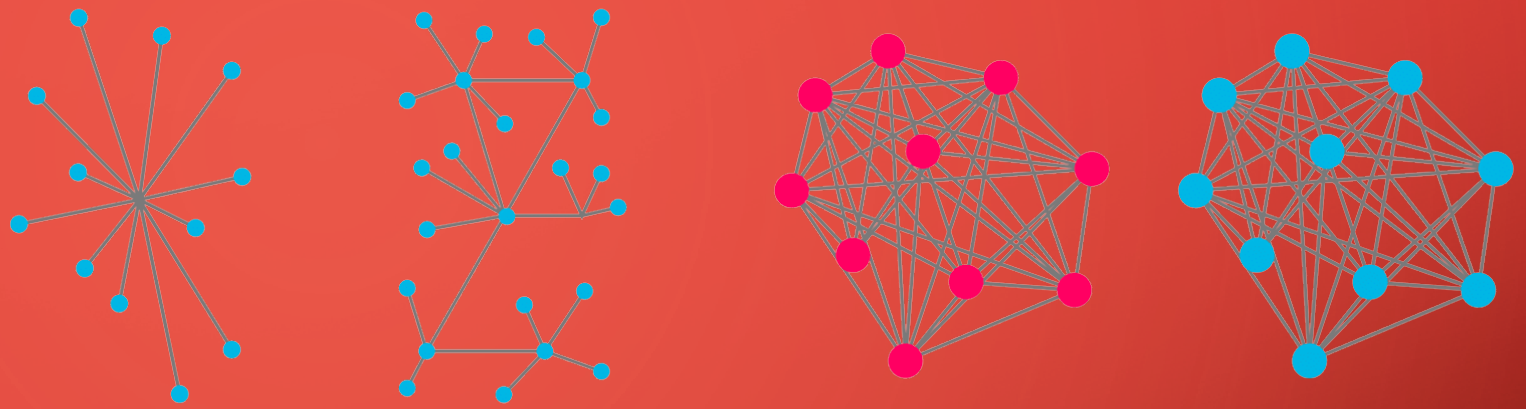


BLOCKCHAIN

LET'S ALL AGREE ON WHAT THIS IS ABOUT

HUGO FLAYAC 2019

THE BITCOIN ERA

“Satoshi Nakamoto”, October 31, 2008

Bitcoin: A Peer-to-Peer Electronic Cash System

Strongly inspired by **1998 ideas**: B-Money and BitGold

S. Nakamoto:
Unknown person or people who developed Bitcoin, authored the bitcoin white paper, and created and deployed bitcoin's original reference implementation…

THE BITCOIN PAPER ABSTRACT (KEYWORDS)

- 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.

PEER TO PEER

Distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes.

"What is needed is an electronic payment system [...], allowing any two willing parties to transact directly with each other without the need for a trusted third party."

ELECTRONIC CASH

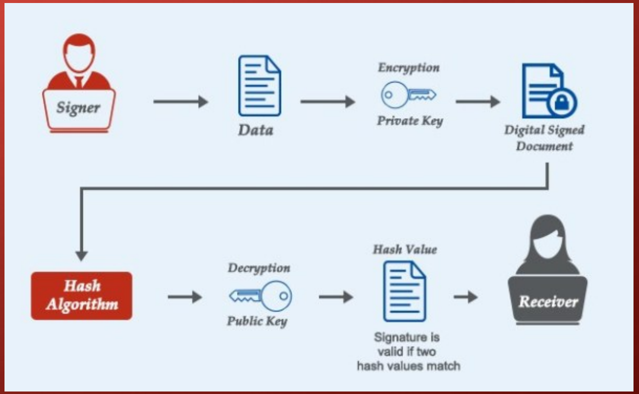
Money kept in electronic form which allows a buyer to pay for goods and services on the internet without using a credit card

“Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model”

DIGITAL SIGNATURES

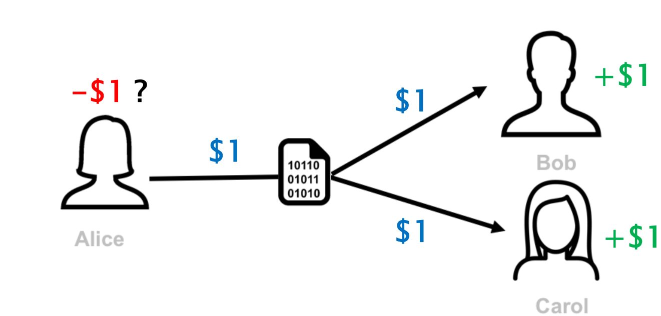
Mathematical scheme for verifying the authenticity of digital messages or documents. A valid digital signature, gives a recipient very strong reason to believe that the message was created by a known sender (authentication), and that the message was not altered in transit (integrity)

