



Creative  
Emergy

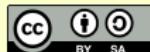
# Introduction to Blockchain

*for George Brown College students*

By Marc Lijour

September 4, 2019

*The Art and Science of Eternal Blossom*



# Who am I?

<https://www.linkedin.com/in/marclijour/>



Access these slides

<https://bit.ly/2lJbAnB>

or find in the folder **GBC - Blockchain Developer Certificate** at

<https://github.com/marclijour/presentations>



# Table of Contents

1 What is Blockchain?

2 Hands-on Introduction to Crypto

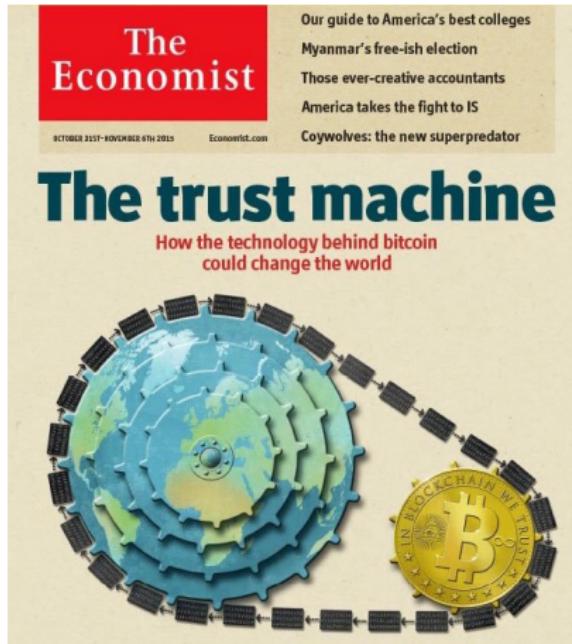
3 Create your own token

- The easy way
- The programmatic way

4 Introduction to Ethereum



## The Trust Machine



# Source of Trust

World Economic Forum: *What is Blockchain* Youtube Video

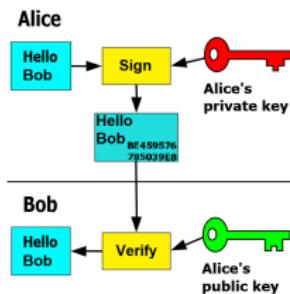


BLOCKCHAIN WILL BECOME A GLOBAL  
DECENTRALISED SOURCE OF TRUST



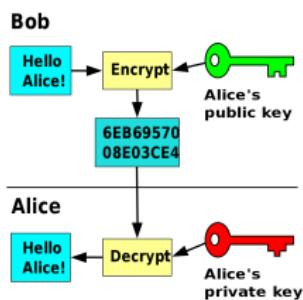
# Key Cryptographic Primitives

## Signing



Credit: FlippyFlink

## Encrypting



## Hashing

Input	cryptographic hash function	Digest
Fox		DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D1A
The red fox jumps over the blue dog		00B6 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EEE6 3ABC
The red fox jumps over the blue dog		8FD8 7558 7851 4F32 D1C6 76B1 79A9 DCA4 AEF8 4819
The red fox jumps oevr the blue dog		FC03 TFD8 5AF2 C6FF 915F D401 C0A9 7D9A 46A7 FB45
The red fox jumps over the blue dog		BACA D682 D588 4C75 4BF4 1799 7D88 BCFB 92B9 6A6C



# Where do we need more trust?

- payments



# Where do we need more trust?

- payments
- national identity



# Where do we need more trust?

- payments
- national identity
- supply chain (e.g. food: Spanghero horse meat trial, Opioids, expensive cargos)



# Where do we need more trust?

- payments
- national identity
- supply chain (e.g. food: Spanghero horse meat trial, Opioids, expensive cargos)
- contracts



# Where do we need more trust?

- payments
- national identity
- supply chain (e.g. food: Spanghero horse meat trial, Opioids, expensive cargos)
- contracts
- *real* news & historical events (e.g. bombings)



# Where do we need more trust?

- payments
- national identity
- supply chain (e.g. food: Spanghero horse meat trial, Opioids, expensive cargos)
- contracts
- *real* news & historical events (e.g. bombings)
- collaborative data reporting
- ...



