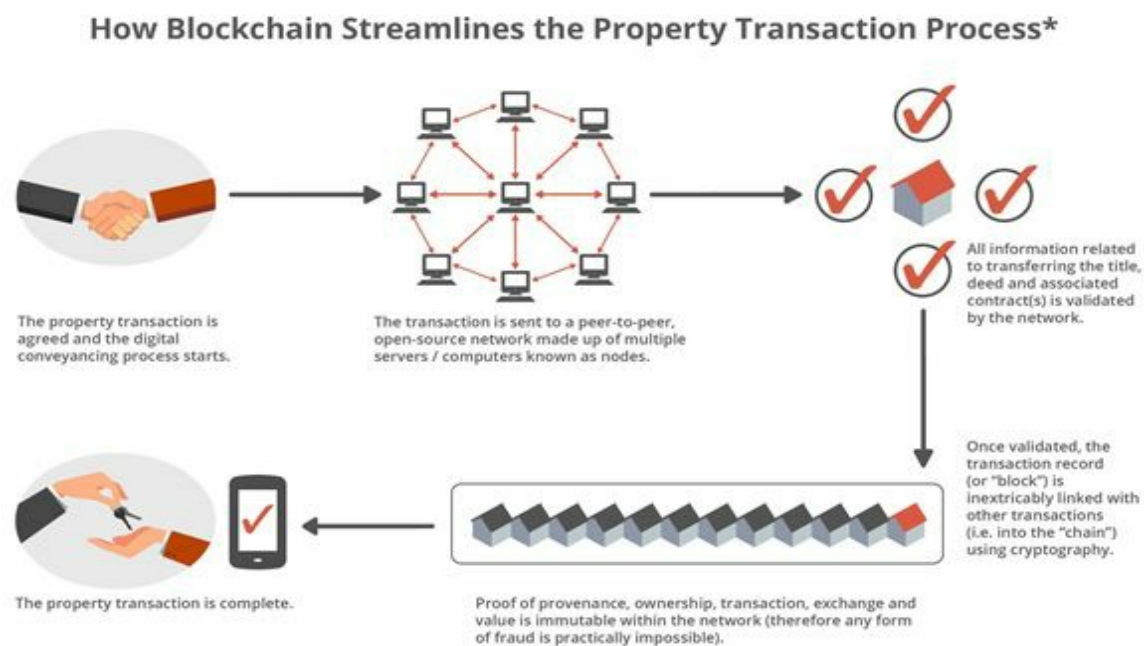
Bitcoin, Ethereum, EOS, IOTA

Language: .Net, Java, C, C++      Solidity, JavaScript, C++

### *Why Blockchain?*

Blockchain is an extremely dynamic, fast growing and incredibly popular emerging technology in current market. Very soon Blockchain technology will replace most of the middlemen by automating business processes and the world will be running on Smart contracts. Blockchain technology can simplify, streamline, automate transactions and revolutionize many industries / sectors like Banking, Finance, Insurance, Logistics, Supply chain management, Medical & Healthcare, Government, Ecommerce, Media and Entertainment, Automotive sector and more.

Here is the Blockchain process for property transactions:



Source: <https://blockgeeks.com/>

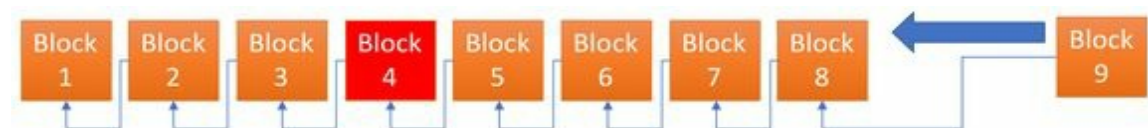
In the above diagram, explains the process of Property transactions. Once both parties agreed the process starts by sending the details to the peer to peer network. Once information sends to Blockchain, miners validate the transactions. Once validation completed, the new block adds to the Blockchain using cryptography method. Once block added in the

Blockchain network, no one individual can alter the information and it is fraud tolerance.

### *How Blockchain works?*

Blockchain is a distributed network and works in Peer to Peer Network with consensus mechanisms. The most popular consensus mechanisms are PoW (Proof of Work) and PoS (Proof of Stake). The other types of consensus mechanisms are Delegated Proof of Stake (DPoS), Proof of Importance (PoI), Multisignature/Byzantine Fault Tolerance (BFT), Federated Byzantine Agreement (FBA), Proof Of Capacity (POC), Proof Of Elapsed Time (POET) etc.

Bitcoin uses PoW and Ethereum uses PoS consensus mechanism to validate its transaction before add to the Blockchain network. Consensus mechanisms are most important in Blockchain in order to function correctly. Its responsibility is to make sure everyone use the same Blockchain network. Also it make sure that each node in the network are connected with each other and agree on the transactions before add to the Blockchain network.



In Blockchain technology data stored with previous block Hash (Cryptographic method). It makes hard for anyone to tamper the data in Blockchain network.

When Bitcoin was released in 2009, people did not know how to use it and where it can be implemented, except booking keeping transactions. After a decade a unique platform created on the Blockchain network, it is

called Ethereum. Ethereum is the first Blockchain based platform implemented the concept of Smart Contract. We will see more about Smart Contract in our next chapter.

### **3. Overview: Smart Contracts**

Smart contracts were first proposed by Nick Szabo, an American computer scientist in 1994, 14 years before the Bitcoin was introduced.

In future there will be very less middle man between buyers and sellers compared to what we have at this moment. The reason for this is *Smart Contracts*. What is smart contract?

Let me explain Smart contract in simple way.

We buy drinks at super market by exchanging fiat currency with the store keeper. We also can be able to buy drinks with an automated vending machine available on the street, which eliminate the super market store keeper. In Vending machine, we need to insert coin, choose our drinks and the vending machine validates our coin and deliver drinks. In this process, machines validate our money and deliver the items, without any middle man (super market). The same concept applies for smart contract, it eliminates the middle man completely by doing the validation by a computer program and runs on decentralized & distributed Blockchain network.



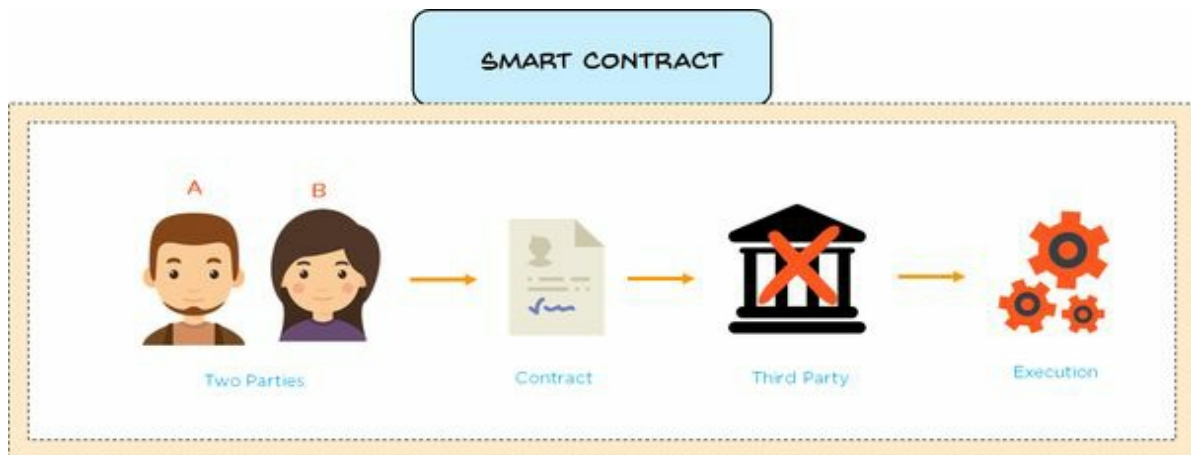
Smart Contracts are a self-operating computer program that runs with set of terms and conditions in decentralized Blockchain technology and once deployed it executed automatically and unable to change it (immutable).

Once Smart contracts are executed, no one can be altered or deleted. If you need any changes in executed Smart contract, it should be executed as a new contract. Existing one cannot be altered.

Let's see how a traditional contract and a smart contract works!



Source: <https://www.simplilearn.com/>

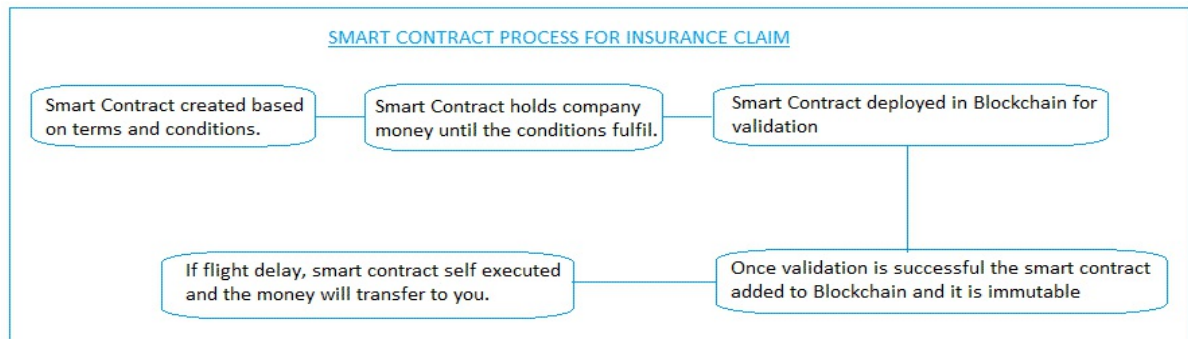


Source: <https://www.simplilearn.com/>

For example, you bought insurance from an insurance company by agreeing set of terms and conditions when you travel from India to U.S. Somehow, the flight delayed, and you should apply your claim with that insurance company. Once you submitted your documents, the insurance company verify your documents and process your payment. The issue here is, the insurance company may delay your payment or unable to pay your claim as per your contract. This issue can be solved by Smart Contracts.

In Smart Contracts, terms and conditions are written in computer program code and deployed in decentralized Blockchain network. This code is immutable. Smart contracts hold the company's money until the condition fulfil. If the flight delay happens the smart contract executed itself and the

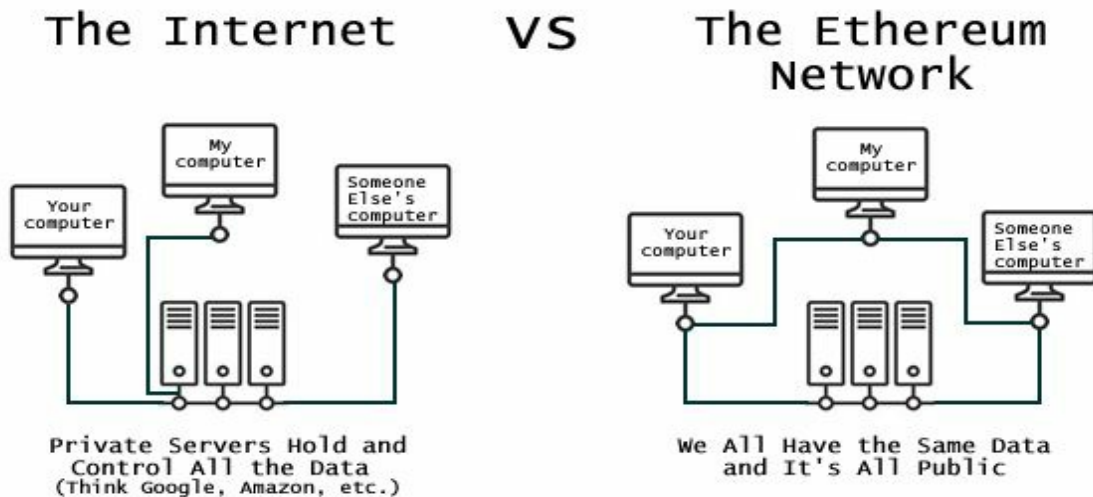
insurance claim money automatically send to your account.



