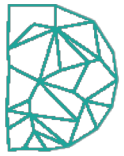


DECENTRALIZED
TRAINING SERIES

Day 1 - Session 2

An Introduction to Ethereum

Klitos Christodoulou, Ph.D.

christodoulou.kl@unic.ac.cy

UNIC | Institute For the Future

CERTH CENTRE FOR RESEARCH & TECHNOLOGY HELLAS | iti Information Technologies Institute

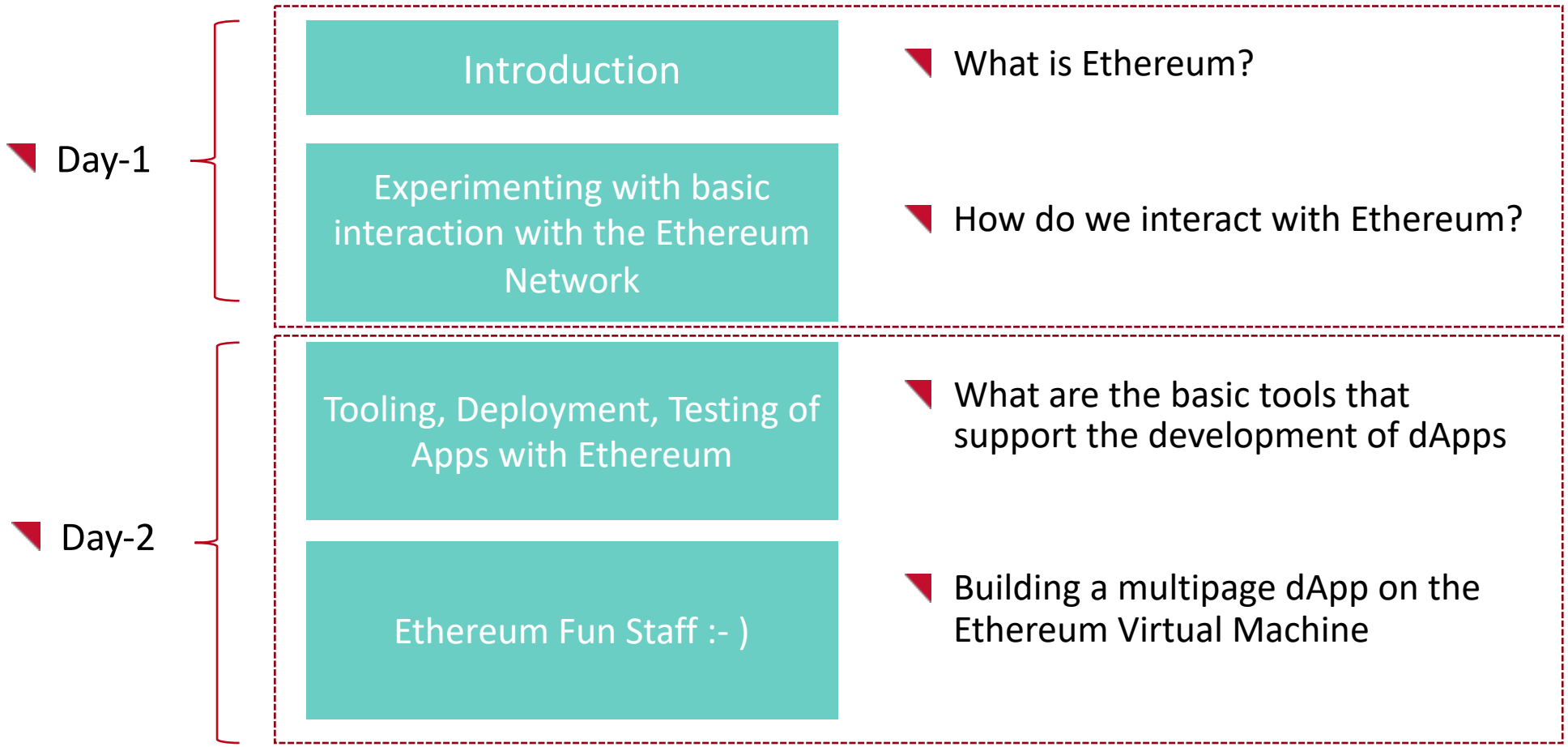
Agenda

- Blockchain Fundamentals
- What is a Blockchain
- Tokenization
- An intro to Bitcoin Script
- The Bitcoin Network
- Alternative Cryptocurrencies
- An intro to Ethereum
- Bitcoin vs Ethereum
- The Ethereum Roadmap
- The World Computer
- From PoW to PoS
- Scaling in Ethereum
- The eco-system of Ethereum
- Scalability Issues
- Potential Directions for dApps

Overview of DTS on Ethereum Developer

Introduction	What is Ethereum?
Experimenting with basic interaction with the Ethereum Network	How do we interact with Ethereum?
Tooling, Deployment, Testing of Apps with Ethereum	What are the basic tools that support the development of dApps
Ethereum Fun Staff :-)	Building a multipage dApp on the Ethereum Virtual Machine

Overview of DTS on Ethereum Developer



What we will not learn in this Workshop

- In this Workshop we will **not** learn how to trade crypto-currencies!

UNIC Institute For the Future

CERTH CENTRE FOR RESEARCH & TECHNOLOGY HELLAS

iti Information Technologies Institute

What we will not learn in this Workshop

- We will not get into the very deep academic discussion of crypto-science, Distributed Ledger Technologies, Blockchains, Byzantine Fault tolerance, zero-proofs, cryptography etc. etc..

What is the aim of this Workshop?

- “The goal of this 2 - Day workshop is to teach you how to build Smart Contracts on the Ethereum Virtual Machine and interact with them using a Web-like environment.”

Smart Contracts + **Web interface** = dApp

Back to the Future…

A Short History Lesson

What is the aim of this Workshop?

- Development on the Blockchain space is changing day by day…

Bitcoin: A p2p Electronic Cash System

Oct 31, 2008

- Bitcoin paper was released…
- The idea of Blockchains is described
- At this original paper Bitcoin is proposed as a storing ledger of transactions…
- Financial transactions are enabled without the need of a middleman!

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

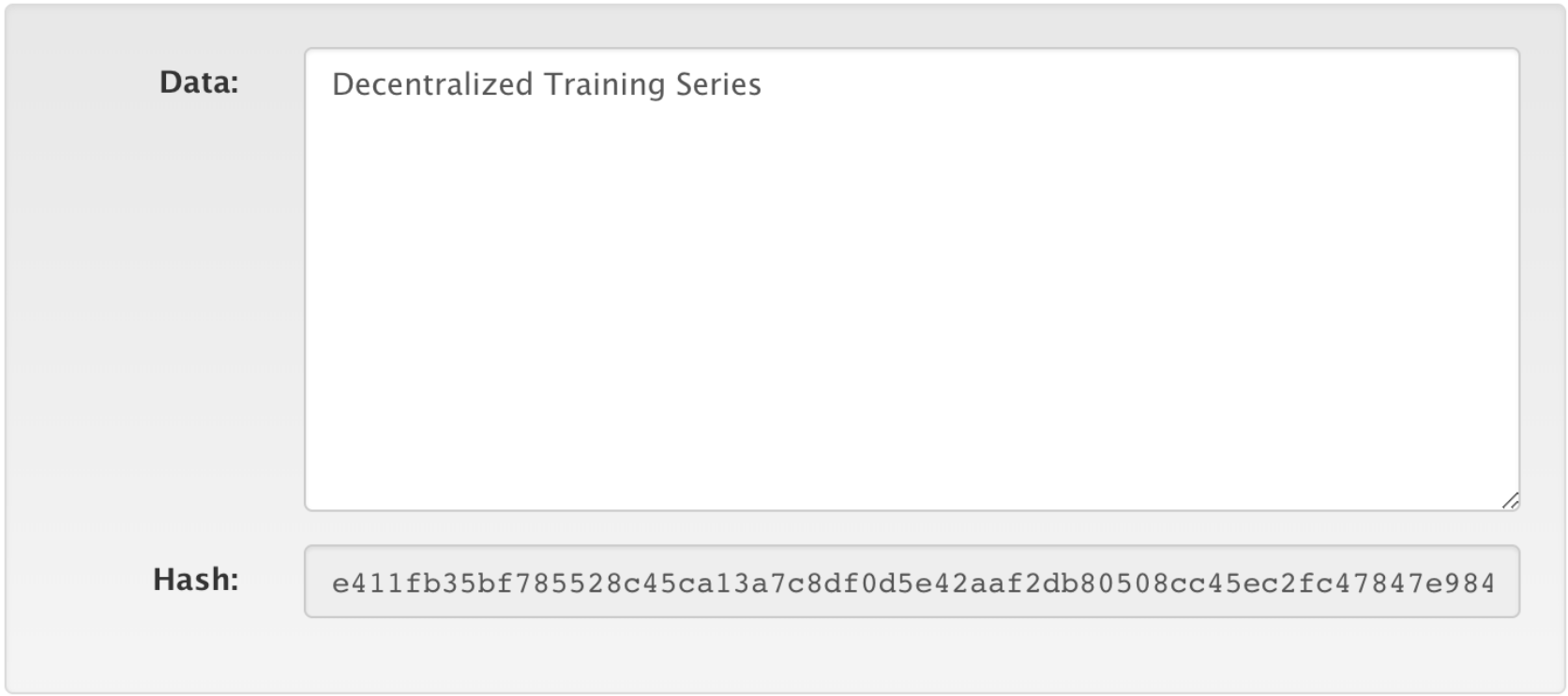
Blockchain Fundamentals: It is Block Time

- What is a hash function (e.g., SHA 256) ?