# Table of Contents

1 What is Blockchain?

2 Hands-on Introduction to Crypto

3 Create your own token

- The easy way
- The programmatic way

4 Introduction to Ethereum



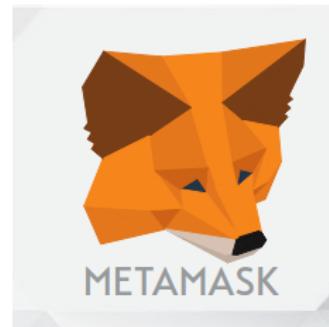
# Let's try things out!



# Install MetaMask

Follow step by step:

- ① Install the [Chrome/Chromium extension](#)
- ② Watch the [intro on Youtube](#)
- ③ Create an account
- ④ Switch to the Ropsten Testnet  
(top-right in MetaMask)
- ⑤ Fill your account with Ether from  
<https://faucet.metamask.io>



<https://metamask.io>



# Request Ether from the faucet (on the Ropsten network)

Do it several times; then donate 1 ether to the faucet

← → ⌛ 🔒 https://faucet.metamask.io

## MetaMask Ether Faucet

faucet

address: 0x81b7e08f65bdf5648606c89998a9cc8164397647  
balance: 4094302.54 ether

request 1 ether from faucet

user

address: 0x56e552eb5a9ab277d9eb841f92d473bf0cae7ebf  
balance: 1.00 ether

donate to faucet:

1 ether    10 ether    100 ether

transactions



# Check the transaction on Metamask

Click on the transaction for a detailed view

The screenshot shows the Metamask interface on the Ropsten Test Network. At the top, it displays "0.9999 ETH" and "\$88.94 USD". Below this, a transaction history shows a single entry: "Sent Ether" from address 0x56E552Eb5a9AB277D9EB841f920473bf0Cae7EBf to address 0x81b7E0BF65Bdf564B606cB9998A9CC8164397647 on March 16, 2018, at 23:23. The status is "CONFIRMED" with "-1 ETH" and "-\$88.95 USD". A "View on Etherscan" button is present. On the left, there's a sidebar with a token icon, "0.9999 ETH", and "\$88.94 USD". It also includes a "Don't see your tokens?" link, an "Add Token" button, and a note to "Click on Add Token to add them to your account".



# Check the transaction on Etherscan

 **Etherscan**  
The Ethereum Block Explorer

ROPSTEN (Revival) TESTNET

Search by Address / Txhash / Block / Token / Ens  Language

HOME BLOCKCHAIN TOKEN MISC

Transaction [0xa8a8f32539cb69bd3e506b6527fcc7d44c43cccead09b30b10a9891c99355a3d](#) 

Home / Transactions / [Tx Info](#)

**Overview**

Transaction Information   Tools & Utilities 

[ This is a Ropsten Testnet Transaction Only ]

TxHash:	0xa8a8f32539cb69bd3e506b6527fcc7d44c43cccead09b30b10a9891c99355a3d
TxReceipt Status:	Success
Block Height:	<a href="#">4610407</a> (348 Block Confirmations)
TimeStamp:	1 hr 12 mins ago (Dec-12-2018 04:23:14 AM +UTC)
From:	0x56e552eb5a9ab277d9eb841f92d473bf0cae7ebf
To:	0xb1b7e08f65bdf5648606c89998a9cc8164397647
Value:	1 Ether (\$0.00)
Gas Limit:	21000
Gas Used By Transaction:	21000 (100%)
Gas Price:	0.000000005 Ether (5 Gwei)
Actual Tx Cost/Fee:	0.000105 Ether (\$0.000000)
Nonce & (Position):	0   {6}
Input Data:	<pre>0x</pre>



# A note about gas price

<https://ethgasstation.info>

ETH Gas Station Estimates over last 1,500 blocks - Last update: Block 5164391 Change Currency ▾

**Std Cost for Transfer** \$0.056 | **Gas Price Std (wei)** 3 | **SafeLow Cost for Transfer** \$0.056 | **Gas Price SafeLow (wei)** 3 | **Median Wait (s)** 29 | **Median Wait (blocks)** 2

**Gas-Time-Price Estimator:** For transactions sent at block: 5164391

Adjust confirmation time: 3

Avg Time (min)	4.38
95% Time (min)	10.95
Gas Price (wei)*	3
Tx Fee (Flat)	\$0.056