Smart contracts can be written in many platforms or frameworks like Ethereum, EOS, NEO, Zilliqa, IOST etc. Ethereum is a premier in developing Smart contracts. We will be discussing more about Ethereum in our next chapter.

## 4. Overview: Ethereum

While the Bitcoin blockchain used to track ownership of digital currency (Bitcoin), the Ethereum blockchain focuses on running the programming code of any decentralized applications, it is called *Smart Contracts*. In Blockchain technology, Bitcoin is known by everyone which is first digital currency in the Blockchain market. The other most popular platform in Blockchain is Ethereum. And, other few platforms are available in the market that are, Zilliqa (ZIL), EOS, NEO and IOST. Ethereum platform has the capability to run the smart contracts, which is lack in Bitcoin.



<https://datarootlabs.com/executives-guide-smart-contracts-on-blockchain/>

Ethereum uses Blockchain technology to add and validate transactions like Bitcoin and it has own currency called, “Ether”. In order to run Smart contracts in Ethereum Blockchain, you should pay Ether as a transaction fee. This is called “Gas”.

*Ethereum was created by Vitalik Buterin, a young programmer who was told about bitcoin by his father and decided to create a similar platform for smart contracts; which bitcoin is not designed to do. The Moscow-native began working on Ethereum after he dropped out of college, according to [CNBC](#).*

*“Blockchain is an Extension of Cryptography...” – Vitalik Buterin.*

## Coins & Tokens:

In Blockchain technology, the words Coins & Tokens are often used as both are same, but in real these are two different things. Coins hold the currency value (Bitcoin, Ether, EOS, NEO) and it operate independently while Tokens are representing a company's share. Tokens are valid within the project only. You can buy token with a coin, but you cannot buy coin with token.

## Wallets:

Wallets are secured digital place to store and manage your coins and tokens. These digital wallets are used to store your bitcoin, ether, EOS, NEO. There are many wallets available in the market to store your like Bitcon, Ethereum, Jaxx, Exodus etc.

## Solidity:

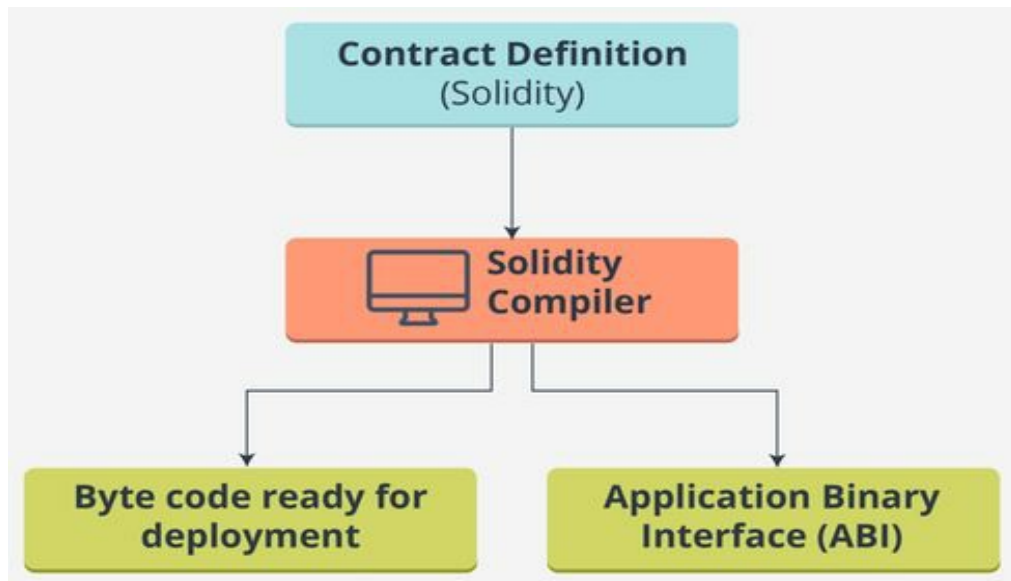
Solidity is a high-level contract-oriented programming language used to code Smart contracts which use Ethereum Virtual Machine (EVM) to convert the users written code into binary format which is understandable by Ethereum Blockchain network.

In Ethereum Blockchain network, Smart contracts should be submitted as Ethereum specific binary format. For this Solidity programming language uses "Ethereum Virtual Machine" to convert Solidity programming language code in to binary format. Solidity is one of the most popular language used to write Smart contract code in Ethereum blockchain network. There are other languages available for coding such as Serpent and LLL.

Here's how Solidity code works

Contract logic is written in solidity language and compiled by the solidity compiler (EVM) which generates two outcomes.

1. Byte code
2. ABI (Application Binary Interface)



ABI is acting as API (middle man) between the machine code and the user interface, with data encoding scheme. This ABI function is to encode/decode the data in/out of machine code. This ABI is used in js files to call functions in a contract and get data back.

Byte code is used in Blockchain network to read the contract code. It is a machine readable format code.

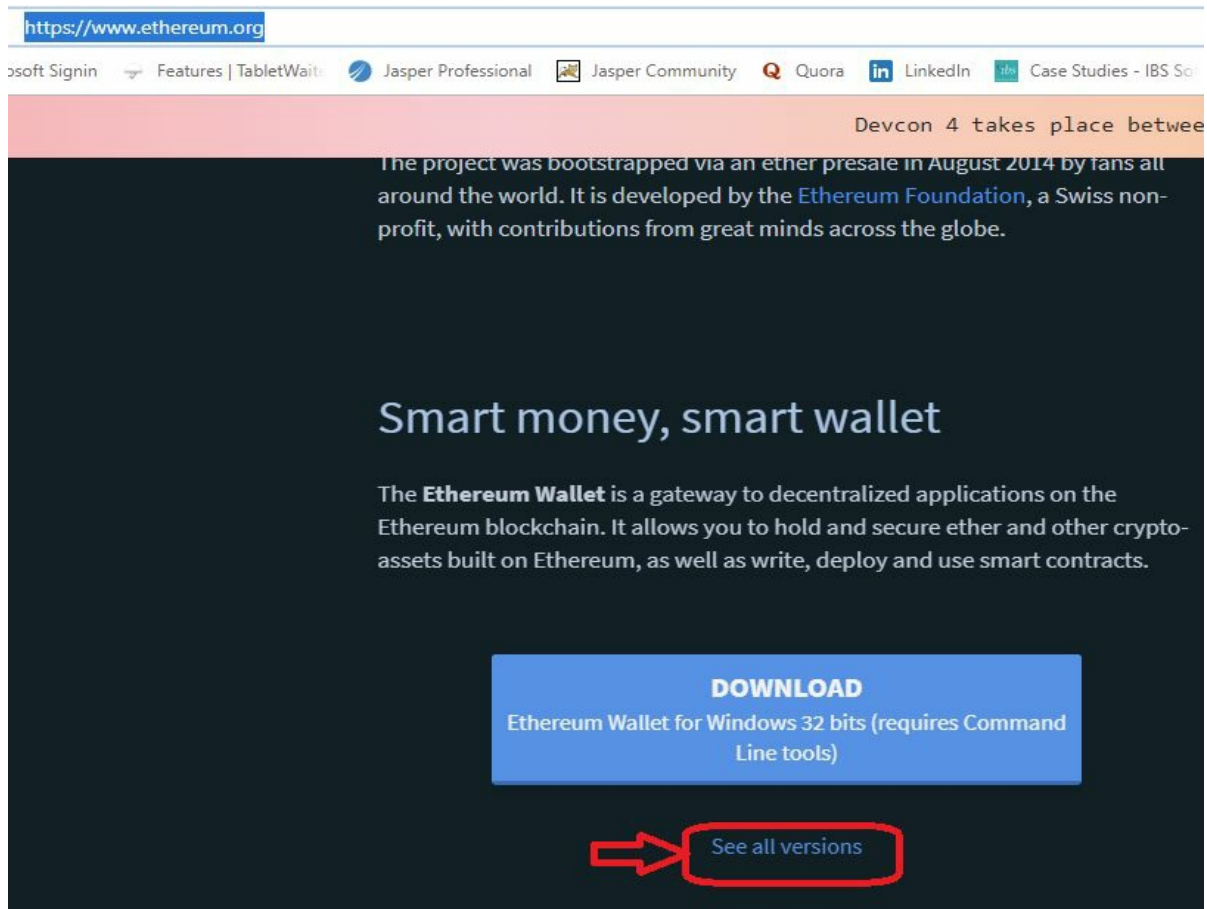
## 5. **Exercise: Ethereum Wallet – Ropsten Test Network**

In this exercise you learn the following

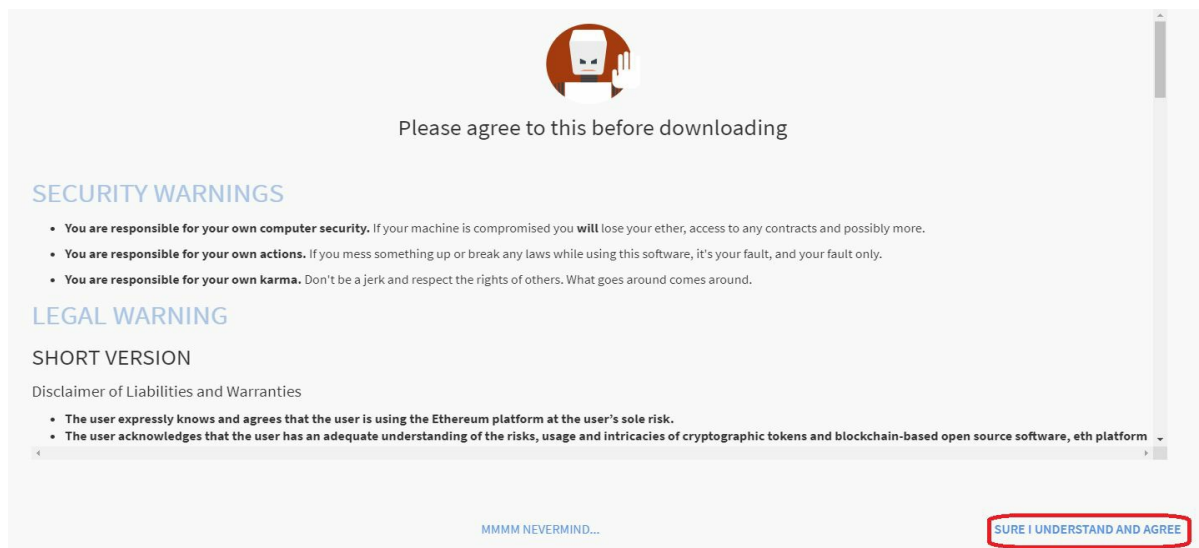
- a) Download Ethereum Wallet:
- b) Creating Account
- c) Get Test Ether
- d) Transfer Ether

### a) Download Ethereum Wallet:

1. Open any browser and go to <https://www.ethereum.org/>
2. Scroll down, you can “DOWNLOAD” button. Click “See all versions”.