A piece of Digital Information → Hash Function → A finger-print of the Digital Data used as input

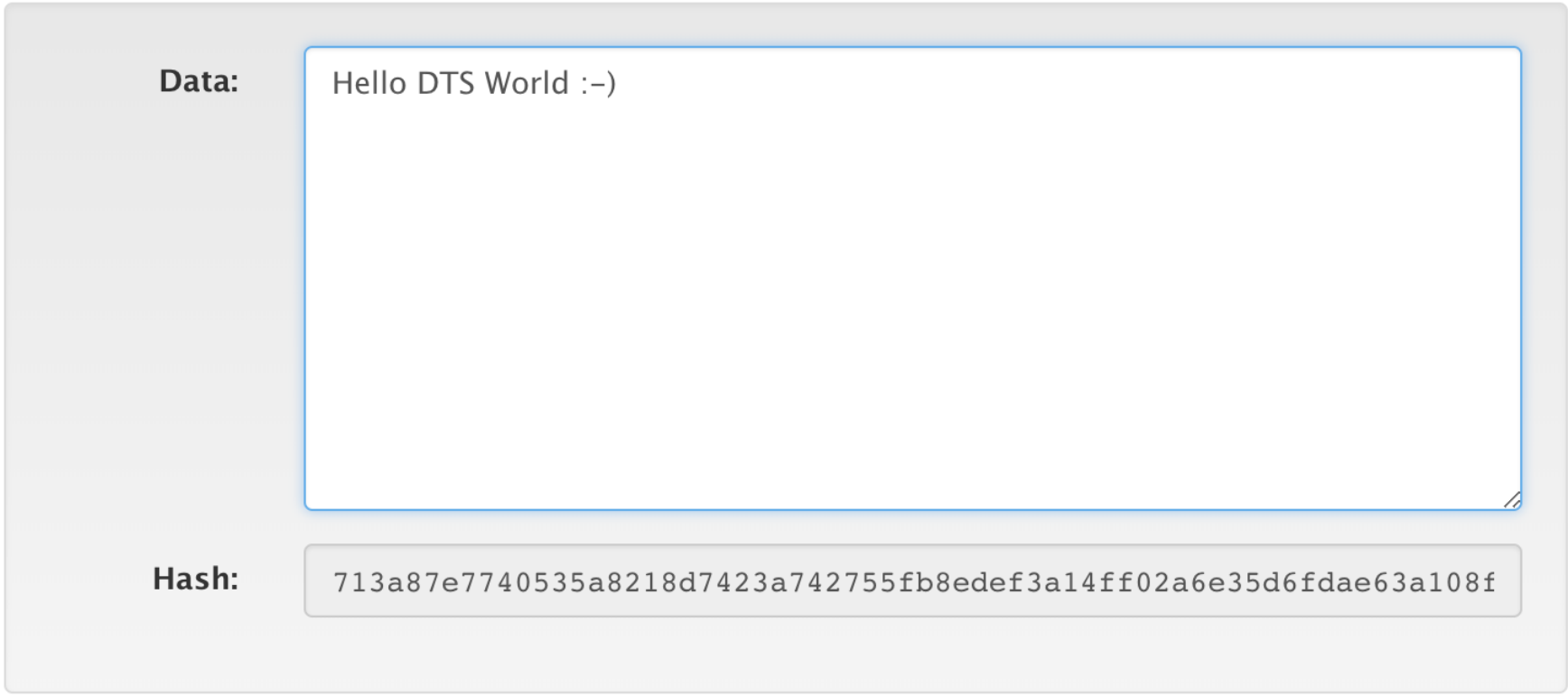
Blockchain Fundamentals: It is Block Time

- Example using the SHA 256



Blockchain Fundamentals: It is Block Time

- Example using the SHA 256



Blockchain Fundamentals: It is Block Time

- … a Hash is a mechanism of creating a digital signature of some piece of information of a certain length (e.g., 256 bits) …
  - For the same piece of information we get the exact same Hash
  - All Hashes have the same length

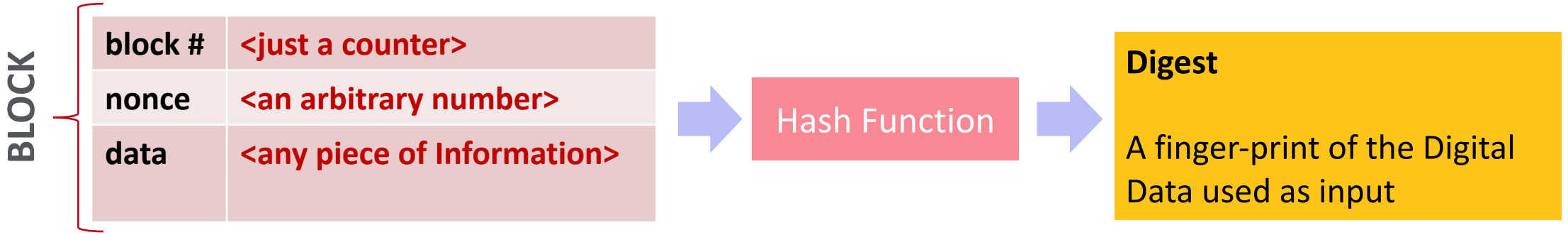
Decenralized Training Series → `e411fb35bf785528c45ca13a7c8df0d5e42aaf2db80508cc45ec2fc47847e984`

Hello DTS World :-) → `713a87e7740535a8218d7423a742755fb8edef3a14ff02a6e35d6fdae63a108f`

- Example of Hashes using SHA-256

Blockchain Fundamentals: It is Block Time

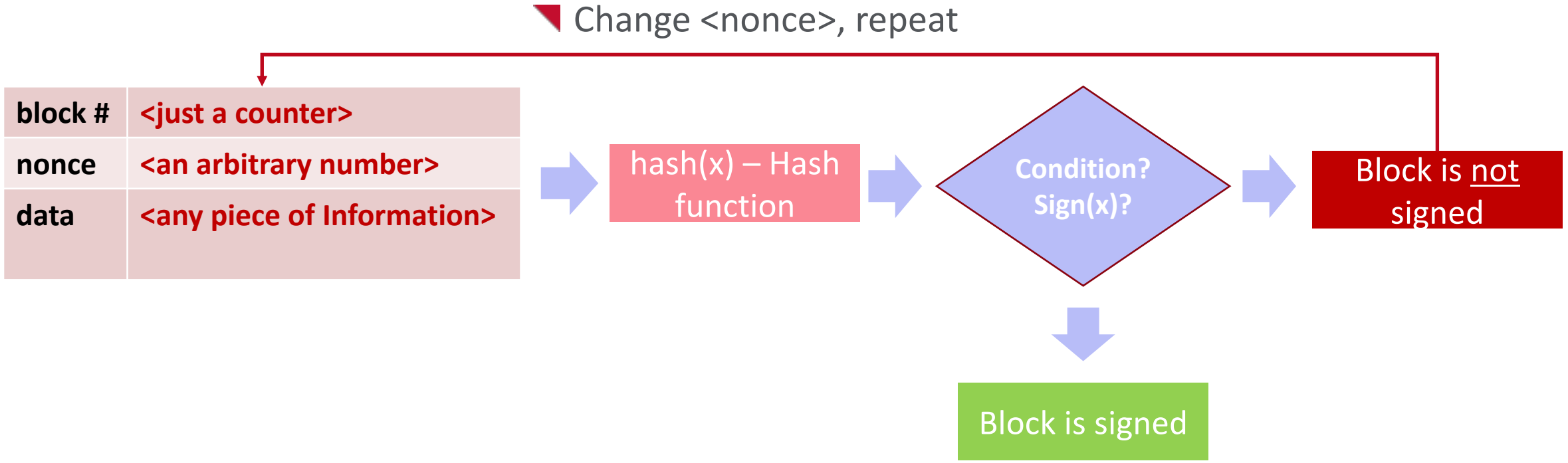
- Let us evolve this idea of a Hash to the idea of a ”BLOCK”
  - Instead of having just a single input of data we have broken down **data** to the following: <block #, nonce, data>



- Additional Requirement
  - The Digest (output Hash) needs to satisfy a certain condition

Blockchain Fundamentals: Signed Block?

- A block is **signed** when the output hash satisfies a certain condition



Blockchain Fundamentals: Signed Block?

- Let us create our own signed condition

Definition – sign()

We consider that any given BLOCK is signed when the digest (output of the hash function) begins with 4 leading zeros

Example: `0000f727854b50bb95c054b39c1fe5c92e5ebcfa4bcb5dc279f56aa96a365e5a`

- Arbitrarily, we consider that a Block is signed if the nonce, for a given piece of data, returns a hash that satisfies the above condition.

Blockchain Fundamentals: It is Block Time

- Example of a Block

BLOCK

block #	**1**
nonce	**72608**
data	**Hello :-)**

Block is not signed

Hash: `f23b5f6168e9c8fecd5aab55f34c992e51c7033cc50b9021e1042f9c7dde25be`

- Nonce **does not** satisfy the condition given the data of the Block.
- Condition: we need to find a Hash that begins with 4 leading zeros.

Blockchain Fundamentals: Block Validation

- Let us define our own *proof-of-work* function

Definition – *proof-of-work*

Keep changing the value of *nonce* until we hit a Hash that satisfies the Condition.

- Here you have it! the first simple idea of what we refer as a *mining process.*
- *This could be a very compute-intensive task!*

Blockchain Fundamentals: Block Validation

- Example of a Block

BLOCK

block #	**1**
nonce	**32904**
data	**Hello :-)**

Block is signed

Hash: `00002462488067cf69de151b63b06aae9324f26023aa49b8d8ea3cfb2ec6e0b5`

- Nonce **does** satisfy the condition given the data of the Block.
- **So if this a Block can you tell me what a Blockchain is??**

Blockchain Fundamentals: What is Block-time

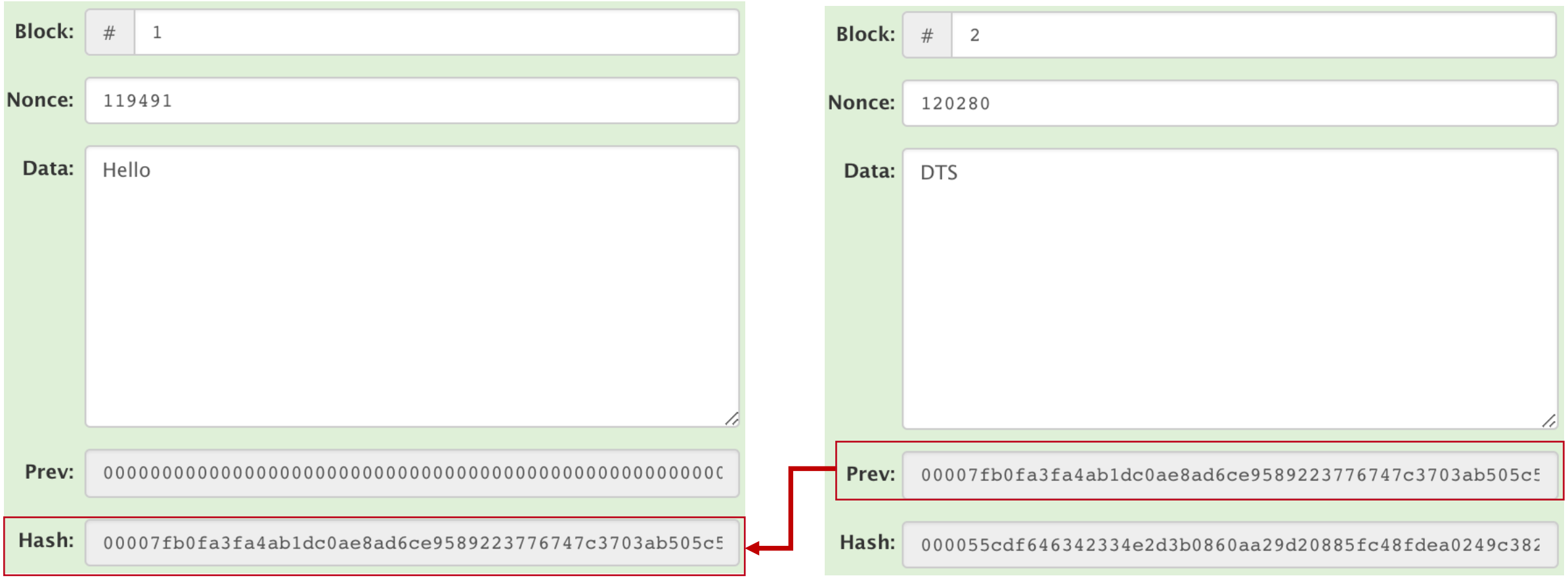
- We refer to the amount of **time** that It takes to Hash everything from 0 *nonce* to the target nonce as a **Block-time**.
- That is the time needed to run all these random Hashes until we find a *nonce* values that satisfies a specific condition according to the sign() function.
- Let us revisit the above once we introduce the distributed dimension to it!