Gas Used*	21000
Avg Time (blocks)	18.02
95% Time (blocks)	45.05
Tx Fee (ETH)	0.00005

**Real Time Gas Use:** % block limit (last 10)

Last Block: 5164391

**Transaction Count by Gas Price**

**Confirmation Time by Gas Price**

**Recommended Gas Prices** (based on current network conditions)

Speed	Gas Price (wei)
SafeLow (<30m)	3
Standard (<5m)	3
Fast (<2m)	18

Note: Estimates not valid when multiple transactions are batched from the same address or for transactions sent to addresses with many (e.g. > 100) pending nonce confirmations.

**Misc Stats** (Last 1,500 blocks)



# Price of (real) ether: ETH

More information: <https://www.tradingview.com/symbols/ETHUSD/>

ETHUSD Crypto Chart



# Wallets

More information: <https://blockgeeks.com/guides/cryptocurrency-wallet-guide/>

Software



Hardware



Paper



# Exchanges

- ① Centralized Exchanges (Coinbase, Quadriga, ...)
- ② Decentralized Exchanges

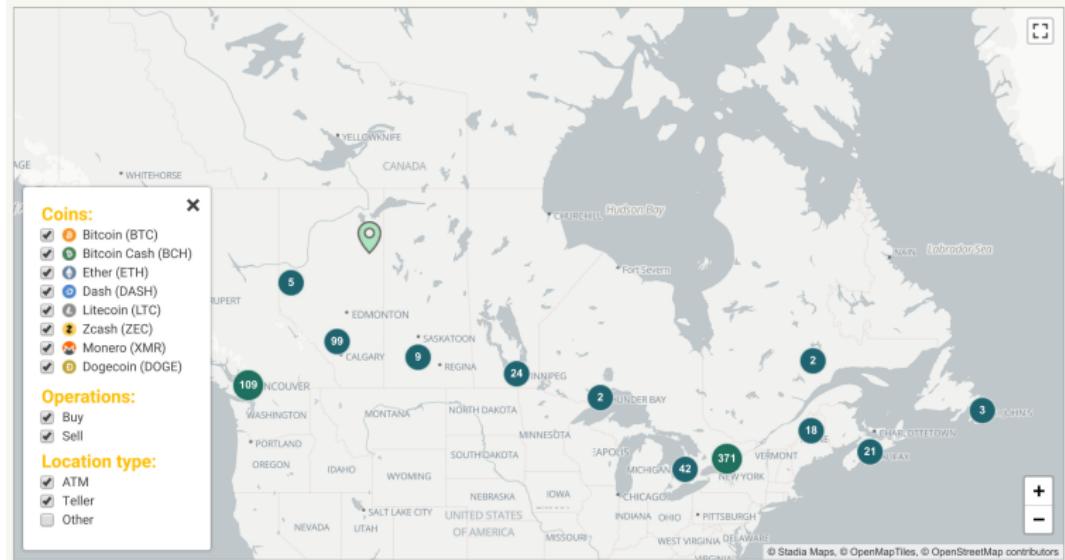


## ATMs

More information: <https://coinatmradar.com/country/38/bitcoin-atm-canada/>

# Bitcoin ATMs in Canada. 🇨🇦

Total number of Bitcoin ATMs / Tellers in Canada: 707

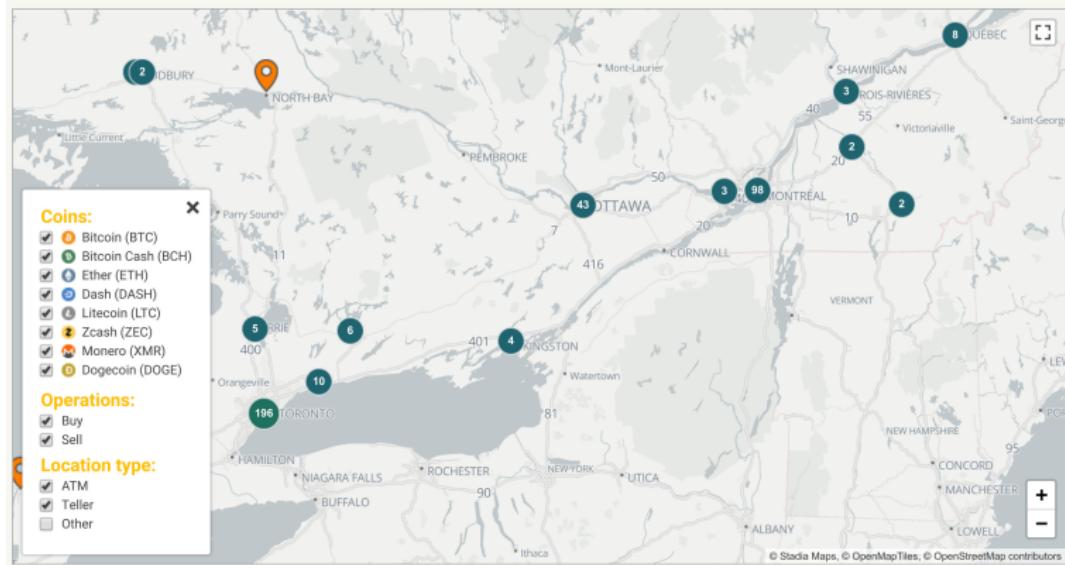


# ATMs

More information: <https://coinatmradar.com/country/38/bitcoin-atm-canada/>

## Bitcoin ATMs in Canada . 🇨🇦

Total number of Bitcoin ATMs / Tellers in Canada: 707



# Other crypto-assets

More information:

<https://www.tradingview.com/markets/cryptocurrencies/prices-all/>