### 3. Click “SURE I UNDERSTAND AND AGREE”



### 4. You can see list of Ethereum Waller for different platforms.

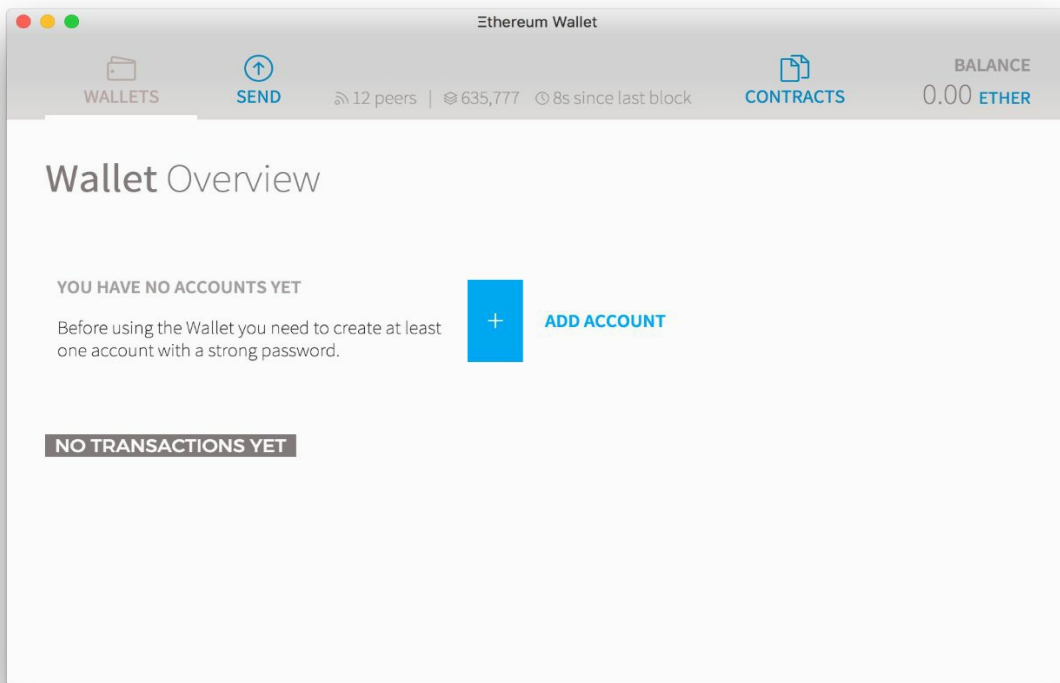
## Ethereum Wallet and Mist Beta 0.11.1 - windows hotfix

evertonfraga released this on Jul 24 · 153 commits to master since this release

▼ Assets 18

Ethereum-Wallet-installer-0-11-1.exe	127 MB
Ethereum-Wallet-linux32-0-11-1.deb	43.8 MB
Ethereum-Wallet-linux32-0-11-1.zip	65.5 MB
Ethereum-Wallet-linux64-0-11-1.deb	42.2 MB
Ethereum-Wallet-linux64-0-11-1.zip	63.4 MB
Ethereum-Wallet-macosx-0-11-1.dmg	67.2 MB
Ethereum-Wallet-win32-0-11-1.zip	59.7 MB
Ethereum-Wallet-win64-0-11-1.zip	67.4 MB
Mist-installer-0-11-1.exe	126 MB
Mist-linux32-0-11-1.deb	43.8 MB
Mist-linux32-0-11-1.zip	64.9 MB
Mist-linux64-0-11-1.deb	42.1 MB
Mist-linux64-0-11-1.zip	62.9 MB
Mist-macosx-0-11-1.dmg	67.1 MB
Mist-win32-0-11-1.zip	59.1 MB
Mist-win64-0-11-1.zip	66.8 MB
Source code (zip)	
Source code (tar.gz)	

5. Click the version of Ethereum Wallet you want to download
6. Unzip the downloaded file.
7. Double click the “Ethereum Wallet” folder and find the .exe file called “Ethereum Wallet”
8. Double click the “Ethereum Wallet.exe” file.

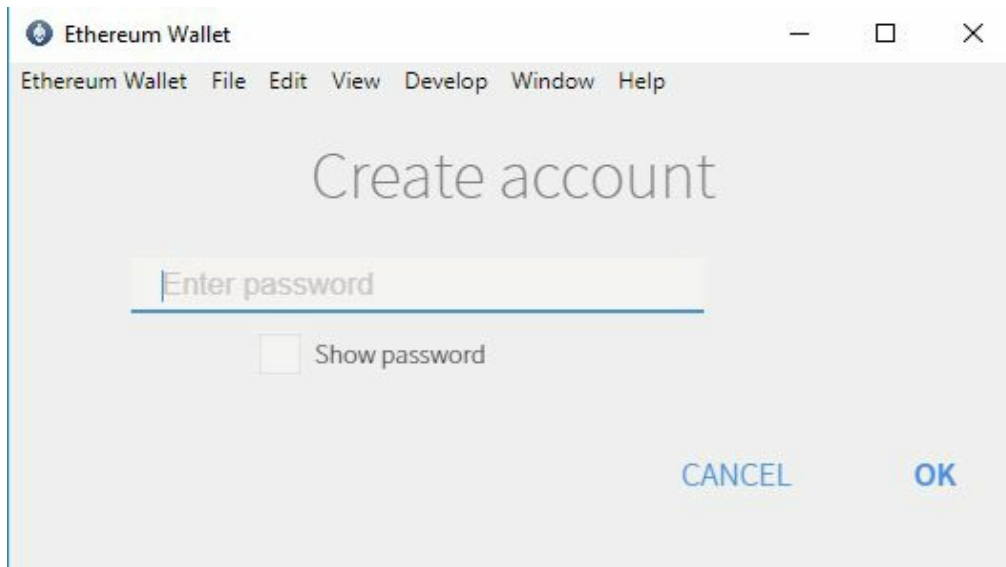


9. Now you can see the Ethereum Wallet without an account.

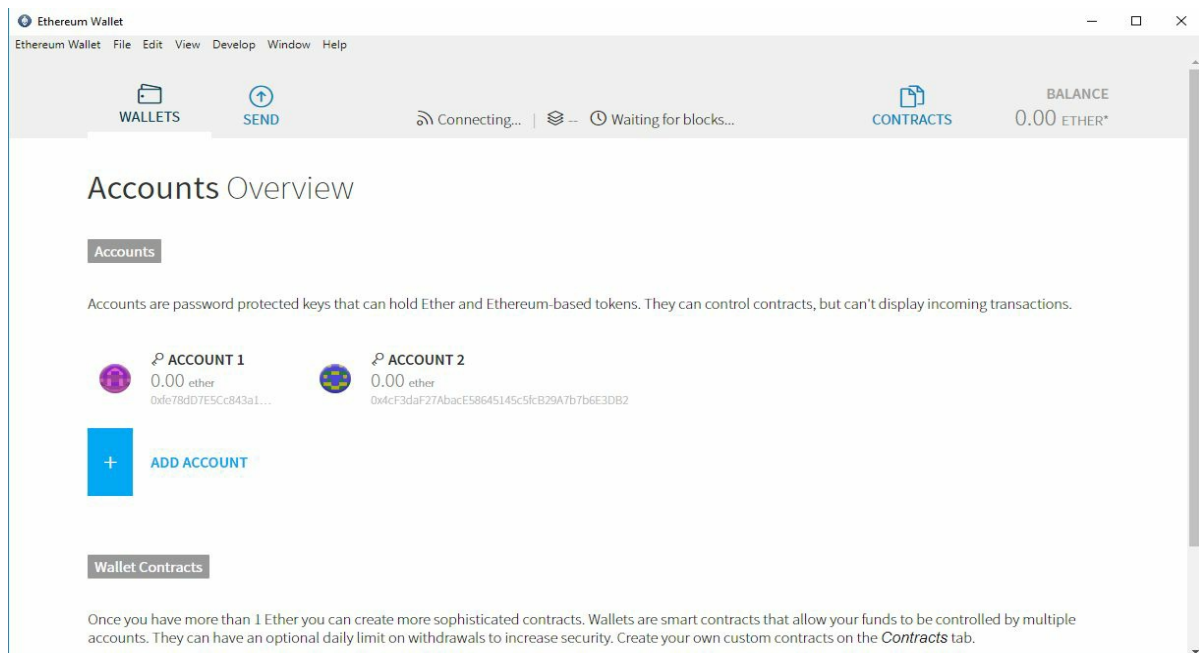
b) Creating Account:

1. Open Ethereum Wallet by double click the “Ethereum Wallet.exe” file.
2. Click “ADD ACCOUNT” button.
3. Type your password for the account and click “OK”

Note: Keep this password in safe place. Whenever do transaction in this network need this password for authentication. Once you lost this password, you will lose all your ether.

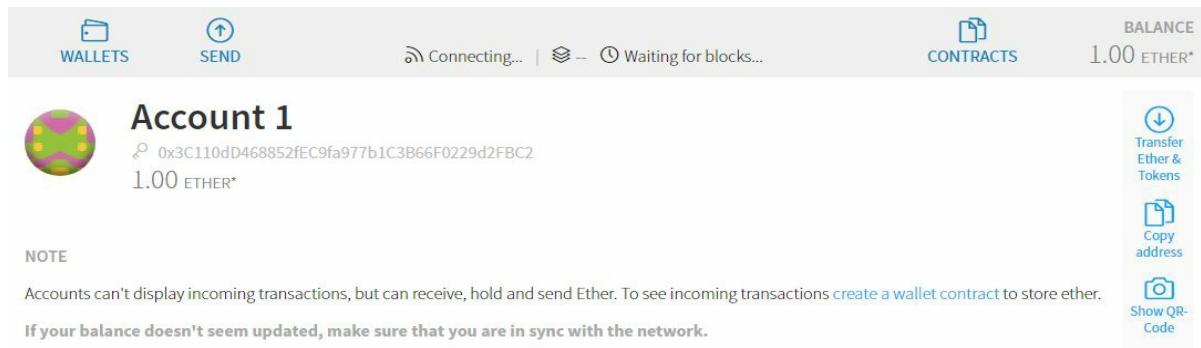


4. Retype your password and click “OK”
5. Account created with the name of “ACCOUNT 1”. You can change this name, if you want.



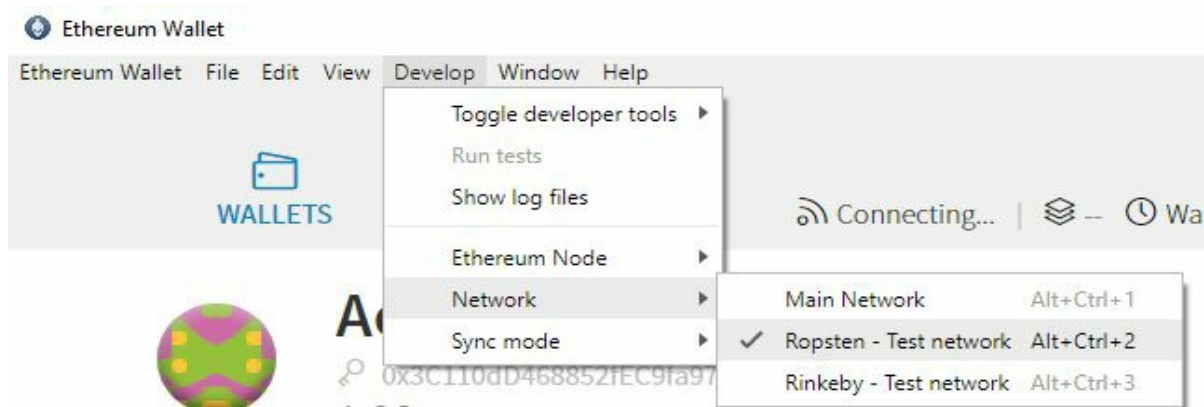
6. Create another account by repeating the steps 1 to 3.
7. Now you can see another account called “ACCOUNT 2”

Note: To view the account address, click the account name.