What is a Blockchain?

Fundamentals

Blockchain Fundamentals: What is it?

- Is a series of valid (signed Blocks) that are put together and they store data!

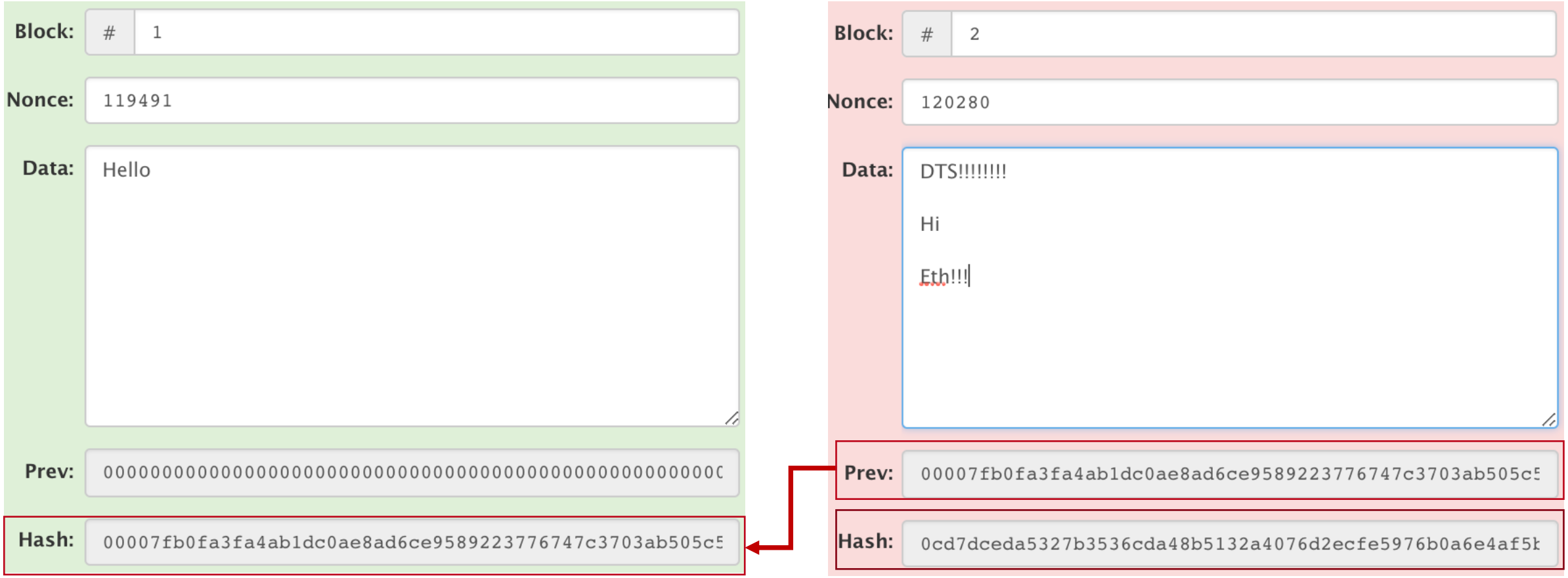


Blockchain Fundamentals: What is it?

- **Fact:** Each Block points back to the one before it.
- **Fact:** If we change the data (information) form a valid Block with a certain *nonce* then the Hash changes and thus the Block gets invalid.

Blockchain Fundamentals: What is it?

- Is a series of valid (signed Blocks) that are put together and they store data!



- **Does not satisfy the condition**

Blockchain Fundamentals: What is it?

- **Fact:** Each Block points back to the one before it.
- **Fact:** If we change the data (information) form a valid Block with a certain *nonce* then the Hash changes and thus the Block gets invalid.
- ***Thus,*** the next Block get invalid because it now has a different Hash, since the Previous Block Pointer changes!
- ***Solution:*** We need to **mine** that Block again!

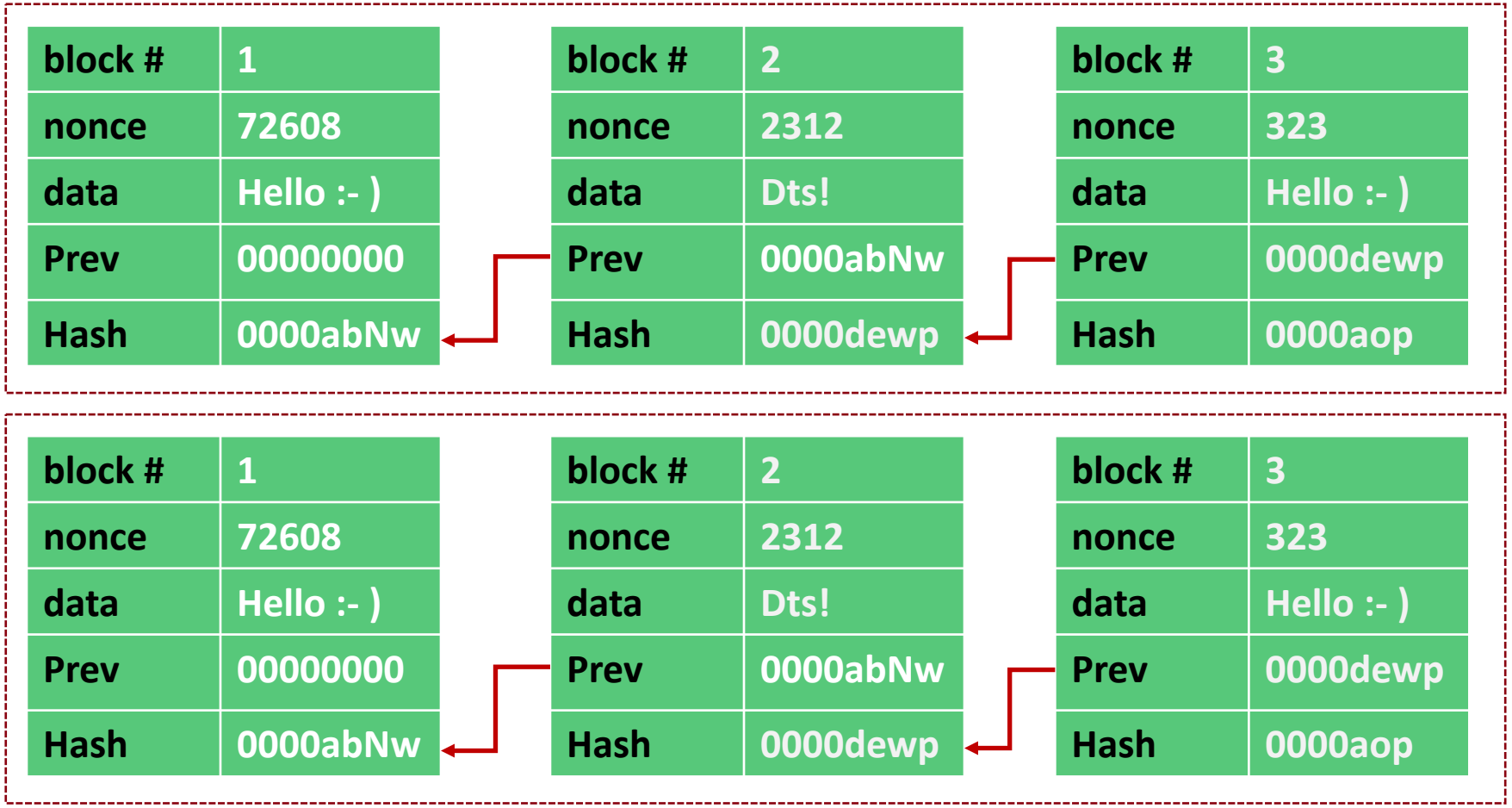
- **The more blocks we alter in the past the more blocks we have to validate!**
- **Imagine how this will scale if we introduce more than one peers in the game!**

Blockchain Fundamentals: What is it?

- **Fact:** Even if we go back in time and alter any Block data from a Blockchain, all the Blocks after that will need to be signed again and thus we need to spend a significant computational power to mine all the Blocks and thus validate the chain.
- Because for this property Blockchains resist to change/alternation.
- **This idea is reinforced when we create a distributed Blockchain copied across various distributed peers.**

Blockchain Fundamentals: Distributed Blockchain

- Each peer has an exact same copy of the Blockchain…



Blockchain Fundamentals: Distributed Blockchain

- **Fact:** If a peer changes any information from its blocks, even if all the Blocks in its own Blockchain are valid, a majority voting algorithm determines if that peer is trustful.
- **Fact:** By looking only the Hash of **the most recent** Block from all peers one can determine if we have a **fraudulent peer!**

- **What is a solution?**
  - Once a peer discovers a nonce that satisfies the condition then this solution is distributed to the other peers for confirmation!

- In the Ethereum/Bitcoin Blockchain we have also the meaning of a **Difficulty**, we will revisit this once we introduce Ethereum…

Tokenization

A Digital Currency is born

Blockchain First Use-case: A digital token

- **Fact:** The idea of a Blockchain is that we can cryptographically store some data, is this actually useful?
- The added value of this simple idea emerges from how we are using this technology, in other words on how we **instantiate** the Blockchain data field with useful information :-)
- **What about recording data for Financial Transactions!**

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Blockchain First Use-case: A digital token

Block:	#	1				
Nonce:	139358					
Tx:	$	25.00	From:	Darcy	->	Bingley
	$	4.27	From:	Elizabeth	->	Jane
	$	19.22	From:	Wickham	->	Lydia
	$	106.44	From:	Lady Cathe	->	Collins
	$	6.42	From:	Charlotte	->	Elizabeth
Prev:	0000000000000000000000000000000000000000000000000000000000000000					
Hash:	00000c52990ee86de55ec4b9b32beefd745d71675dc0eddfbc7b8					

- In practice, in each transaction we are not remembering balances but only money movements
- Here we can ponder on the meaning of the pointers
  - We can backtrack and trace that one account has enough money to send to other people.
  - An efficient way to reach agreement!
- We are resisting any kind of modification by keeping track of the same blocks on various peers

Bitcoin Script: Introduction

- Bitcoin Script: the underlying mechanism of Bitcoin transactions
  - A (simple) programming language
  - "script": an executable program - a list of instructions
  - A transaction is validated if the respective script is successfully executed
- The constituents of each script can be distinguished into two basic types
  - **Opcodes** (commands/functions): begin with prefix 'OP_', e.g., 'OP_HASH160'
  - **Data**: store data related to transactions - appear within '<.>', e.g., '<pubKeyHash>'

Bitcoin Script: Introduction

Script is a Forth-like **reverse polish notation, stack-based** execution language, which is **not Turing-complete** and does not include loops.