NAME 301 matches	SEARCH	MKT CAP	FD MKT CAP	LAST	AVAIL COINS	TOTAL COINS	TRADED VOL	CHG %
Bitcoin		59.852B	72.158B	3436.1	17,419M	21M	4.712B	0.02%
XRP		12.167B	29.729B	0.29729	40,927B	100B	369.224M	0.17%
Ethereum		9.257B	9.257B	89.23	103,745M	101,745M	1,739B	0.10%
Stellar		2.165B	11,811B	0.11297826	19,165B	104,543B	197.41M	-0.34%
Tether		1.838B	2.554B	0.99000000	1,856B	2,58B	2.97B	-0.10%
Bitcoin Cash		1.729B	2.074B	98.760	17,506M	21M	71.709B	0.92%
EOS		1.692B	1.879B	1.8672	906,245M	1,006B	740.947B	0.44%
Bitcoin SV		1.548B	1.858B	88.45300000	17,505M	21M	59.813M	-2.60%
Litecoin		1.431B	2.02B	24.045	59,532M	84M	380.518B	1.14%
TRON		877.572M	1.314B	0.013247	66,247B	99,218B	72.465B	-0.88%
Cardano		772.388M	1.341B	0.029791	25,927B	45B	9.766B	-0.09%
Monero		722.858M	722.858M	43.445	16,638M	16,638M	11.502B	1.38%
NEM		642.663M	642.663M	0.07141	9B	9B	8.23M	-1.53%
Binance Coin		635.009M	926.299M	4.8548350	130,799M	190,799M	20.772B	-2.52%
IOTA		631.509M	631.509M	0.2272	2.78B	2.78B	4.203M	1.07%
Dash		554.925M	1.234B	65.28257000	8.5M	18.9M	160.684B	2.12%
Ethereum Classic		402.997M	402.997M	3.777	106,698M	106,698M	106.521M	0.68%
NEO		387.602M	596.31M	5.96	65M	100M	100.484M	1.51%



# Table of Contents

1 What is Blockchain?

2 Hands-on Introduction to Crypto

3 Create your own token

- The easy way
- The programmatic way

4 Introduction to Ethereum



# Let's create your own token!



# Create your own (ERC-20) token

## Create Token

Create Token Contract with the following parameters.