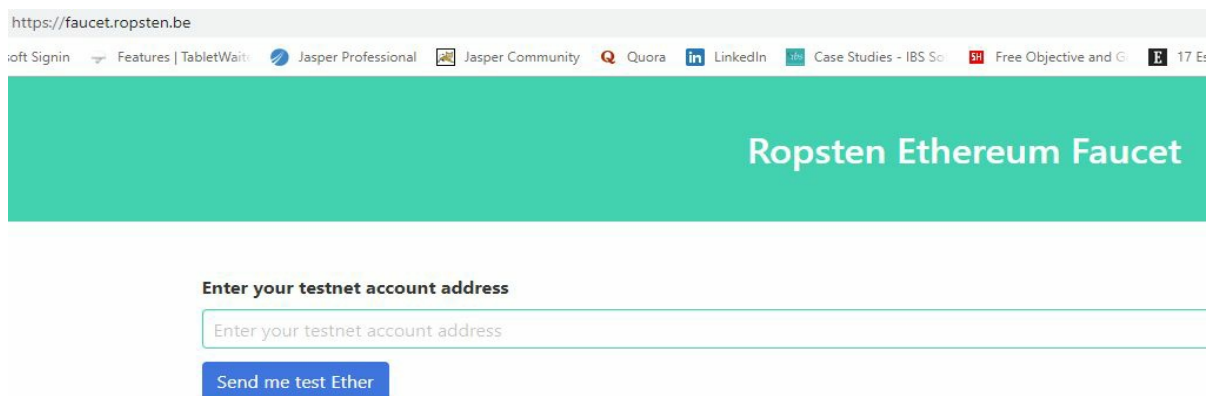


### c) Get Test Ether:

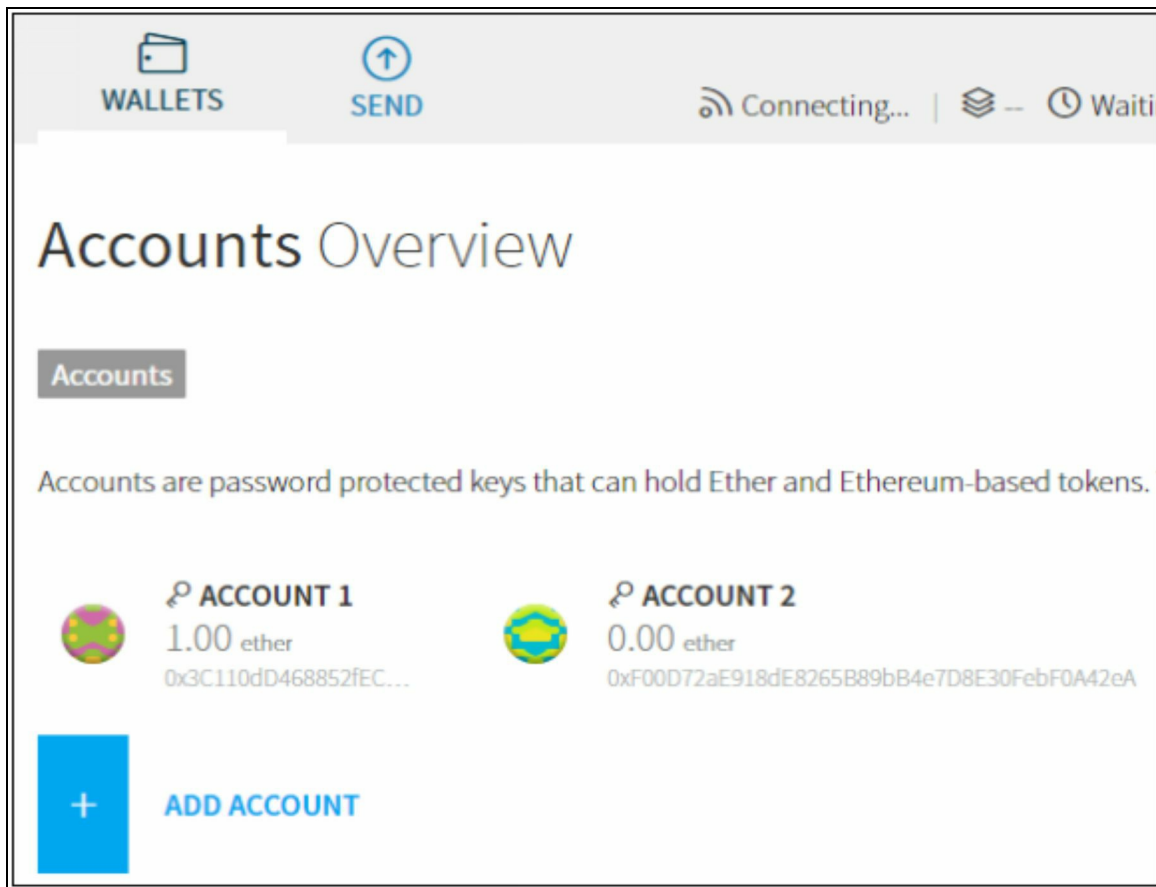
1. Open your Ethereum Wallet. (Double click 'Ethereum Wallet.exe' from downloaded folder)
2. Make sure "Ropsten – Test network" is selected under main menu: Develop -> Network -> Ropsten – Test network.



3. Copy your Account 1- Address from Ethereum Wallet
4. Open web browser and type <https://faucet.ropsten.be/>



5. Type or Paste the copied address in the text box.
6. Click “Send me test Ether”
7. You will be receiving 1 Ether into your Account 1.



d) Transfer Ether:

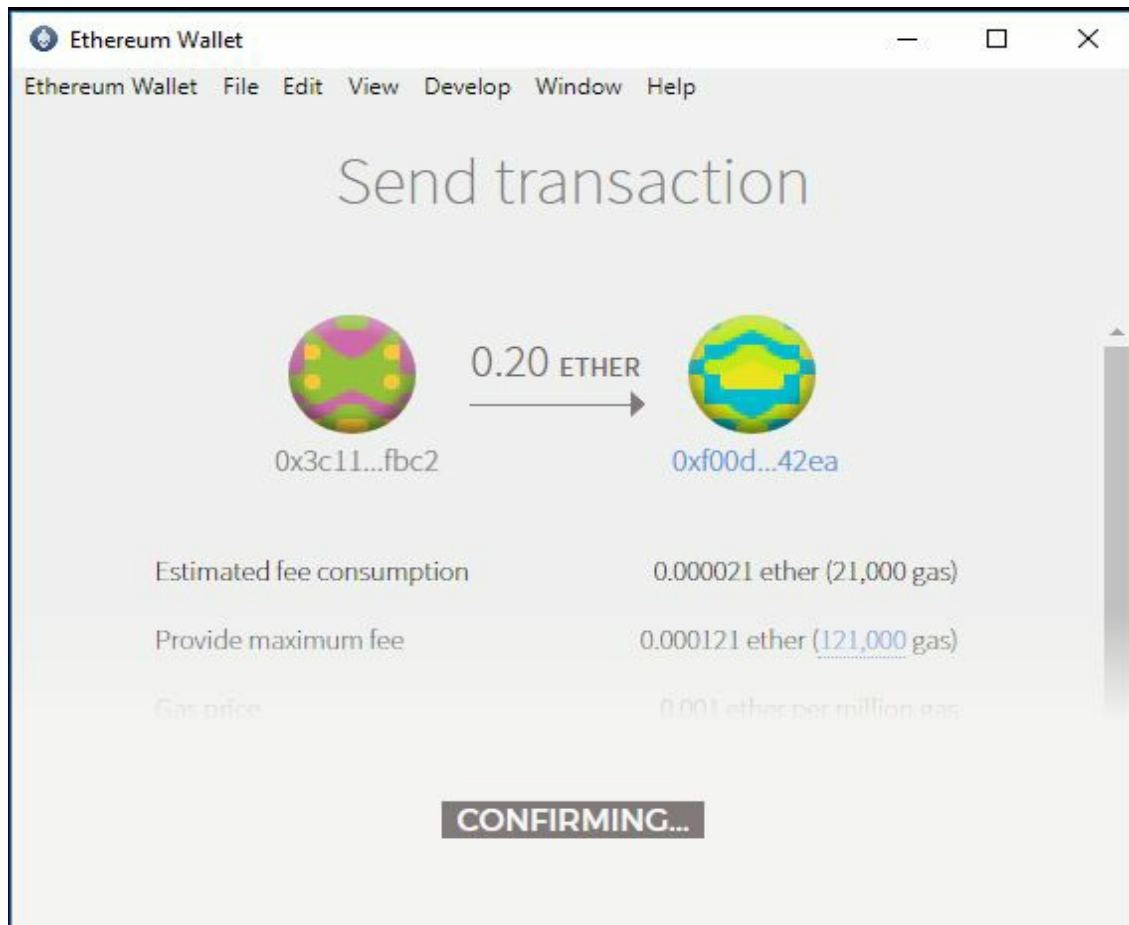
1. Open Ethereum Wallet
2. Click “Account 2” and copy the address.
3. Click “SEND” from the top.

The screenshot shows the 'Send funds' screen of a wallet application. At the top, there is a navigation bar with 'WALLETS' and 'SEND' (highlighted with a red box). The status bar shows 'Connecting...' and 'Waiting for blocks...'. The balance is '1.00 ETH'. The main form has 'FROM' (Account 1 - 1.00 ETH) and 'TO' (0x000000...) fields. The 'AMOUNT' is set to '0.0' with a unit selector for 'ETHER'. A 'Send everything' checkbox is present. A summary line states 'You want to send 0 ETH'. A 'SHOW MORE OPTIONS' button is at the bottom left.

4. Now you can see the “FROM” account as your Account 1 and type or paste the “Account 2” address in “TO” text box.
5. Type the amount for example 0.02 and click “SEND”

This screenshot shows the fee selection and total amount section. Under 'SELECT FEE', the current fee is '0.006 ETH'. A slider allows adjustment between 'CHEAPER' and 'FASTER'. Below this, the 'TOTAL' amount is displayed as '0.206 ETH'. A large blue 'SEND' button is at the bottom.

6. It will prompt for password. Please type the “Account 1” password and click “OK”.
7. Once transaction is completed, 0.2 Ether will be added to “Account 2”

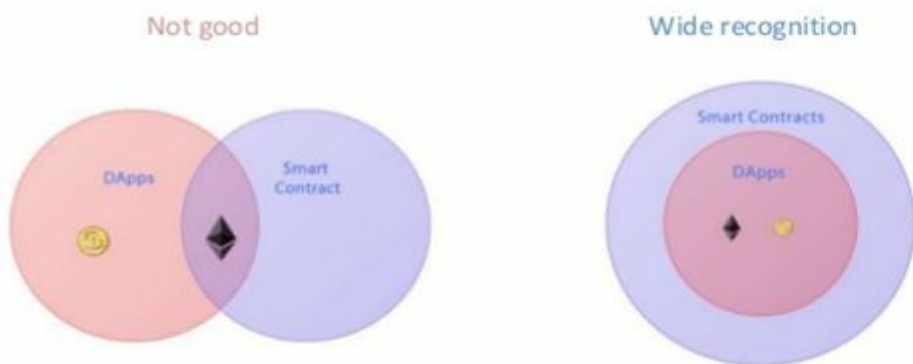


In the crypto currency world, this represents your wallet address (public key) and your private key is what lets you authorize transfers, withdrawals, and other actions with your digital property like crypto currencies. As an aside, this is why it's so important to keep your private key safe — anyone who has your private key can use it to access any of your digital assets associated with your public key and do what they want with it!