- Forth is an imperative stack-based computer programming language and environment, originally designed by Charles "Chuck" Moore
- Capable of running in a few kilobytes of memory.

But …

- What is a **Turing-complete** language?
- What is a **stack-based** language?
- What is **reverse polish notation**?

(Non)-Turing Complete

In computability theory, a system of data-manipulation rules (such as a computer's instruction set, a programming language, or a cellular automaton) is said to be Turing-complete if it can be used to simulate any Turing machine.

- In Script we do not have the halting problem

  Halting problem: given a program and an input, determine whether the execution of the program will be terminated

  Example 1

```
while (true) {
   a = a + 1
}
```

  Example 2

```
for (i=1; i<=10; i++) {
    a = a + 1
}
```

- **Every Bitcoin Script will terminate in finite steps**.

Reverse Polish Notation

In Reverse Polish notation, the operators follow their operands.

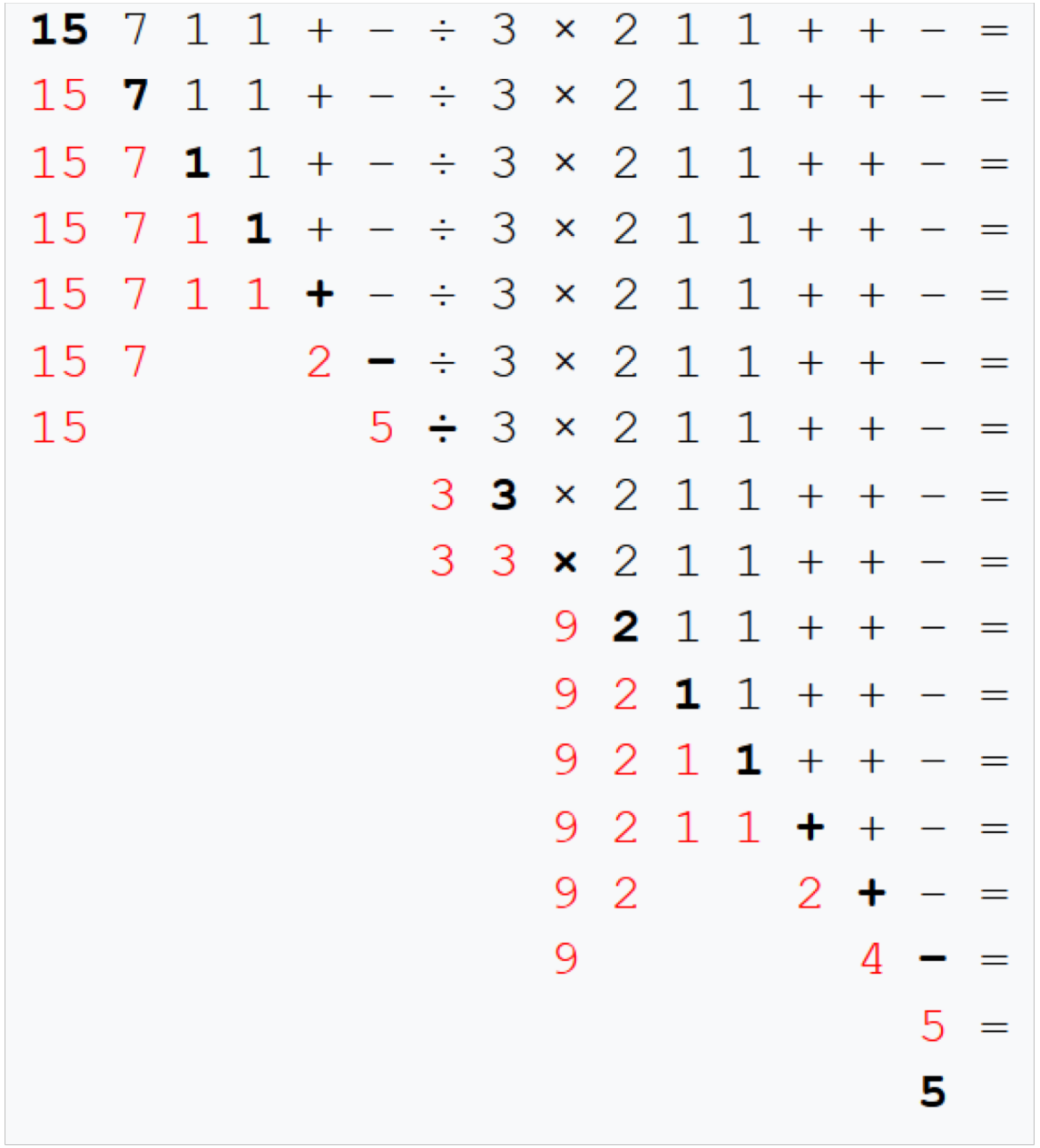
In RPN notation should be ordered like this:

<FirstNumber> <SecondNumber> <Operation>

rather than the normal convention(infix) of:

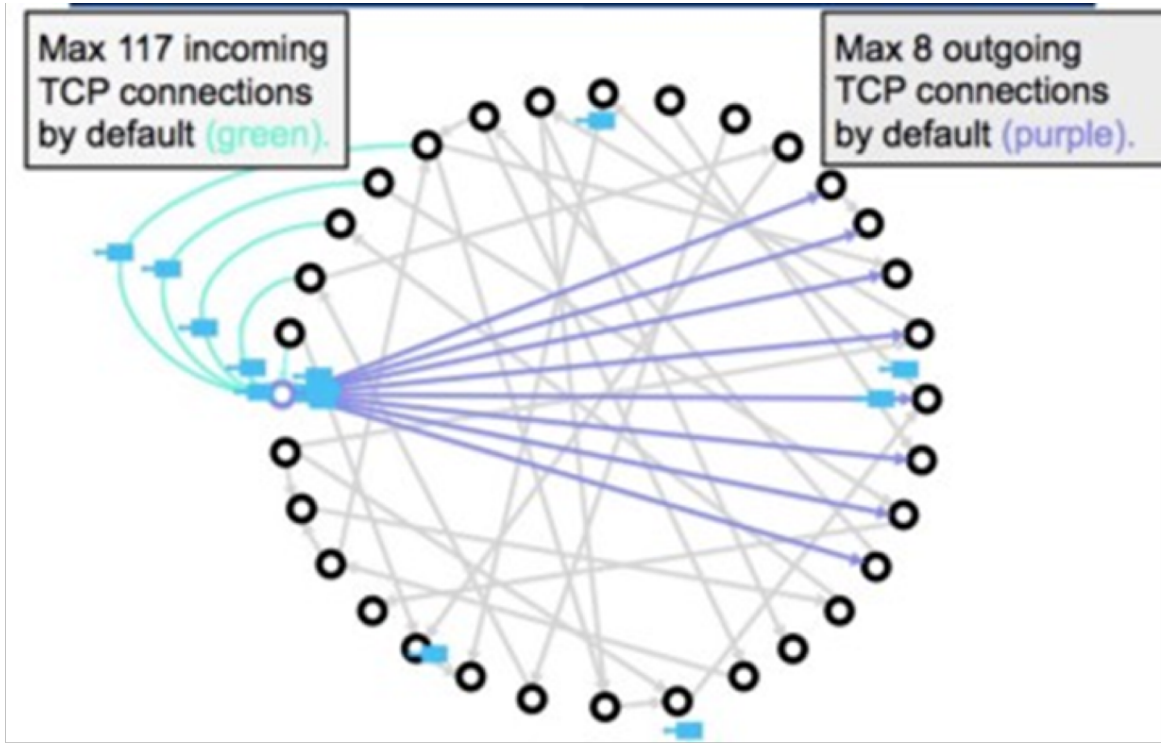
<FirstNumber> <Operation> <SecondNumber>

- For example, to add 3 and 4, the standard way is to (3+4) rather than (RPN) 3 4 +.
- The infix expression ((15 ÷ (7 − (1 + 1))) × 3) − (2 + (1 + 1)) can be written like this in reverse Polish notation: 15 7 1 1 + − ÷ 3 × 2 1 1 + + −



Bitcoin Network: Abstract View

- The Bitcoin network is a peer-to-peer network of distributed nodes.
- By design, each node can have **up to 117 incoming** connections and **up to 8 outgoing connections**.
- When joining the network, nodes connect to each other, thus forming a **gossip network**, where information propagates across the whole network and peers are able to broadcast transactions and blocks.



- There are two major ways of controlling the network:
  1. **Controlling the information flow** between peers.
  2. **Controlling the computational power** of the network – remember that decisions are based on consensus/majority.

Bitcoin Network: 51% Attack

- An adversary that controls **more than half** of the network's computing power can effectively control the entire network.
  - When nodes are in doubt (i.e. they receive conflicting information), they trust the majority.
  - Hence, an adversary that controls the majority of network resources can propagate information they want to the network.
- While controlling the network, **the attacker can**:
  - Reverse transactions that s/he sends, thus **double-spending** own funds.
  - **Prevent other miners** from mining valid blocks.
  - **Prevent valid transactions** from gaining confirmations.
- However, **the attacker cannot**:
  - Reverse other people's transactions or spend outputs belonging to others.
  - Create new coins.
  - Prevent transactions from being sent across the network.

Bitcoin Network: Flood attack

- **A flood attack is the process of sending thousands of nano-value transactions, in order to fill the blocks to the maximum size**.
  - This will create delays to other legitimate transactions, thus delaying the whole network and increasing confirmation time for all transactions.

- **A flood attack is performed very easily**, with the attacker just sending thousands of transactions to himself.

- However, **it is expensive to sustain for a long time**, due to transaction fees.

- In July 2015, a flood attack hit the Bitcoin network, leaving more than 80,000 uncofirmed transactions in the mempool.

Bitcoin Network: Selfish Mining

- Selfish mining is an attack on the integrity of the Bitcoin network, which **can be used by large miners to increase their returns** by not playing fair.