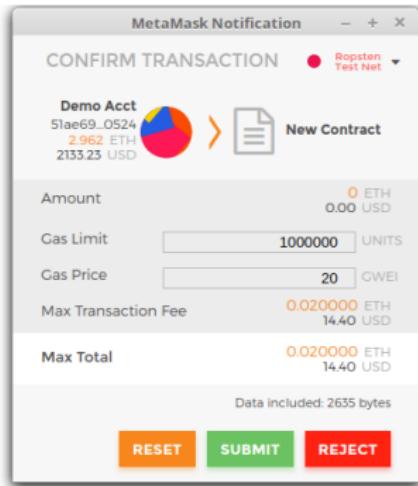
100

Marc's Coin #2

8

MLD

Create Token



- 1 Use the Token Factory Dapp at <https://tokenfactory.surge.sh/#/factory>
- 2 MetaMask will pop up (see picture above)
- 3 Submit the transaction (on the Ropsten Testnet)
- 4 Check your transaction on <https://ropsten.etherscan.io>



# Check your Smart Contract

SENT                    TOKENS

2      December 29 2017 04:49  
Contract Published      0 ETH

- ① Select the “Sent” tab
- ② Check the orange Copy icon (Tx Hash)
- ③ Click on “Contract Published”
- ④ That should bring you to Etherscan (see next page)



Verify the status of your transaction on Etherscan

Transaction Information: note the "To" line with your contract address



# Watch your Token



- ① Click on the “Add Token” button
- ② Wait for the next window (picture on the right)
- ③ Copy your contract address (from Etherscan)
- ④ Go back to your Token Factory tab, which should show an UI to interact with your contract or go to the URL:  
<https://tokenfactory.surge.sh/#/token/0x...> (replace 0x... by your contract address)
- ⑤ Move coins around
- ⑥ In MetaMask, click on your token to check the tx on Etherscan



# Too easy?

Let's code it in Solidity  
like the pros!



# Compile your first ERC-20 Smart Contract

The screenshot shows the Remix IDE interface. On the left, there's a sidebar with compiler settings: Compiler (0.5.11+commit.c082d0b), Language (Solidity), and EVM Version (compiler default). Below these are buttons for 'Compiler Configuration', 'Auto compile', 'Enable optimization', and 'Hide warnings'. At the bottom of the sidebar is a 'Contract' dropdown set to 'TokenERC20 (TokenRecipient.sol)'. The main area is titled 'TokenRecipient.sol' and contains the following Solidity code:

```
1 pragma solidity >=0.4.22 <0.6.0;
2
3 interface tokenRecipient {
4     function receiveApproval(address _from, uint256 _value, address _token, bytes calldata _extraData) external;
5 }
6
7 contract TokenERC20 {
8     // Public variables of the token
9     string public name;
10    string public symbol;
11    uint8 public decimals = 18;
12    // 18 decimals is the strongly suggested default, avoid changing it
13    uint256 public totalSupply;
14
15    // This creates an array with all balances
16    mapping (address => uint256) public balanceOf;
17    mapping (address => mapping (address => uint256)) public allowance;
18
19    // This generates a public event on the blockchain that will notify clients
20    event Transfer(address indexed from, address indexed to, uint256 value);
21
22    // This generates a public event on the blockchain that will notify clients
23    event Approval(address indexed _owner, address indexed _spender, uint256 _value);
24
25    // This notifies clients about the amount burnt
26    event Burn(address indexed from, uint256 value);
27
28    /**
29     * Constructor function
30     * Initializes contract with initial supply tokens to the creator of the contract
31     */
32    constructor(
33        uint256 initialSupply,
34        string memory tokenName,
35        string memory tokenSymbol
36    ) public {
37    }
```

- ① Open the Remix IDE at <http://remix.ethereum.org>
- ② Close the ballot file
- ③ Create a new file named `TokenRecipient.sol`
- ④ Copy the code from <https://raw.githubusercontent.com/ethereum/ethereum-org/master/solidity/token-erc20.sol>
- ⑤ Your compiler can be the default one (5.11)
- ⑥ Click the "Compile TokenRecipient.sol" button



# Deploy your smart contract

The screenshot shows the Remix IDE interface. On the left, there's a sidebar with icons for Ethereum, file operations, deployment, and a search bar with a '4' notification. The main area has tabs for 'Home' and 'TokenRecipient.sol'. The 'Home' tab displays the following Solidity code:

```
pragma solidity >=0.4.22 <0.6.0;
interface tokenRecipient {
    function receiveApproval(address
}
contract TokenERC20 {
    // Public variables of the token
    string public name;
    string public symbol;
    uint8 public decimals = 18;
    // 18 decimals is the strongly suggested default
    uint256 public totalSupply;
    // This creates an array with all
    mapping (address => uint256) publ
    mapping (address => mapping (addr
    // This generates a public event
    event Transfer(address indexed fr
    // This generates a public event
    event Approval(address indexed _c
    // This generates a public event
    event Burn(address indexed burner, uint256
}
```