## 6. Relationship between DApps & Smart contracts

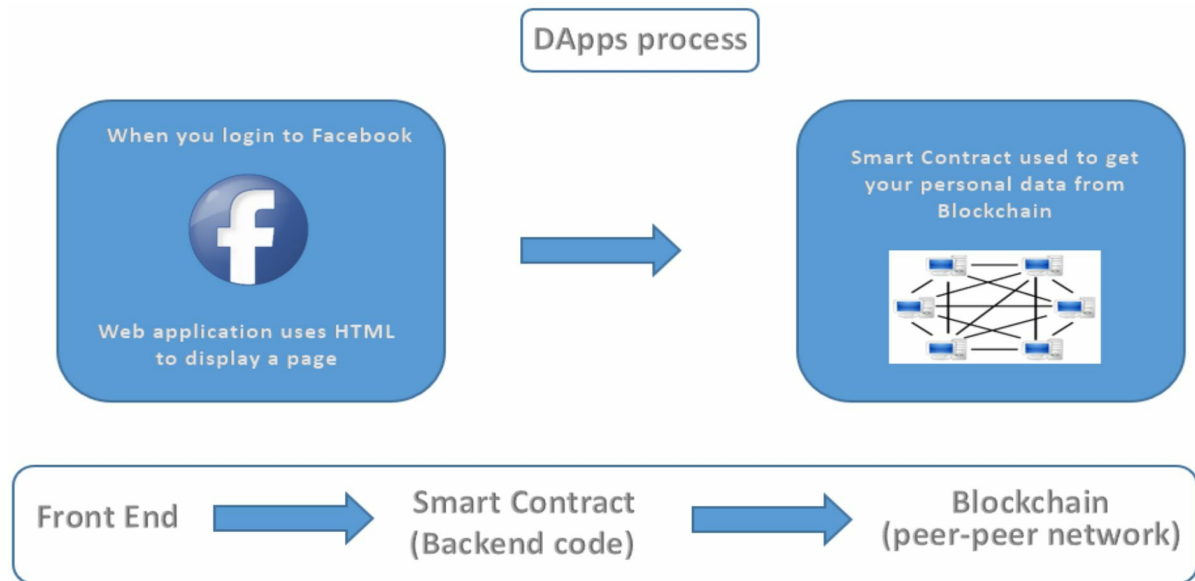
The best examples for DApps are Bitcon and Ethereum. In fact, Bitcon is the first Decentralized Application developed live with Blockchain network. When we talk about Bitcon and Ethereum, some people do not understand the concept behind it. The underlying concept is Cryptographic algorithm. Blockchain is using this Cryptographic algorithm to do the transparent transactions with peer to peer network.

### The relationships between DApps and Smart contracts ?



DApps consists of Front End, Smart Contract and Blockchain. It is a full concept of any application. Whereas smart contract is part of it.

Smart Contracts are basically a computer program runs on top of Blockchain network.



## 7. Overview: Solidity



Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs which govern the behaviour of accounts within the Ethereum state.

Solidity was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM). Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.

With Solidity you can create contracts for uses such as voting, crowdfunding, blind auctions, and multi-signature wallets.

The sample code for Solidity as below:

## Storage

```
pragma solidity >=0.4.0 <0.6.0;

contract SimpleStorage {
    uint storedData;

    function set(uint x) public {
        storedData = x;
    }

    function get() public view returns (uint) {
        return storedData;
    }
}
```

You can read more detailed documentation about Solidity in the below site.

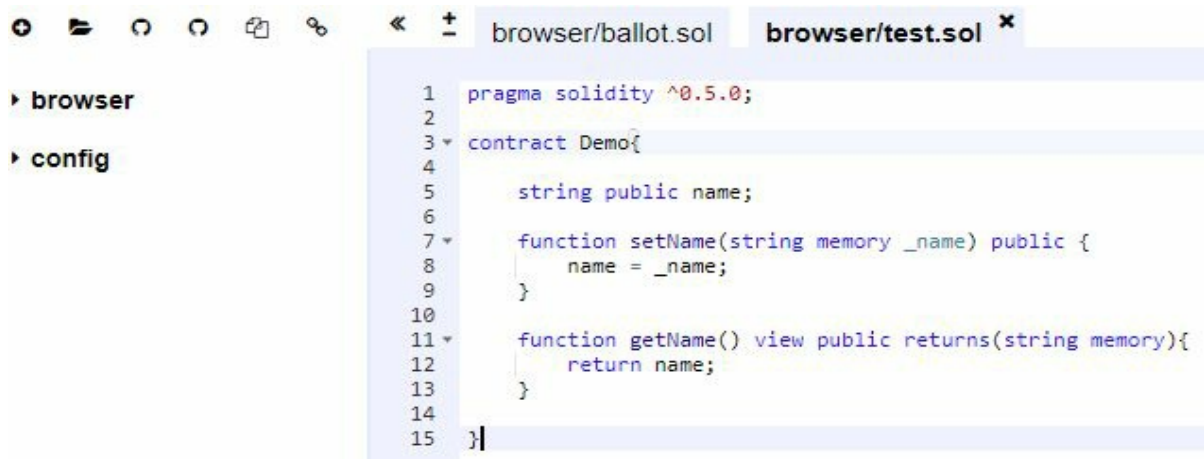
<https://solidity.readthedocs.io>

## 8. Exercise: Deploy simple Smart Contract

We use remix site (<https://remix.ethereum.org>) to test a simple Smart Contract. Remix is an open source and powerful tool that helps to write Solidity code in Browser. It supports to test the code, debug and deploy the smart contracts.

Create Smart Contract, Compile & Run:

1. Open <https://remix.ethereum.org> (in Google Chrome Browser)
2. Create new file by clicking '+' from top left-hand side.
3. Type file name as "demo.sol" and click "OK"



The screenshot shows a web-based IDE with a file explorer on the left containing 'browser' and 'config' folders. The main editor displays 'browser/test.sol' with the following Solidity code:

```
1 pragma solidity ^0.5.0;
2
3 contract Demo{
4     string public name;
5
6     function setName(string memory _name) public {
7         name = _name;
8     }
9
10    function getName() view public returns(string memory){
11        return name;
12    }
13 }
14
15 }
```

4. Add the below code

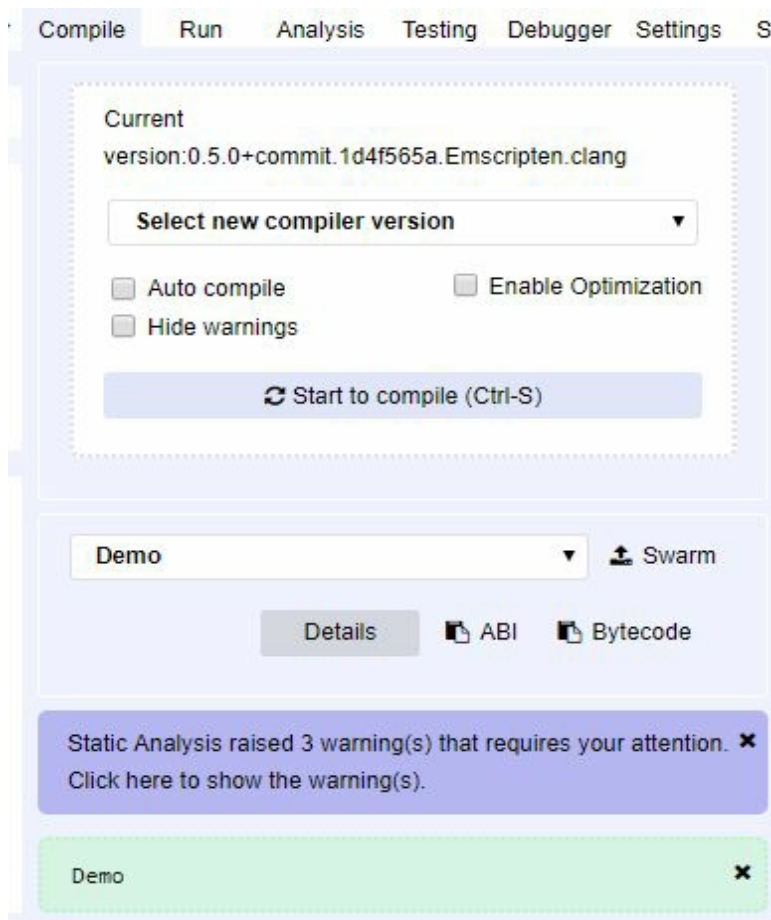
```
pragma solidity ^0.5.0;
contract
    uint
    uint
    function getNumbers uint    uint    public

    function addNumbers    view public returns uint
```

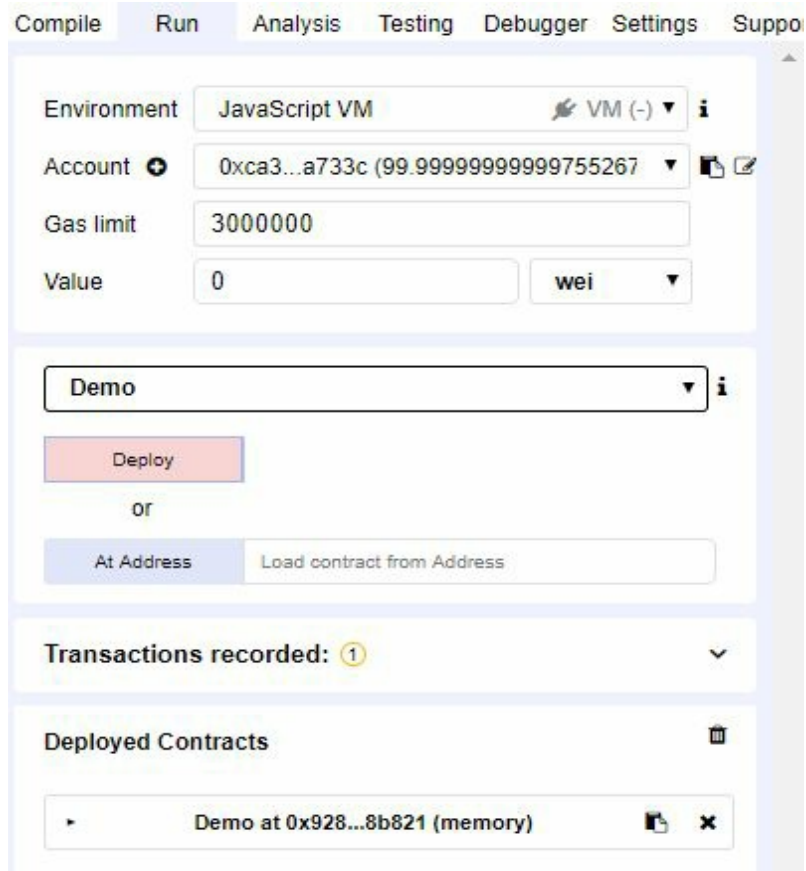
5. Click “Start to compile” button.

6. Once code compared, the contract name will be displayed on the screen.

**Note:** Select the compiler version as indicated in code (ex: 0.5.0) by clicking the select box “Select new compiler version”.



7. Deploy the Smart contract, by clicking “Run” tab and “Deploy” button.
8. Once contract successfully deployed, it will be displayed in the “Deployed Contracts” section.