- Here is how it works:
  - **The selfish miner starts building a chain of blocks, but does not publish and distribute it to the rest of the network**. Obviously, the selfish miner needs to have large mining power and a bit of luck to do this.
  - When the rest of the network is about to catch up with the selfish miner, the miner releases a portion of the chain to the public.
  - **Because the chain of the selfish miner will be longer and more difficult, the rest of the network will discard the blocks of other miners and will adopt the chain of the selfish miner**.
  - This strategy is repeated to ensure that the private chain built by the selfish miner will always be better (longer and more difficult) that its competitors

Bitcoin Network: Selfish Mining - Consequences

- **The computing power of honest miners is wasted**, as they repeatedly find themselves working on the wrong chain.
- As a result, **selfish miners increase the impact of their own mining power on the network** and enjoy additional power and profits.
- **Selfish mining increases transaction confirmation times**, because transactions confirmed by the selfish miner in private, are not broadcast to the public immediately.
- **Selfish mining also increases the threat of double spending**, as both honest and selfish miners can add mutually exclusive transactions to the private and public chains.

- It is estimated that a malicious miner needs to control only one third of the network's mining power to launch and sustain this type of attack (**33% attack**).

Bitcoin Network: Denial of Service (DoS) Attack

- A **denial of service (DoS)** or **distributed denial of service (DDoS)** attack is an attempt to make an online service unavailable by overwhelming it with traffic.
  - In a typical DoS attack, the attacker will overload a network/computer with requests above the capacity that the network/computer can handle.
  - In Bitcoin, this can be achieved by **sending lots of junk data to a node**. The nodes under attack will not be able to process normal Bitcoin transactions/blocks or receive new ones.
  - The block size limit is a first **countermeasure** against DoS attacks to Bitcoin nodes.

- Bitcoin has **built-in prevention mechanisms** against basic DoS attacks:
  - At the **protocol** level
  - At the **node (peer)** level
  - However, the network is still vulnerable to newer, more sophisticated, types of DoS.

Bitcoin Network: Protocol-based anti-DoS rules

- Various limitations have been embedded on the Bitcoin protocol to prevent DoS attacks:
  - The **maximum block size** (currently 1Mb)
  - The **maximum number of signature checks** that a transaction or block may request
  - The **maximum script size** (currently 10Kb)
  - The **maximum size of values pushed while executing a script** (currently 520 bytes)
  - The **maximum number of "expensive" operations** in a script (up to 201 operations – anything but push operations).
  - The **maximum number of keys in multi-sig transactions** (currently 20 keys)
  - The **maximum number of stack elements stored** (currently 1,000 elements)

Bitcoin Network: Node-based anti-DoS rules

- To prevent DoS attacks, a bitcoin node/peer:
  - does not store more than 10,000 orphan transactions;
  - does not forward orphan transactions/blocks;
  - does not forward double-spend transactions;
  - does not forward the same block or transaction to the same peer;
  - does not forward or process non-standard transactions;
  - **bans IP addresses** that misbehave;
  - keeps a **DoS score** for each peer;
  - **penalizes peers** that send duplicate/expired/invalid signature messages;
  - **disconnects from peers** that send messages that fail to comply with the rules;
  - stores only UXTO (unspent transaction output set) in memory
  - checks all inputs are unspent before fetching a transaction from disk to memory – thus preventing a type of DoS, known as **continuous hard disk activity DoS**

Bitcoin Recap

- Bitcoin is the first application of a technology that paves the way forward, revealing an opportunity for innovation that was not apparent before.
- Bitcoin is wholly open source, so every element of it can be tweaked, modified, altered and tested for potentially improved iterations, just like evolution.
- Bitcoin's blockchain has grown large (approximately 185GB by October 2018) – and will only become larger, as Bitcoin use becomes more widespread.
- The process of mining is power intensive, which may be argued is with a disproportionate benefit towards the network, unless this is mutualized to many more transactions.
- The nature of a predetermined, and eventually deflating monetary base as coins are irrecoverably lost, may also be among the dissuading factors of some using it.

Why alternative currencies?

- The freedom to try out every possible solution has driven many to spawn their own “alt – coins”, with their own rules and their own networks. Some older concepts like Ripple have been augmented by the innovation of the blockchain and have developed in their own right.
- While some are merely small modifications of the Bitcoin protocol and have limited audiences, others are interesting sources of innovation. The differences derive from changes in the basis of each coin’s philosophy which are achieved in a variety of ways, such as:
  - Altering the issuance method to less energy intensive processes
  - Adding more functions like smart contracts
  - Improving fungibility and privacy characteristics of the currency itself
  - Altering the monetary supply and issuance rate
  - Altering the hashing algorithms or other parameters
  - Introducing other concepts such as demurrage to increase the velocity of money

Ethereum: The Ultimate Smart Contract & Decentralized Application Platform

- A strong believer that Blockchain could be proven useful for more than financial transactions….

Vitalik Buterin – Ethereum

Ethereum: The Ultimate Smart Contract and Decentralized Application Platform

In the last few months, there has been a great amount of interest into the area of using Bitcoin-like blockchains, the mechanism that allows for the entire world to agree on the state of a public ownership database, for more than just money. Perhaps the first, and oldest, such alternative application is colored coins, which is a protocol that allows users to label specific bitcoins and treat them as assets representing some real world value - whether company shares, collectibles or even existing currencies like gold and USD. A more independent alternative, Ripple, also includes the ability to create custom currencies and assets, but adds a decentralized exchange. More recently, Mastercoin has started to go even further, allowing more complex financial contracts such as hedging, trust-free dice rolls, binary options and self-stabilizing currencies - essentially, almost any common financial instrument imaginable. Taken together, all of these projects can be thought of as initial efforts toward a sort of "cryptocurrency 2.0" - they are to Bitcoin what Web 2.0 was to the World Wide Web circa 1995.

Dec .2013

Ethereum: The Ultimate Smart Contract & Decentralized Application Platform

- ”Whitepaper” discusses the need for having a more programmatic control over transactions.
- Wanted to enable the creation of ‘decentralized autonomous corporations’
  - Sub crypto-currencies
  - Domain registration systems
- To enable the creation of advanced applications on the Blockchain
- Introduces the idea of **‘Smart Contracts’** as an entity for sending currency beyond just humans!

Ethereum: The Ultimate Smart Contract and Decentralized Application Platform

In the last few months, there has been a great amount of interest into the area of using Bitcoin-like blockchains, the mechanism that allows for the entire world to agree on the state of a public ownership database, for more than just money. Perhaps the first, and oldest, such alternative application is colored coins, which is a protocol that allows users to label specific bitcoins and treat them as assets representing some real world value - whether company shares, collectibles or even existing currencies like gold and USD. A more independent alternative, Ripple, also includes the ability to create custom currencies and assets, but adds a decentralized exchange. More recently, Mastercoin has started to go even further, allowing more complex financial contracts such as hedging, trust-free dice rolls, binary options and self-stabilizing currencies - essentially, almost any common financial instrument imaginable. Taken together, all of these projects can be thought of as initial efforts toward a sort of "cryptocurrency 2.0" - they are to Bitcoin what Web 2.0 was to the World Wide Web circa 1995.

Dec .2013

Bitcoin vs Ethereum

Founder	Satoshi Nakamoto	Vitalik Buterin
Release Date	9 Jan 2008	30 July 2015
Release Method	Genesis Block	Presale
Consensus Algorithm	Proof-of-work	Proof of Work (Planning Proof of Stake)
Usage	Digital Currency, limited programming	Smart Contracts – Programming Cababilities Programmable Money – Micor payments
Cryptocurrency	Bitcoin (Satoshi)	Ether
Algorithm	SHA256	EthHash
Mining	ASIC Miners	GPUs
Scalable	Not yet – various attempts	More Scalable than Bitcoin but still

Ethereum

- **Ethereum** is a a hybrid meta/alt-coin that attempts to build, in their own words, “a revolutionary new platform for applications”, targeting anything from voting to: financial exchanges, to smart property, and most importantly, **decentralized applications**.
- Even though the currency used in the network is an alt-coin (ether), it is used more as the computational fuel than a scarce currency:
  - A standardized foundation platform (i.e. the enhanced Ethereum programming abstractions, protocol and network)
  - A programming language to facilitate the creation of distributed applications by anyone
  - Its own currency or cryptofuel – the “Ether”

Source: ethereum.org

- One of the very important concepts that Ethereum attempts to achieve is a level of being **“Turing Complete”.** (Definition) So far, the explanation given by the Ethereum team is that they are attempting to make a quasi-Turing-complete system. The cost of each step of these recursive processes or loops is the fuel of the system (ether) as a fee.
- Ethereum is based on the concept of **self-executing smart contracts**, software contracts that execute specific instructions upon interacting with them through transactions.

Ethereum

- **Ethereum is a decentralized open-source platform developed to host smart contracts**. Ethereum blockchain is able to run the programming code of any decentralized application.
- Developers can build thousands of different applications, different to anything we have seen before, because of Ethereum's real innovation, the **Ethereum Virtual Machine** (EVM).
- **EVM** enables users to run any program, no matter what the programming language is, given there is enough time and memory available. In simple words, EVM is able to perform any calculation that any other programmable computer is capable of, therefore capable of designing any type of smart contract.
- Instead of having to build a new blockchain for each new application, Ethereum enables the development of many applications all on one platform.

Ethereum – in detail

- Ethereum approaches the existing Bitcoin infrastructure as a “state machine”, where transactions (which store messages) serve as “state transitions” between Ethereum accounts without the Unspent Transaction Output (UTXO) basis we have in Bitcoin. There are two types of Ethereum accounts:
  - **Externally-owned accounts** – used for sending messages, and do not contain code
  - **Contract accounts** – used for executing a specific contract code upon receiving a message
- An ***Ethereum account*** consists of:
  - **An Ether balance** – used for paying transaction fees
  - **A contract code** – used by contract accounts to implement application logic
  - **Storage** – used by contract accounts for retrieving or storing information accordingly as their code executes, otherwise it is empty
  - **A nonce** – used for ensuring that transactions are only processed once
- ***Ethereum messages*** serve as “functions” and have the following characteristics:
  - They can be created by an external entity or a contract
  - They can contain data
  - They can only receive responses from contract accounts

Ethereum

- **Ether**, is the token mined which fuels the network. It used by application developers to pay for transaction fees and services on the Ethereum network.
- Does a vending machine has any uncertainty whether it will deliver your chocolate? An Ethereum application is programmed without any possibility of fraud and downtime.
- When a transaction is sent with a message addressed to a specific contract, depending on the code embedded into that contract (think about the code as the terms of a legal contract), the contract may **execute transactions, modify its storage, trigger other contracts, etc.**
- List of Dapps on Ethereum with smart contract functionality: https://www.stateofthedapps.com/

```
1  pragma solidity ^0.4.25;
2
3  contract Twitter {
4
5      //Variables declaration
6      string public tweetMessage;
7
8      //constructor
9      constructor(string _newTweet) public {
10         tweetMessage = _newTweet;
11     }
12
13     //Methods
14     function setMessage(string _m) public {
15         tweetMessage = _m;
16     }
17
18     function getMessage() public view returns(string) {
19         return tweetMessage;
20     }
21 }//end contract
```

Example of a smart contract on Ethereum

Ethereum – in detail

- **Transactions** in Ethereum are viewed as “signed data packages” and contain:
  - **A message** to be sent from an externally-owned account
  - **A Sender signature** – which indicates the sender of the message
  - **A Receiver address** – which indicates the receiver of the message
  - **An Ether amount** – which indicates the amount of Ether to send
  - **Data** – which encapsulates the data to be sent
  - **A Start Gas field** – which limits the number of computational steps over which a contract code will execute
  - **A Gas Price field** – which is the fee that will be paid to a miner at each computational step
- To achieve its goals, Ethereum defines its own logic for state transitions processing and code execution, whose details are beyond the scope of this Session.