The 'TokenRecipient.sol' tab shows the interface definition. Below the tabs, there are input fields for 'Environment' (set to 'JavaScript VM'), 'Account' (set to '0xCA3...a733c (99)'), 'Gas limit' (set to '3000000'), and 'Value' (set to '0 wei'). A dropdown menu shows 'TokenERC20 - browser/TokenReci'. At the bottom is an orange 'Deploy' button with the parameters '10, "Marc's Token", "ABC"'.

- ① Go to the "Deploy and run transaction screen (Ethereum-like logo on the left bar)
- ② Fill in the parameters next to the orange "Deploy" button
- ③ Click the "Deploy" button



# Deployed Contract

The screenshot shows the Ethereum Remix IDE interface. At the top, there's a dropdown menu set to "TokenERC20 - browser/TokenReci". Below it is a "Deploy" button with the value "10. "Marc's Token","ABC" and a dropdown menu set to "or". There are two buttons: "At Address" and "Load contract from Address". Underneath, a section titled "Transactions recorded:" shows one transaction. The main area is titled "Deployed Contracts" and lists the deployed contract "TokenERC20 at 0x692...77b3A (memory)". Below this, a list of functions is shown with their parameters:

- approve: address\_spender,uint256\_val
- approveAndCall: address\_spender,uint256\_val
- burn: uint256\_value
- burnFrom: address\_from,uint256\_value
- transfer: address\_to,uint256\_value
- transferFrom: address\_from,address\_to,uint256\_value
- allowance: address,address
- balanceOf: address

- ① Expand the "Deployed Contracts"
- ② All the contract functions are available from Remix



# Table of Contents

1 What is Blockchain?

2 Hands-on Introduction to Crypto

3 Create your own token

- The easy way
- The programmatic way

4 Introduction to Ethereum



# Ethereum

Ethereum is a **decentralized platform that runs smart contracts**: applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third party interference.

— <https://ethereum.org>



# A short history of Ethereum

## Key Milestones:

- (late 2013) Vitalik Buterin describes Ethereum in a paper



# A short history of Ethereum

## Key Milestones:

- (late 2013) Vitalik Buterin describes Ethereum in a paper
- (Summer 2014) Ethereum raises more than \$14 million in pre-sale



# A short history of Ethereum

## Key Milestones:

- (late 2013) Vitalik Buterin describes Ethereum in a paper
- (Summer 2014) Ethereum raises more than \$14 million in pre-sale
- (July 30, 2015) Launch of Frontier, initial (beta) version of Ethereum



# A short history of Ethereum

Key Milestones:

- (late 2013) Vitalik Buterin describes Ethereum in a paper
- (Summer 2014) Ethereum raises more than \$14 million in pre-sale
- (July 30, 2015) Launch of Frontier, initial (beta) version of Ethereum
- (March 14, 2016) Launch of Homestead, first production release



# A short history of Ethereum

Key Milestones:

- (late 2013) Vitalik Buterin describes Ethereum in a paper
- (Summer 2014) Ethereum raises more than \$14 million in pre-sale
- (July 30, 2015) Launch of Frontier, initial (beta) version of Ethereum
- (March 14, 2016) Launch of Homestead, first production release
- (Spring 2016) The DAO



# A short history of Ethereum

## Key Milestones:

- (late 2013) Vitalik Buterin describes Ethereum in a paper
- (Summer 2014) Ethereum raises more than \$14 million in pre-sale
- (July 30, 2015) Launch of Frontier, initial (beta) version of Ethereum
- (March 14, 2016) Launch of Homestead, first production release
- (Spring 2016) The DAO
- (July 2, 2016) ETH – ETC split