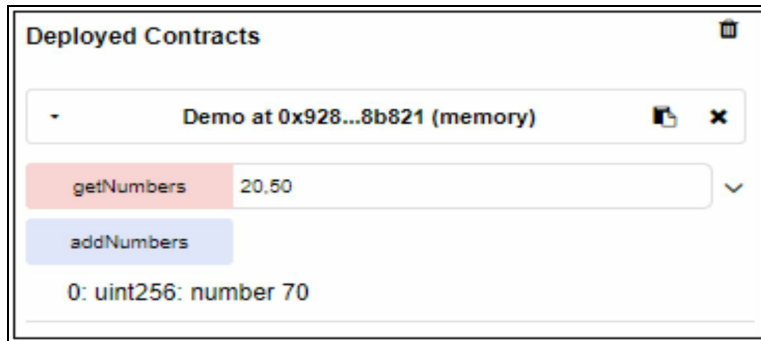


9. Click “>” symbol from the Deployed Smart contract. It will display the parameters to key-in and get the results like below.



10. To test the contract key in any positive integers number separated by comma like 20, 50 and click “getNumbers” button.

11. To get the results click “addNumbers” button.



With this, we deployed a simple smart contract to add 2 integer values.

## 9. Pre-requisition to build DApps

## Requirements:

- Node.js
- NPM
- Test RPC

## Technologies using

- Frame-work            Web3.js
- Test network        Test RPC
- Scripting language    Solidity
- Compiler            Remix Browser

Download and install the latest version of Node.js. It includes the NPM installation.

<https://nodejs.org/en/download/>

To check if you have Node.js installed, run the following command in your terminal:



To check if you have npm installed, run the following command in your terminal



Goal for this chapter: Setup Test Network (Test RPC)

### Setup Test Network (Test RPC):

1. Open command prompt with Admin permission.
2. Change directory to “C:\dapp\”
3. Install test rpc by typing: npm i -g ganache-cli
4. Start test rpc by typing: ganache-cli

```

C:\WINDOWS\system32>ganache-cli
Ganache CLI v6.2.3 (ganache-core: 2.3.1)

Available Accounts
=====
(0) 0xa3fddd5ac26c511a3bcc7d4f99a28dd9bc266c03 (~100 ETH)
(1) 0x834de21fb816d06665af60daaa6a12833d4ebb85 (~100 ETH)
(2) 0x2e14d90b6b3c0ea9d225e9186c269826a505c60b (~100 ETH)
(3) 0xafb323fce8aa3326c5553867fbd6372fc35aa1db (~100 ETH)
(4) 0x215b808451a290e084b729cd8db7441d5eae3fd3 (~100 ETH)
(5) 0xc4984f6413c9cf71244f68ecaa7d74ce88a6f900 (~100 ETH)
(6) 0x491d40b8f98a907871ce36ebde7829897182a93d (~100 ETH)
(7) 0x8e395c502436459ccc1cb02150ebac8b110bdf36 (~100 ETH)
(8) 0xb50a8b930caf178895b4a00346ef73c17cc5be82 (~100 ETH)
(9) 0x284f672f4131f8cd9b63f5382d4e6e2eef1525c1 (~100 ETH)

Private Keys
=====
(0) 0x0302bf56eeb9912a45b892bde0f329c4080b6f77c2ccd3c20ba2fad1aed24a2e
(1) 0x077ee17687690c8d9448058a3a9af33d617234a6058fcecbb8aebec15e67ce1
(2) 0xfca0fadeef6bdbfa1324c4d5b8fa170c49b210a27840ecb35d52fdec462baffe
(3) 0x06870cee74437e74dc3430f7951950902d0aca99e012475069b1cfd8cd81817b
(4) 0xe1323ab1cc5ee76874b9c7d5edb634d86a391f32ec08cd70ca22ccfab9c8aaef
(5) 0x74ca9a77f1909c2d7537e5d7d06123c16207b7c8744640b93da349655afdd6e1
(6) 0x8f50bb23cab3f8af3da6cf11e30007b94a041b21198b405bbd9e622f1da710f0
(7) 0xfaf0f77ae8fe782e217f3442ddc872e663b0236dd7b93b78894270efd41a4423
(8) 0x4410c926d63d6a712cfb9df6a593de11a692489c6c6ed62216e595c21c260a43
(9) 0x584fcd73b87537c35d9d2cfca2c6cbd14c0e4252b4815f99f2e3d18a3058d155

HD Wallet
=====
Mnemonic:      topic fork brand prefer tongue unlock ring estate subway humor vote sting
Base HD Path:  m/44'/60'/0'/0/{account_index}

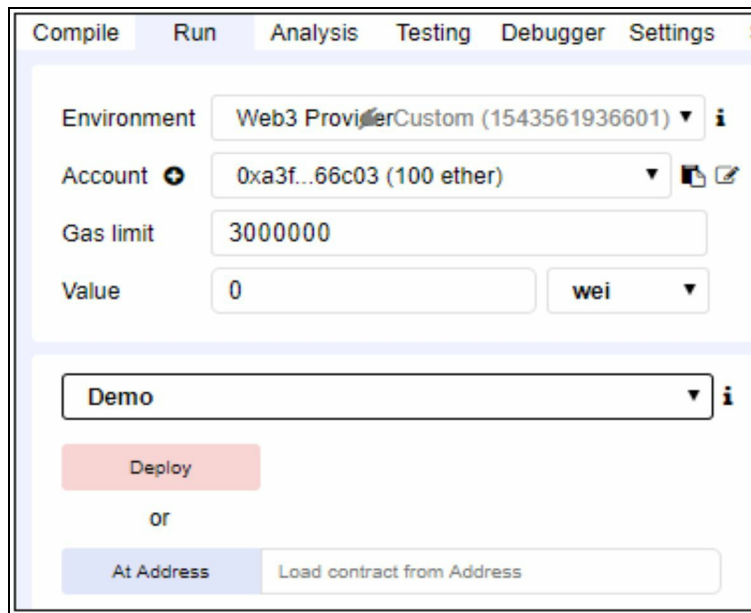
Gas Price
=====
20000000000

Gas Limit
=====
6721975

Listening on 127.0.0.1:8545
net_listening
eth_getBlockByNumber
net_version
eth_accounts
net_version

```

5. Test network started with 10 accounts (Public key and private key).
6. Go to browser <https://remix.ethereum.org> -> “Run” tab.
7. Change the Environment to “Web3 Provider”.



Now the test network is ready and starts code to build the DApp.

## 10. Exercise: Build your first DApps

Goal for this chapter:

- a) Create a Smart Contract with Solidity in Remix site.
- b) Deploy the contract to a private blockchain network.
- c) Build a front-end app that interacts with the contract using the Web3 library.
- d) Test the first DApp.

### a) Create a Smart Contract with Solidity in Remix site

For smart contract, we use the same what we have created in Chapter no. 8. (Exercise: Deploy simple Smart Contract). Open your chrome browser and go to <https://remix.ethereum.org> and paste the code copied from below.

```
pragma solidity ^0.5.0;

contract

    uint

    uint

    function getNumbers uint    uint    public


    function addNumbers    view public returns uint
```

- b) Deploy the contract to a local blockchain network.

Web3.js is a Ethereum Javascript API. It is a collection of libraries which allows interacting with local Ethereum node, using http.

Make sure that you have running Test RPC, if not run by the following command.



- i) In Remix site, select Environment as “Web3 Provider” under “Run” Tab.
- ii) Click “Deploy”.

c) Build Front-End App

1. Go to C drive
2. Create folder dapp under c:\
3. Initialize the json files by typing the command: “npm init”
4. Install framework: Npm install –save Ethereum/web3.js
5. Create “index.html” under c:\dapp with the following code.

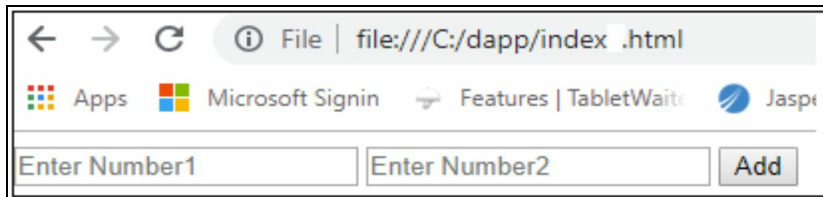
```
<html>
<head>
  <title>Smart contract</title>
  <script type="text/javascript"
src="C:/dapp/node_modules/web3/dist/web3.js"></script>
</head>

<body>
  <div>
    <input id="input1" type="number" name="number1" placeholder="Enter Number1"/>
    <input id="input2" type="number" name="number2" placeholder="Enter Number2"/>

    <button id="buttonAdd">Add</div></button>
    <pre id="output"></pre>
```

```
</div>
</body>
</html>
```

6. To test the code open browser and type “C:/dapp/index.html”. You will see the html page as below.



7. Now we are going to connect this html page with smart contract with local blockchain environment.  
Please refer to the complete code as below. You can get this code from github. <https://github.com/thaniindia/blockchain/tree/master/DApp>

```
<html>
<head>
  <title>Smart contract</title>
  <script type="text/javascript" src="C:/dapp/node_modules/web3/dist/web3.js"></script>
</head>

<body>
  <div>
    <input id="input1" type="number" name="number1" placeholder="Enter Number1"/>
    <input id="input2" type="number" name="number2" placeholder="Enter Number2"/>

    <button id="buttonAdd">Add</button>
    <pre id="output"></pre>
  </div>

  <script type="text/javascript">
    var web3;

    if(typeof web !== "undefined"){
      web3 = new Web3(web3.currentProvider);
    }else{
      web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
    }
    web3.eth.defaultAccount = web3.eth.accounts[0];

    var new_contract = web3.eth.contract([
    {
      "constant": true,
      "inputs": [],
      "name": "addNumbers",
```

```

        "outputs": [
            {
                "name": "number",
                "type": "uint256"
            }
        ],
        "payable": false,
        "stateMutability": "view",
        "type": "function"
    },
    {
        "constant": false,
        "inputs": [
            {
                "name": "a",
                "type": "uint256"
            },
            {
                "name": "b",
                "type": "uint256"
            }
        ],
        "name": "getNumbers",
        "outputs": [],
        "payable": false,
        "stateMutability": "nonpayable",
        "type": "function"
    }
]);

var contract = new_contract.at("0xbd1629e8df06fb98aded4224da926fd028f24c69");

buttonAdd.addEventListener("click",function() {
    contract.getNumbers(input1.value,input2.value);
    output.innerHTML = contract.addNumbers();
})

</script>

</body>
</html>

```

Let me explain this code for each section.