Ethereum Benefits & Challenges

Benefits:

- **Security** – Being decentralized and using advanced cryptography, apps are protected against hackers
- **Immutability** – A third party cannot make any changes to data. Because of the consensus mechanism, corruption and attempts to change the state of the apps is impossible.
- **Longevity – Zero downtime** – An app never goes down even if someone loses interest in maintaining it

Challenges: (see also next slide)

- Do not forget that code is written by humans. Therefore smart contracts are as good and reliable as their creators
- Smart Contracts are still difficult to understand and be implemented by a non-programmer. Perhaps we should expect that creating a smart contract may become simpler and possible for an average person in the future. OpenLaw is an initiative working towards that idea. http://openlaw.io/

Still a long way to mature

- The concept is still in an early phase. That's why we cannot see a decentralized application which have gone mainstream and disrupted an industry
- Questionable if it is ideal beyond the finance/business sector even though the concept of ERC-20 tokens aims to boost adoption via a universal standard.
- Oracles are able to link events from the outside world and combine information while the contract code is executed. Oracles can include business logic, laws and other agreed terms within a contract. However, this concept is still in an early stage. Oraclize is a promising project. http://www.oraclize.it/
  - E.g. my insurance company is compensating me in case of flood according to the amount of water entered into my field – How is this going to be recorded reliably? What kind of oracle is going to be used?
- Legality of smart contracts is questionable
- Privacy? Current implementations are mostly public
- Difficult to understand for non-programmers
- Scalability issues – every transaction needs to be processed by every node

Ethereum, progress so far and roadmap

- Ethereum is so far, one of the most highly crowdfunded project globally, gathering a staggering 31,529.49449551 BTC by September 3rd 2014 (address). Total amount of Bitcoins received as of March 2018 is approximately 31,550.50.
- To perform everything the team is poised for, in a scalable and secure manner is a very tall order in itself. Several implementations of the Ethereum VM already exist, including C++, Go, Java, Python, Javascript, Haskell bkirwi& jamshidh , Node, .NET
- Homestead is the second release of the Ethereum project, moving beyond developers and to the mainstream, after the successful hard fork towards it. This is not to be confused with the DAO sustained hard fork which happened later, and resulted in two versions of the protocol and two chains (ETH and ETC).
- "Metropolis" is the third release with the aim to reduce complexity of the EVM and provide flexibility for smart contract developers. zk-SNARKs and ring signatures support is added. It is divided into 2 steps: Byzantium and Constantinople which is still expected.
- The last phase is Serenity – the conversion of the Ethereum Network from Proof-of-Work to Proof-of-Stake

The World Computer?

- While Frontier allowed only for command line, the production release of Ethereum called Homestead was released via a hard fork of the blockchain, and it allows users to build more on the platform. More information on the improvements of Homestead can be found here
- A private version of the Ethereum network served as the platform for the first major test conducted by blockchain consortium startup R3CEV in January, 2016, with the trial uniting 11 major banks in a high-profile proof-of-concept.
- More resources and use cases are springing daily, making the Ethereum blockchain grow far faster than Bitcoin's ever has.
- A consortium of large companies including JPMorgan, Intel, Microsoft And Others formed the Enterprise Ethereum Alliance (EEA), which aims at creating a standard version of the Ethereum software that businesses around the world can use to track data and financial contracts.
- The most recent update came in October 3, where the EEA and Hyperledger announced that they are joining each other's organizations to enable *"more active and mutual cross-community collaboration through event participation, connecting with other members, and finding ways for our respective efforts to be complementary and compatible".* The goal is to accelerate the evolvement of blockchain technology for businesses.

The DAO hack and the ensuing fallout

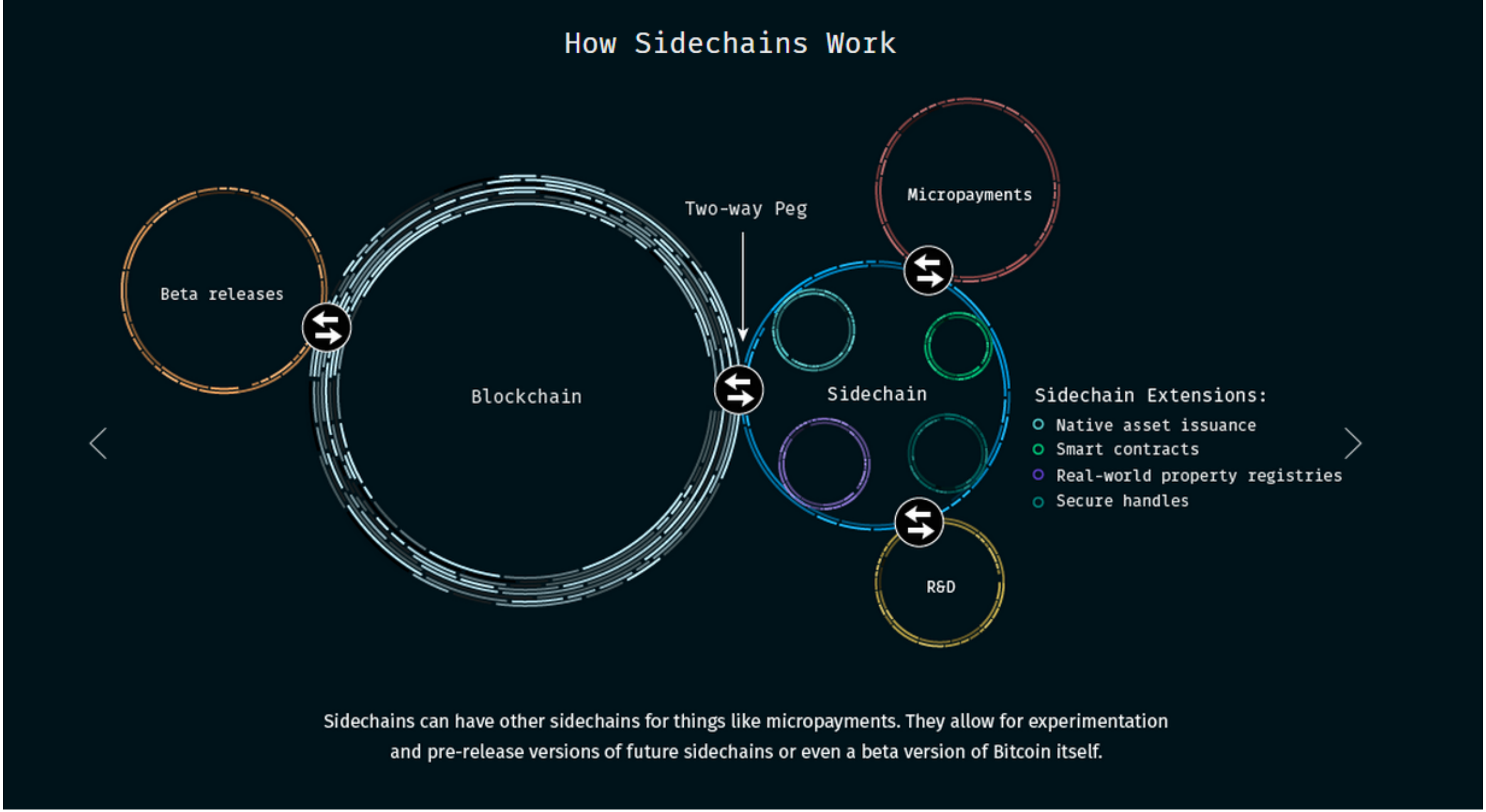
- "The DAO" (Decentralized Autonomous Organization) had a formidable calling, to create the first decentralized crowdfunding platform, a place where investors would have proportional decision making ability on the investment of funds which a decentralized organization held. At its height it gathered about $160 million (at that time), and became the largest crowdfunded project ever by then.
- Despite criticism on the "too much, too fast, too early" nature of the project while it was starting, it went on, and on Friday June 17th 2016, an attacker syphoned about $50 million worth of the native tokens away. The exploit used was suggested as an attack vector before, and was even, reportedly, fixed.
- The proposed solution by the ETH community was a hard fork to remove the funds from the attacker. This caused a split in the community as not everyone was in favor of "bailing out" the DAO since it was a construct on ETH and not an ETH vulnerability itself. This led to a hard fork and the creation of two Ethereum blockchains. The majority one (retained the Ethereum name) and the minority one was named Ethereum Classic.
- Some further reading :
- https://www.cryptocompare.com/coins/guides/the-dao-the-hack-the-soft-fork-and-the-hard-fork/
- http://qz.com/730004/everything-you-need-to-know-about-the-ethereum-hard-fork/

Ethereum switch to Proof of Stake (PoS)

- Several very novel approaches are being implemented in Ethereum, including an improved version of the GHOST protocol to decrease block times, and the transition to Proof of Stake, called Casper.
- One reason for this switch from Proof of Work (PoW) to PoS is environmental friendliness as PoW is very wasteful in terms of electricity usage
- It was also taken into consideration that if PoW remained in place, people with money and resources to mine the most ETH would be the majority of miners, therefore creating an unbalanced distribution of wealth and mining power. The average person can not afford to set up and maintain a mining rig and this is directly opposite to the ideals of a decentralized economy
- PoS addresses this issue by making the mining process affordable to the average person. This allows the mining environment to continue to grow and attract more participants, supporting the essence of a decentralized economy
- This is because under PoS, in order to mine ETH a miner needs to own a certain amount of ETH staked for the mining operation. The amount of ETH mined would be based on the amount staked
- The calculations processed in POS are simpler to solve than PoW. As a result no wasteful miners are needed.
- The obvious drawback is that miners owning a considerable amount of ETH have a distinct advantage over new miners entering the game, but in any case it is a step towards the right direction