# A short history of Ethereum

## Key Milestones:

- (late 2013) Vitalik Buterin describes Ethereum in a paper
- (Summer 2014) Ethereum raises more than \$14 million in pre-sale
- (July 30, 2015) Launch of Frontier, initial (beta) version of Ethereum
- (March 14, 2016) Launch of Homestead, first production release
- (Spring 2016) The DAO
- (July 2, 2016) ETH – ETC split
- (October 16, 2017) Launch of Metropolis (vByzantium) –version 3



# A short history of Ethereum

Key Milestones:

- (late 2013) Vitalik Buterin describes Ethereum in a paper
- (Summer 2014) Ethereum raises more than \$14 million in pre-sale
- (July 30, 2015) Launch of Frontier, initial (beta) version of Ethereum
- (March 14, 2016) Launch of Homestead, first production release
- (Spring 2016) The DAO
- (July 2, 2016) ETH – ETC split
- (October 16, 2017) Launch of Metropolis (vByzantium) –version 3
- (2017) ETH goes from \$7 to more than \$700 (100x increase)

*Check the nice infographic (Invezz, 2017).*

Also, see the official *Ethereum White Paper*.



# Store of value

## Ethereum (ETH) Price

Closing Price  OHLC



Figure: ETH price (Coindesk, 2017)



# Decentralization

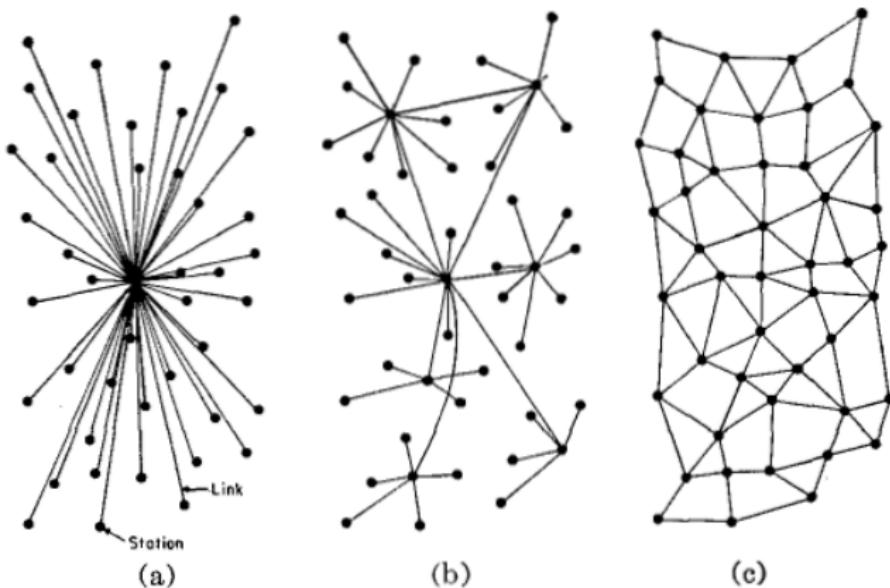


Fig. 1—(a) Centralized. (b) Decentralized. (c) Distributed networks.



# Client Types

- Full node



# Client Types

- Full node
- Light node



# Client Types

- Full node
- Light node
- Something in between (e.g. “fast” for geth)



# Disk Space

## Full Archive Ethereum node

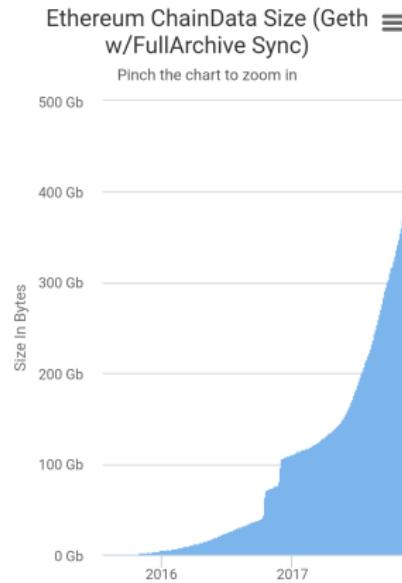
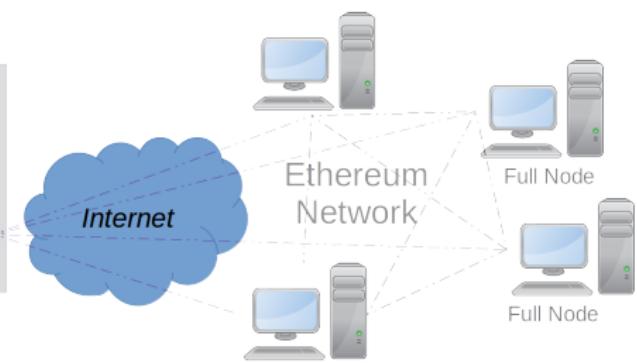
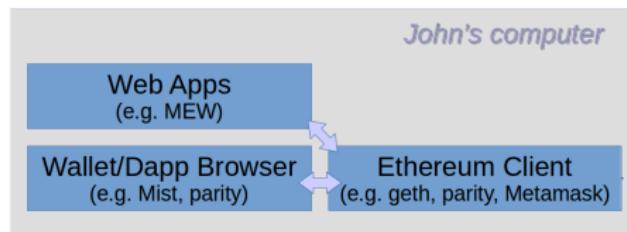