The below code is setting environment as local blockchain (Test RPC). The Test RPC is running at the port 8545.

```
var web3;

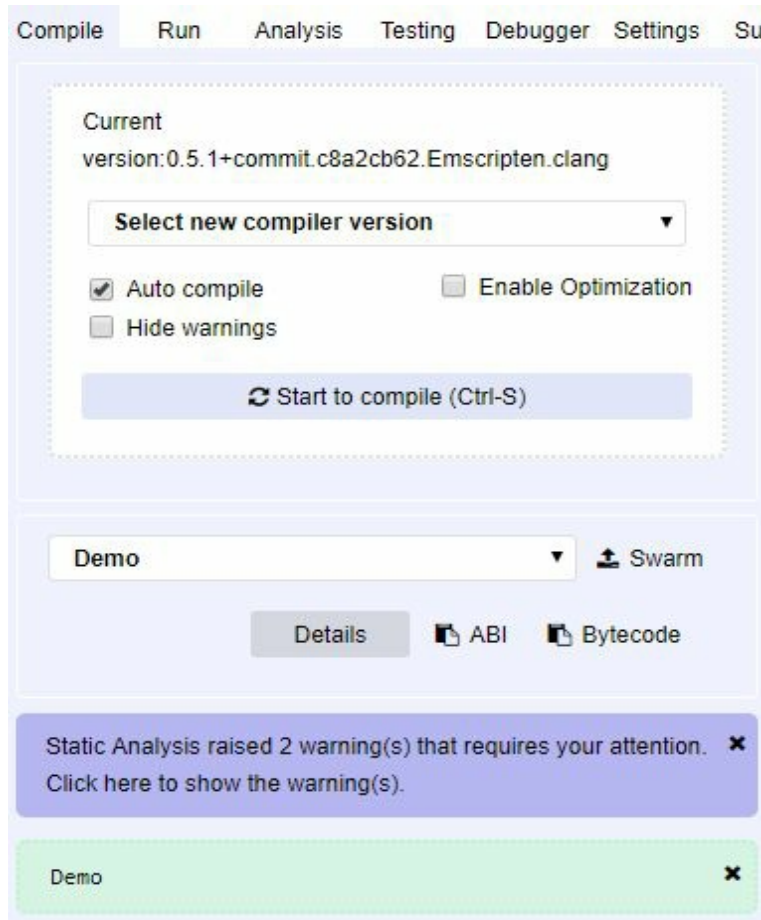
if(typeof web !=="undefined"){
    web3 = new Web3(web3.currentProvider);
}else{
    web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
}
```

Set the default account to run the contract. By Default, Test RPC creates 10 accounts.

```
web3.eth.defaultAccount = web3.eth.accounts[0];
```

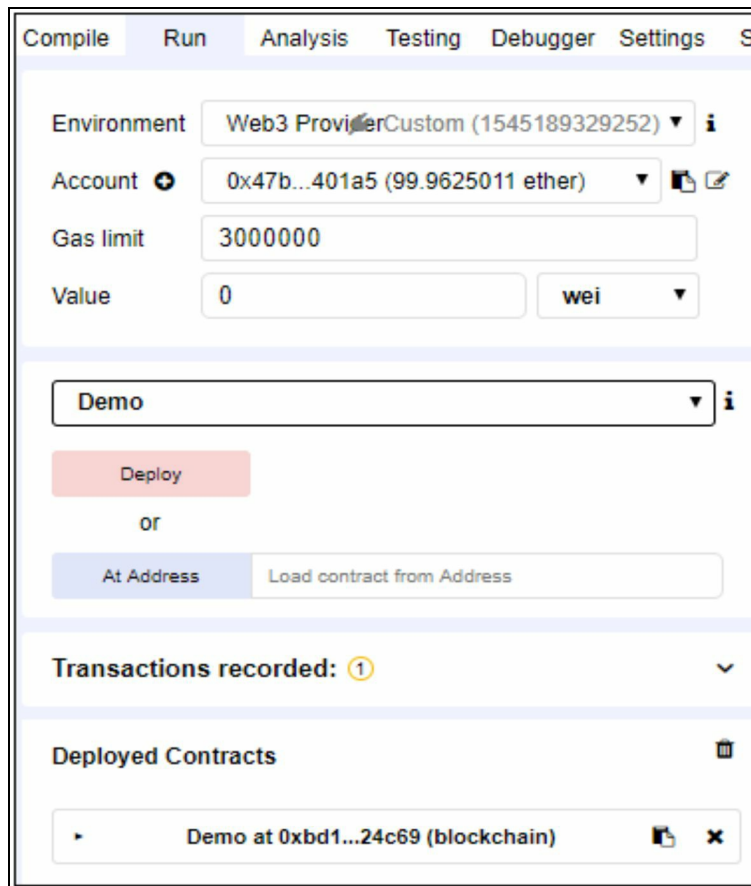
In below code, need to input contract ABI (Application Binary Interface). ABI tells about how to interact with the contract. This ABI created once the contract compiled successfully. Copy the ABI code and paste inside the contract parenthesis.

```
var new_contract = web3.eth.contract();
```



Below code, need to input the contract address. Once contract deployed successfully you can get the contract address from remix site under “Run” tab. Copy the contract address and paste in the below code.

```
var contract = new_contract.at("0xbd1629e8df06fb98aded4224da926fd028f24c69");
```

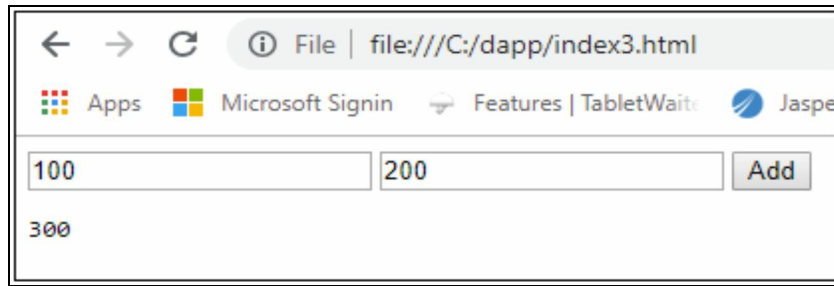


Add Button click event code as below.

```
buttonAdd.addEventListener("click",function() {
    contract.getNumbers(input1.value,input2.value);
    output.innerHTML = contract.addNumbers();
})
```

8. Copy the above complete code in index.html file.
9. Replace the “ABI” & “contract address”
10. Successfully deployed contract in local Blockchain network.
- e) Test the first DApp

1. Open a Chrome Browser, and type url as “C:/dapp/index.html”
2. Type number in Number1 & Number2 text box
3. Click “Add” button.
4. The result will be displayed as below.



## 11. Debug

Once contract deployed in mainnet, the [www.etherscan.io](https://www.etherscan.io) block explorer can be used to view internal state and EVM execution logs.

Right click on Chrome browser and select 'Inspect'. Click on Console tab to view the error messages.

Use Events log. Read the below code included the event handling. Read the logs under once contract executed under logs section in Remix IDE.

```
pragma solidity ^0.5.0;

contract

    uint

    uint

    event          string          uint
function getNumbers uint    uint    public
    emit

    emit


function addNumbers    public returns uint

    emit
```



## Common Error messages & solutions

**Error:** [Attempting to run transaction which calls a contract function, but recipient address is not a contract address.](#)

**Solution:** Set Environment in browser as: Web3 Provider.

**Error:** Uncaught Type Error: Cannot read property 'providers' of undefined.

**Solution:**

```
web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
```

**Error:** Uncaught TypeError: Cannot read property 'call' of undefined

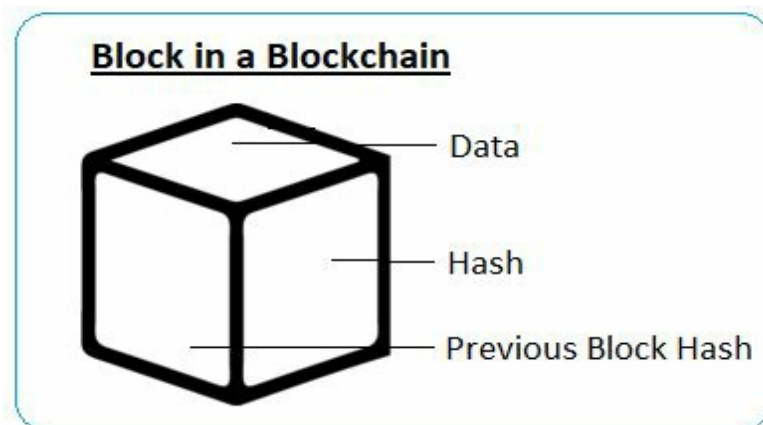
**Solution:** Solidity code has changed. Copy the ABI and paste it in .js file.

Note: Whenever the contract code changed and deployed, need to copy the ABI and contract address in js file.

## 12. Security

The key aspect for any emerging technology is “Security”. The user gets used the new technology, when it proves its reliability and security functions. In IT industry “Security” always gives challenge for any new innovative product. Blockchain use the most trusted and secured ‘Cryptographic’ algorithm to do each transaction in blockchain network. In this chapter let’s explain how the Cryptographic algorithm works.

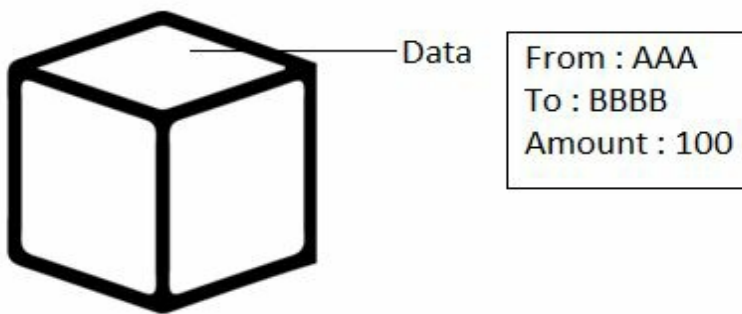
Blockchain works in Peer to Peer Network, is a decentralized network that distribute workloads among peers. In Blockchain technology records are called “*blocks*”. Each block has Data, Hash and Previous Block Hash.



### *Data:*

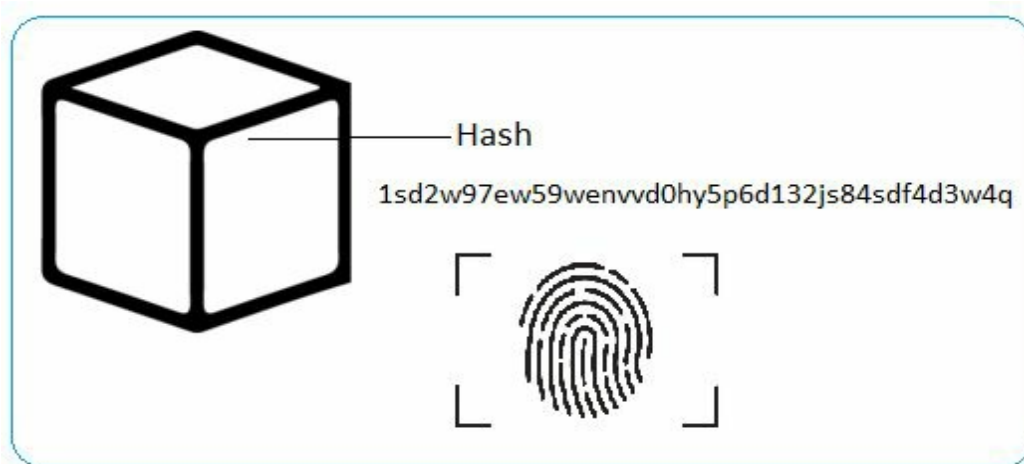
In Blockchain block, data is about information. It can be a set of rules or just a storage file. For example, in Bitcoin it's about from account, to account and the amount.