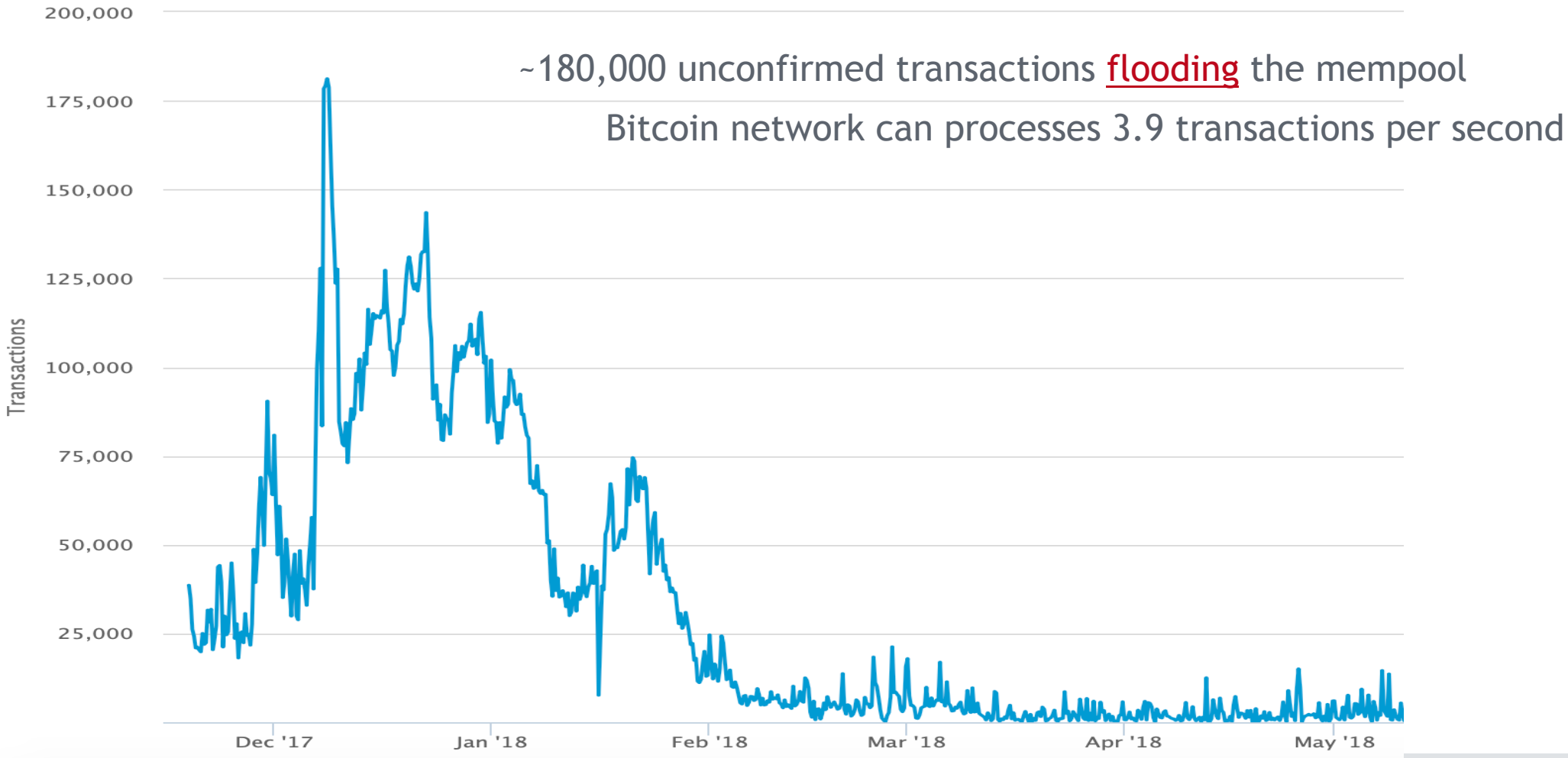
Scalability Issues - The problem

- To achieve decentralization, Bitcoin was designed so that anyone would be able to run a node from a personal computer, without a need for specialized hardware or dedicated high-end servers.
  - In other words, an average home computer should be able to perform all tasks needed to constantly maintain consensus with other peers, such as verifying transactions and storing the entire blockchain.
  - The biggest problem is transaction verification, since hashing operations and signature verifications take time – this imposes processing and bandwidth issues.
  - Storing the entire blockchain imposes storage issues.
- To address some of these issues, the Bitcoin network limits the size of each block and, hence, the number of transactions it can carry.
  - The current limit stands at 1Mb per block.
  - At the time of writing this, average block sizes are quickly approaching to the limit.
  - As the network grows to reach the sizes of competing payment networks – the current limits are going to become problematic.


How Sidechains Work
Micropayments
Two-way Peg
Beta releases
Blockchain
Sidechain
Sidechain Extensions:
Native asset issuance
Smart contracts
Real-world property registries
Secure handles
R&D
Sidechains can have other sidechains for things like micropayments. They allow for experimentation
and pre-release versions of future sidechains or even a beta version of Bitcoin itself.

Alternative uses of the blockchain technology

- Mempool of Unconfirmed transactions to be processed by the network.



~180,000 unconfirmed transactions flooding the mempool

Bitcoin network can processes 3.9 transactions per second

Transactions speeds (per second)

Source (Jan 2018): https://howmuch.net/articles/crypto-transaction-speeds-compared

Scaling Ethereum

- Layer 1 Solution: What if each node does not have to process each transaction
  - Aim: Increase Ethereum network transaction capacity
  - Notable Proposal: **Sharding**

- In Sharding, the idea is to split the network into many shards each one containing their own dependent piece of blockchain history. Someone can see this system as many small blockchains running with their own validating nodes which would increase the throughput of transactions processed.

- To avoid a single-shard attack the idea is a random sampling of validators on each shard. Validators will not know which shard they will get. Many validators will work on each shard and the ones that will actually be validating will be randomly selected from this set.

For further explanation of Sharding click here

Scaling Ethereum

- Layer 2 Solutions – built on top of the Ethereum main-chain:

- **State channels:** The equivalent of payment channels in blockchain – can be used for payments and blockchain state updates such as modifications in a smart contract
- **Plasma:** Creation of “child blockchains” attached to the main-chain. Child blockchains can create their own “child blockchains” and so on. The interaction with the main-chain is very limited, while thousands of decentralized applications can run on the child blockchains faster and with lower confirmation times.

What is a smart contract?

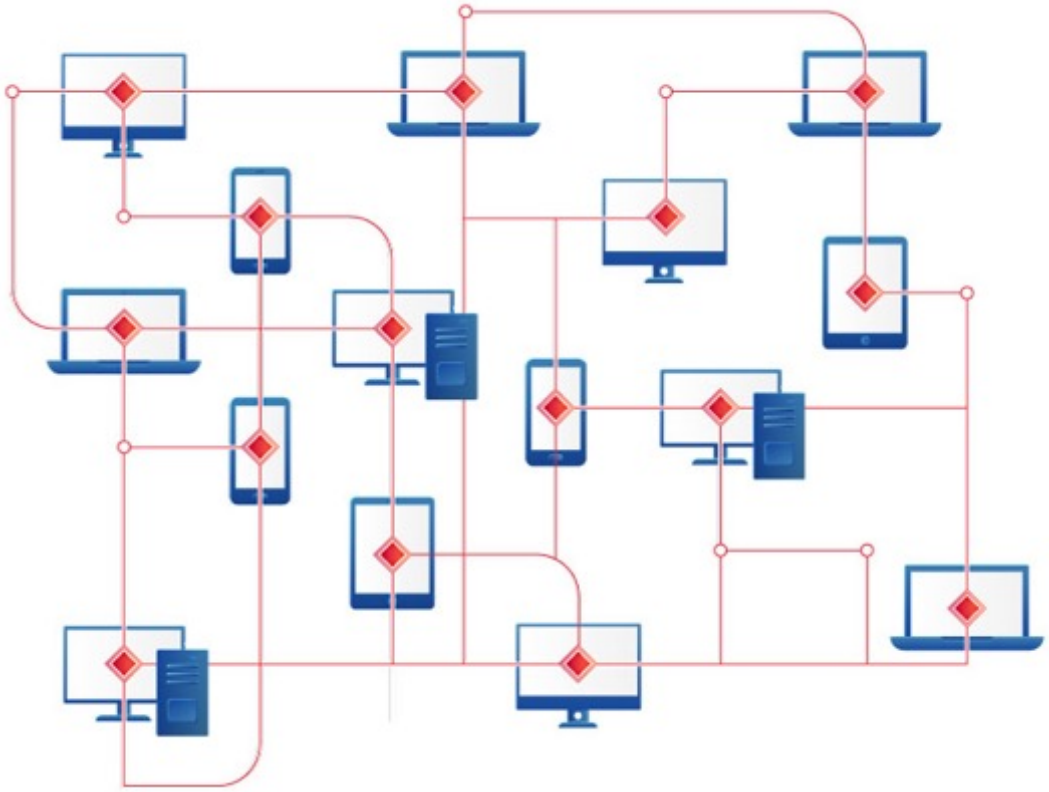
- "… is a piece of code that leaves on the Ethereum blockchain, it can be instructed to do a certain action, by having a person, or another contract send a message to it …"

- Smart Contracts = the absolute core of what Ethereum is!

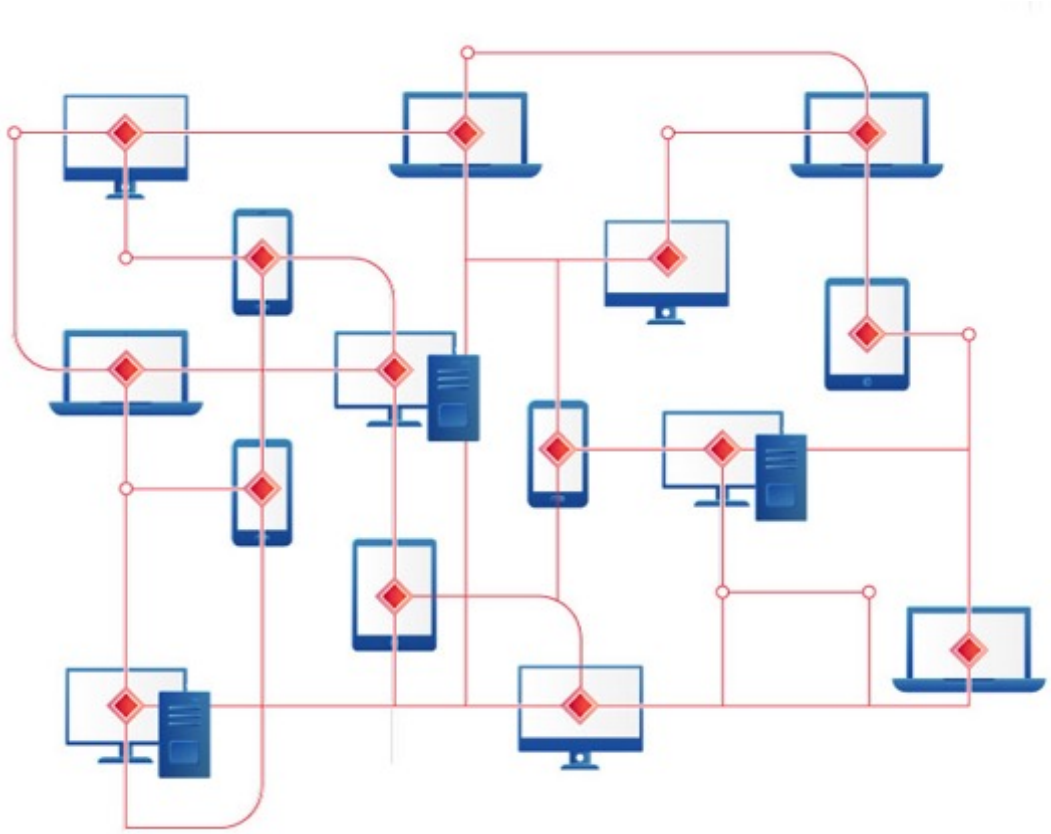
13 July 2015 – Ethereum goes Live!