Figure: Miners need a lot of space (Reddit, 2017)



# Practical Applications

for personal or business use



## Reference books

# Blockchain in practice

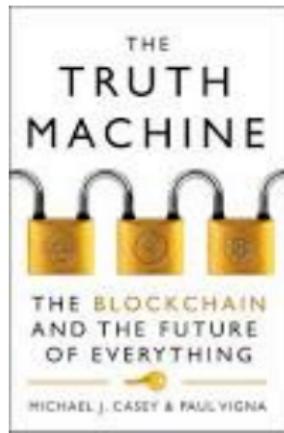
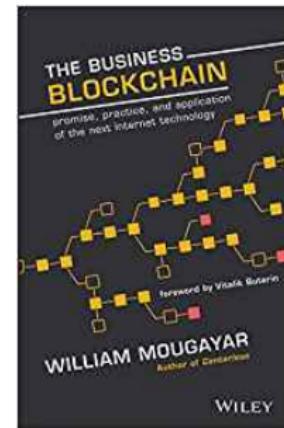


Figure: Book from Vigna and Casey,  
2018



## Figure: Book from Mougayar, 2016



# Ultimate references

The must-read white paper from legendary Satoshi Nakamoto: *Bitcoin: A Peer-to-Peer Electronic Cash System* ([2008](#)),

the Ethereum white paper from home-town Toronto Vitalik Buterin: *A Next-Generation Smart Contract and Decentralized Application Platform* ([2013](#)),

and the yellow paper authored by Prof. Gavin Wood: *Ethereum: A secure decentralised generalised transaction ledger* ([2014](#)).



# Thank you!

Email: [marc@creative-emergy.com](mailto:marc@creative-emergy.com)

Twitter: [@marclijour](https://twitter.com/marclijour)

[www.metameshgroup.com](http://www.metameshgroup.com)



# References

- Buterin, V. (2013). A next-generation smart contract and decentralized application platform. Retrieved from <https://github.com/ethereum/wiki/wiki/White-Paper>
- Coindesk. (2017). Ethereum (ETH) Price. Retrieved from <https://www.coindesk.com/ethereum-price/>
- Invezz. (2017). Infographic: The story of Ethereum. Retrieved from <https://cdn4.benzinga.com/files/images/2017/July/05/invezz-eth-history-base.jpg>
- Mougaray, W. (2016). *The business blockchain: Promise, practice, and application of the next internet technology* (1st ed.). Wiley.
- Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Retrieved from <https://bitcoin.org/bitcoin.pdf>
- Reddit. (2017). Ethereum blockchain size...we have a problem. Retrieved from [https://www.reddit.com/r/ethtrader/comments/7axn5g/ethereum\\_blockchain\\_sizewe\\_have\\_a\\_problem/](https://www.reddit.com/r/ethtrader/comments/7axn5g/ethereum_blockchain_sizewe_have_a_problem/)
- Vigna, P., & Casey, M. J. (2018). *The Truth Machine: The Blockchain and the future of everything* (1st ed.). St. Martin's Press.
- Wood, G. (2014, April). Ethereum: A secure decentralised generalised transaction ledger. Retrieved from <https://ethereum.github.io/yellowpaper/paper.pdf>