### Block in Bitcoin



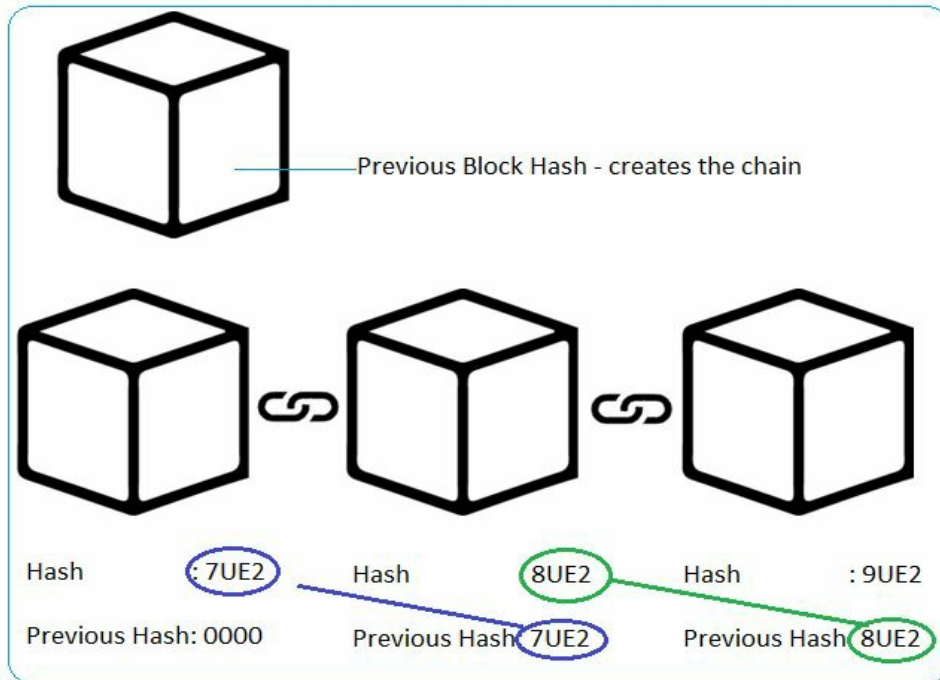
### *Hash:*

Hash is like a fingerprint of a block. It is a unique value for each block. When information changed in a block, the new Hash value will be created, and that modified block cannot be participating in the chain. Thus, it is hard to edit information once block added into the blockchain.

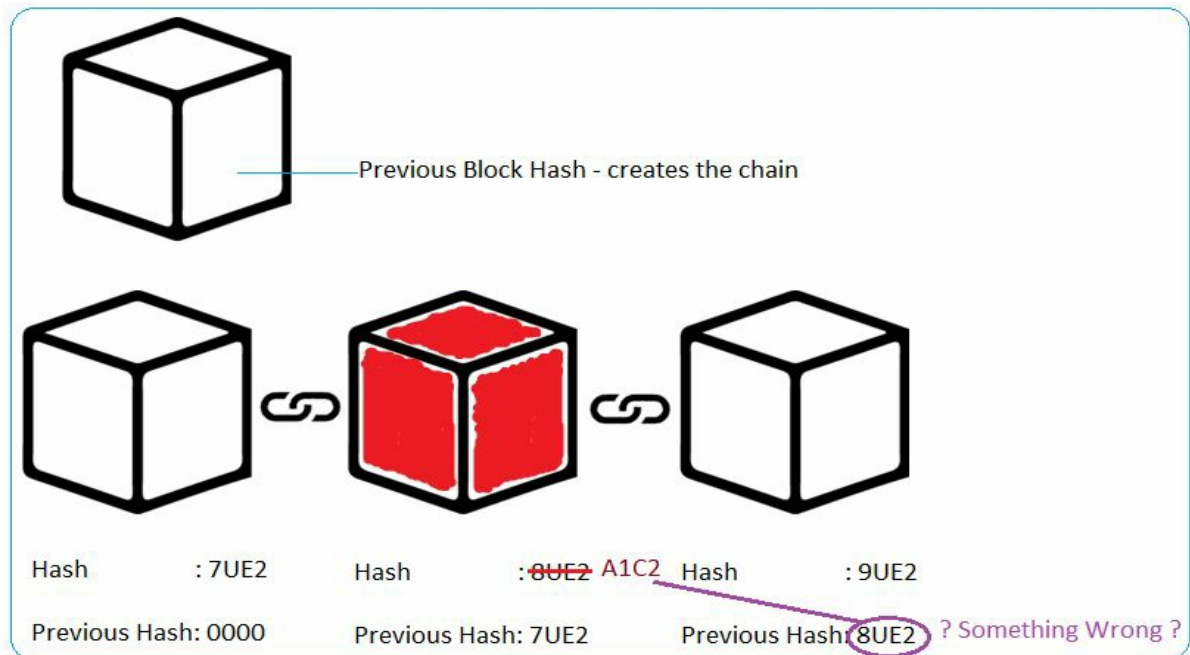


### *Previous Block Hash:*

Each block has a hash of its previous block. In order to alter the information in a block, need to update the new cash with the next block, which is not extremely difficult process in the blockchain. Hash value of first block is 0, since there is no previous block, and this is called “Genesis Block”.



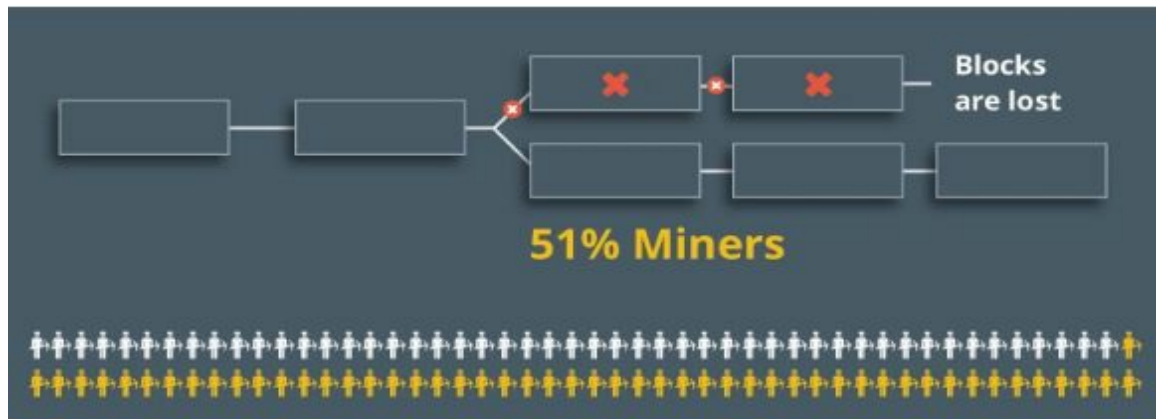
For example, if you try to alter the 2<sup>nd</sup> block information, the hash value “8UE2” changed as “A1C2” and the 3<sup>rd</sup> Block won’t be able to accept the new cash value. So, it’s hard to tamper the block once deployed into blockchain. With huge computational process, by generating a random hash value of the block, one can be able to identify the hash value. Blockchain won’t be secured only with Hash mechanism. To avoid this issue blockchain has something called Proof of Work, it slows down the creation of work. In Bitcoin, it takes 10 minutes to calculate the Proof of Work and add a new block with the chain. It makes the hackers hard to tamper the block, if one block is tampered need to calculate the Proof of Work for the following blocks on the network. Thus, security of Blockchain comes from effective use of hashing and proof of work.



Another way of security in Blockchain is, distributed system. Anyone can join in this distributed network and get the full copy of the Blockchain data. When a new block is trying to add in Blockchain, the new block sends to all nodes in the network for validation and in order to add a new block in a Blockchain need more than 50% of nodes consensus. For this validation process, miners (validators) get a small amount of incentives.

### 51% Attack:

Maintaining consensus on a blockchain network it uses Proof of Work (PoW) algorithm. PoW is the original consensus algorithm in Blockchain network. The most famous crypto currency Bitcon is using PoW algorithm to validate its transactions. The average time to take to solve the mathematical puzzle in PoW is 10 minutes. For this, need high computational power to solve the puzzle and add a transaction in Blockchain network. At this moment, it is impossible to get high computational power to gain 51% access in blockchain network.



But this may change in future with Quantum Computing Technology. Quantum computing powered computers will surpass the limit of today's classic computers. We have no answer at this moment for this.

With the above information you can understand as Blockchain network is having more secured network but when you develop DApps, the smart contract error may cause the whole system collapse. Need to do multiple test, before launch into mainnet.

## 13. Further Reading

The following websites are useful for you to develop your DApps.

Ethereum home page

<https://www.ethereum.org/>

Ethereum official documentation

<http://ethdocs.org/en/latest/contracts-and-transactions/developer-tools.html>

Ethereum white paper

<https://github.com/ethereum/wiki/wiki/White-Paper>

Ethereum github

<https://github.com/ethereum>

Solidity documentation

<https://solidity.readthedocs.io>

List of DApps developed on Ethereum

<https://www.stateofthedapps.com/>

7 DApps build on Ethereum

<https://www.coindesk.com/7-cool-decentralized-apps-built-ethereum>

## 14. Learning Activity

To challenge your knowledge, please try the following projects. The basic smart contract code is available at my github site.

<https://github.com/thaniindia/blockchain>

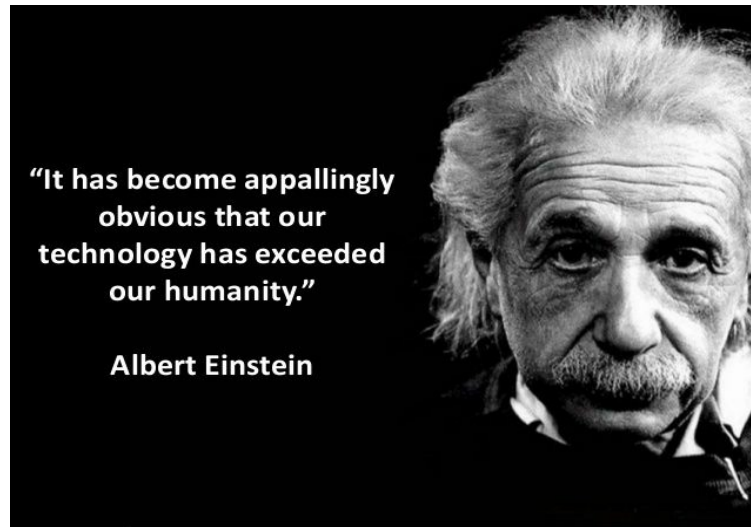
1. Goal: Create database for Do Not Call Registry
  - a. Create user interface to get mobile number
  - b. Add Mobile Number
  - c. Remove Mobile Number.
  - d. List all DNC Registry Mobile Numbers
2. Goal: Customer Form
  - a. Create user interface to get customer details
  - b. Get customer id, first name, last name, mobile no and email
  - c. List all customer details.

## 15. Summary

Innovation is born from taking risks. Blockchain is un-doubtfully a promising breakthrough technology. It has immense business opportunities for developers those are taking challenges and invest their valuable time.

Many people do not aware of DApps value same like many people not aware of Organic food compared to normal food. But when time comes, everyone will look for Organic food. The same thing for Blockchain and DApps, once the industry decision makers knows the value of this emerging technology, then they will come in to this to implement their business process and it becomes popular.

You must be clearly able to understand Blockchain is not for all. Those industry or process need to avoid middle man, need a trusted, immutable and transparent system for their operation, the Blockchain will be useful in this case. To avoid disappointment after developing your DApps for not relevant industry or process, please be careful before choosing to develop your DApps.



All the best for your DApps development journey.



### **About Author**

**THANIKASALAM KRISHNASAMY**

Thanika is a Cisco certified CCNA Professional and has 15 years of IT industry working experience in India and Singapore, completing a bachelor's degree in computer science in India and master's degree in business administration in Heriot-Watt University, UK. During his working experience he has went through the various technical skills like Cloud Computing, IT Infrastructure Network Management, IT Infrastructure Analyst and Software Application Development.

Thanika always keen to learn emerging technologies in this career. He enthusiasm over emerging Blockchain & Hyper Ledger technology and learned with so much excitement and passion. Currently, he is willing to share more with people around the world.

Special Thanks to Blockchain Expert Mr. Farook Ismail, who is my friend and mentor. His guidance for my blockchain journey and this book release is invaluable.

Thanika can be contacted in the following ways:

Email: [thanikakrishna@gmail.com](mailto:thanikakrishna@gmail.com)

LinkedIn: <https://www.linkedin.com/in/thanikasalam/>

Website: <https://attocloud.net/>