How the eco-system of Ethereum Works?

- FACT: On the Ethereum Network we are working with a network of computers that communicate with each other.
- FACT: On Ethereum networks we can
  - transfer money between nodes, and
  - we can store data on each node.
- FACT: Many Ethereum Networks
  - Main network where 1 Ether = some $
  - Test Ethereum Networks
  - Private Eth networks (e.g., on our machine)
  - Private Eth network with open access



How the eco-system of Ethereum Works?



- In this Workshop we will be creating our own private network…
- FACT: The ETH network consists of Nodes
  - Each node is any machine that runs the ETH client.
  - All nodes connect to each other to form the network.
- FACT: Each node holds its own replica of the full blockchain (ledger to transactions)
- **So, What is a Blockchain!??**

An Abstract Definition of a Blockchain

- "… let's think of the blockchain as a database that stores a *record* of every transaction that has ever taken place … "

Amount	From	To
$30.00	Darcy	Tom
$5.25	Jane	Elizabeth
$123.43	Charlotte	Jane

Transactions Table

Potential Directions

For Bitcoin & Ethereum

What is the future of the blockchain?

As we have seen, the Bitcoin blockchain can be used for various purposes. Amongst other things, experts envision that the blockchain concept may be further used to keep:

- **Public Records**, for instance:
  - Land titles (as is currently explored with Factom)
  - Criminal records
  - Voter records
  - Court records
- **Private Records**, for instance:
  - Wills
  - Trusts
- **Other uses**, for instance:
  - Certifications (like our university uses to store certificates of MOOC completion)
  - Medical records (like the Hashed Health initiative)
  - Supply chain Management
  - Shipping

Prove Ownership and get Compensation via Blockchain

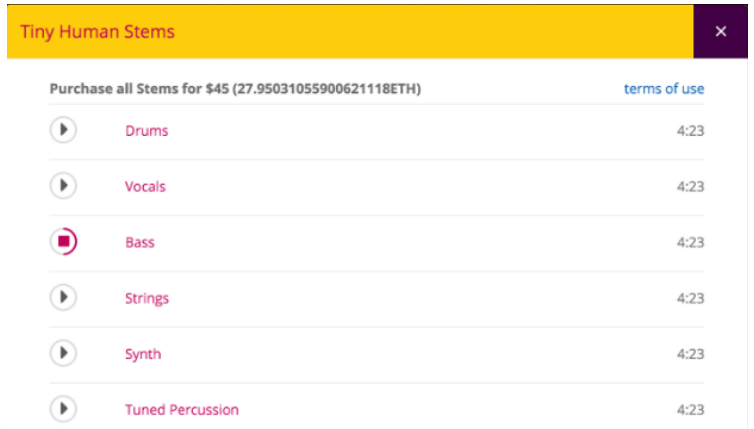
- Blockchain can store a cryptographic hash representing a new song's:
  - Artist
  - Composer
  - Title
  - Official Video/Audio
  - Any other relevant information
- Ownership is registered permanently therefore no need for record labels to have a share of the artist's work
- UjoMusic - Based on the Ethereum blockchain and allows artists to publish their work immediately after uploading and manage licensing on their own terms.
- Users fund their accounts with Ether
- Smart Contracts technology allows artists to set automated payments to them based on licenses they design themselves

https://ujomusic.com/

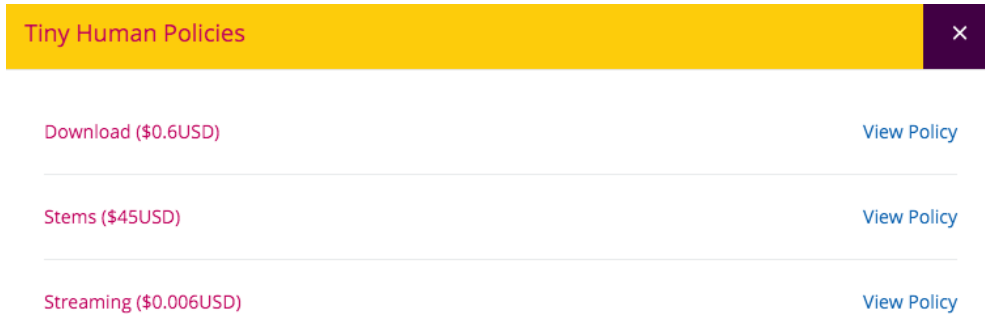
http://myceliaformusic.org/

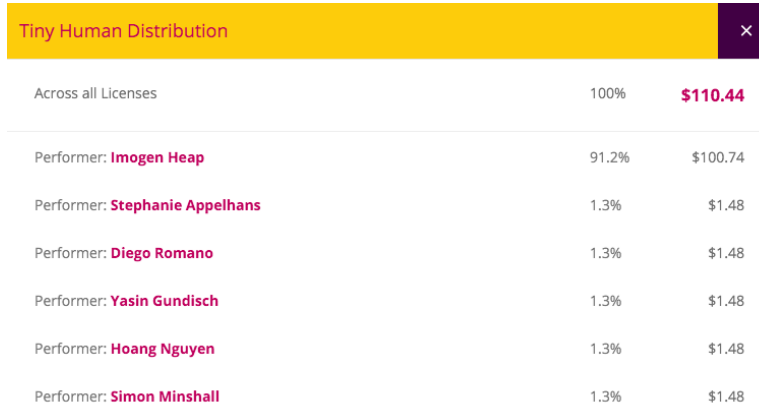
Imogen Heap's 'Tiny Human' was the only track available on Ujo as an initial attempt. (almost 150 purchases)
Latest updates:
https://blog.ujomusic.com/announcing-the-ujo-alpha-91f4489f6110
https://techfinancials.co.za/2017/11/29/ujo-music-using-blockchain-revolutionise-music-industry/

Ujo's interface – During the pilot run

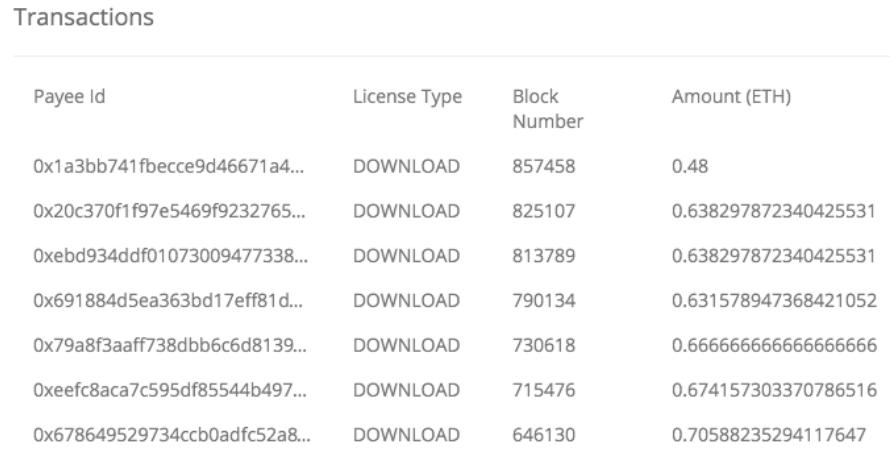


Each stem of the song could be individually purchased





Ownership and artist compensation publicly visible in a "smart contract"



Every transaction publicly available

Academic Certificates - Blockchain Solutions

- Ease of Publication & Distribution
- Independent validation
- Immutable Records - Digital fingerprints (hashes) of the individual certificates issued, are placed permanently in a blockchain transaction
- Reduced time to issue Certificates
- Costs of re-issuing certificates in the case the hard copy is lost are minimal
- Ease and instant authentication by interested parties (e.g. employers) even if the application used or the institution's website no longer exists. Operational costs minimized
- Universities and issuing authorities protect their brand names from being tarnished
- Employers can examine job applications more efficiently, ensuring that a candidate employee is presenting true information, without long waiting times or processing costs
- Our solution: http://block.co/our-approach/

Real Estate Management - Blockchain Solutions

- Authenticity: Property holders could digitally prove and transfer ownership immediately without the need to pay and wait for third-party verification
- Eliminate fraud and costs: Funds of sender and recipient can be logged using the multisig technology and be triggered upon smart contract execution i.e. transfer a land title when funds are received. A “digital ownership certificate” cannot be replicated, and can be linked to one property in the system, making selling or advertising properties you don’t own almost impossible. No further middlemen, paper work and delays
- Transparency: Creation of unique digital IDs for real estate assets, buyers and sellers. Enable faster mortgage process and transfer of ownership. For the buyer, credit history and income could be instantly verifiable, avoiding time-consuming tasks involving banks, lawyers and estate agents. Homeowners can prove ownership and time of residence within a property. For assets, digital identities could be assigned, which would include the chain of ownership, list of repairs etc.
- Examples: Bitfury & velox.re

Supply Chain Shipping – Blockchain Solutions

- Digitize Supply Chain Process
- Track the paper trails of shipping containers
- Reduce time spent in transit and shipping process
- Enhance transparency and security of product information exchanged between parties
- Reduce costs and complexity
- Improve stock management
- Reduce fraud and errors on the quality of products
- Examples: IBM Watson

Image Source: https://www.sgkglobal.com/

Solar Energy Management - Blockchain Solutions

- Example: The Brooklyn Microgrid
- **Transparency** through the whole process
- Decentralized and direct buying/selling of energy among participants(mostly electricity) – Independence from a third party power provider
- Storage of transaction data and recording of electricity generated per participant within a network
- Smart contracts application on distribution upon smart devices recordings
- Blockchain technology can allow a neighborhood or a region to put together an energy trading system derived from solar panels, to record transactions between locals. This would save them money and hassle
- Users can trade excess energy between them instead of selling it back to the power company. Participants will be able to access the transparent ledger any time they wish. Participants can decide **how much, at what price and to whom to sell their excess energy**, while all the transactions will be recorded on the Blockchain.

https://www.energymatters.com.au/panels-modules/choosing-solar-panels/

Decentralized Training Series Workshop: Certified Ethereum Specialist

19-20 November 2018, Athens, Greece

UNIC | Institute For the Future

CERTH CENTRE FOR RESEARCH & TECHNOLOGY HELLAS | iti Information Technologies Institute