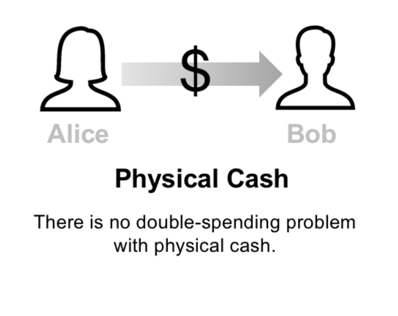
“Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending.”



DOUBLE SPENDING

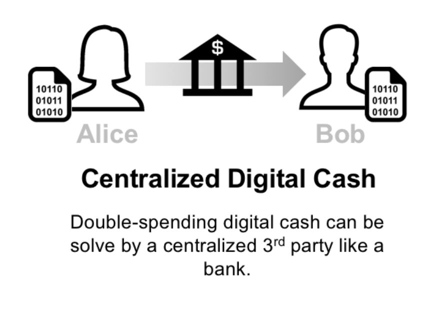
Spending the same amount of money twice e.g. Alice sends $1 to Bob, he cannot be sure that Alice has -$1 on her account and she could spend the same dollar to Carol and many others.





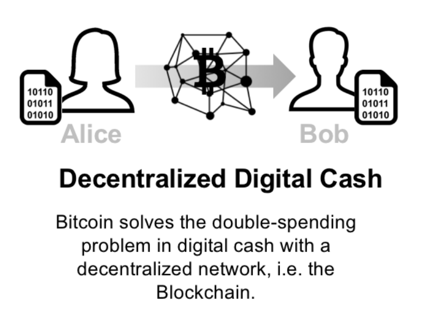
Physical Cash

There is no double-spending problem with physical cash.



Centralized Digital Cash

Double-spending digital cash can be solve by a centralized 3rd party like a bank.



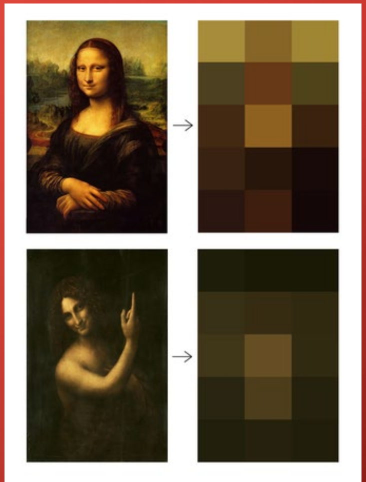
Decentralized Digital Cash

Bitcoin solves the double-spending problem in digital cash with a decentralized network, i.e. the Blockchain.

“We propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions”

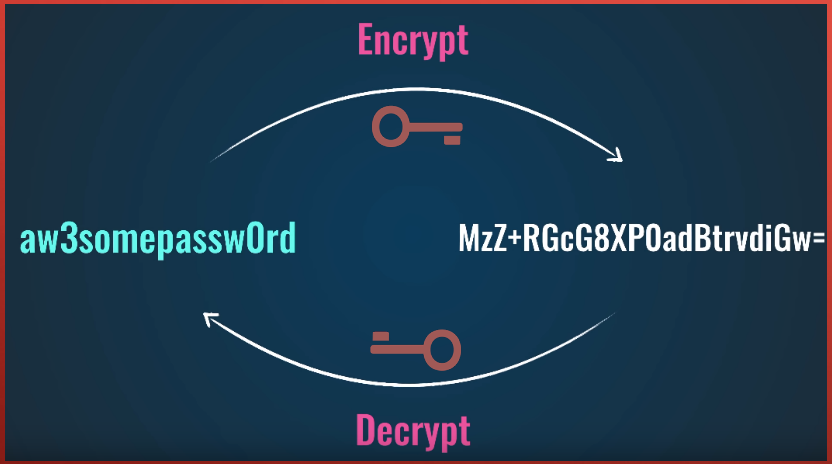
HASHING

Image Hash

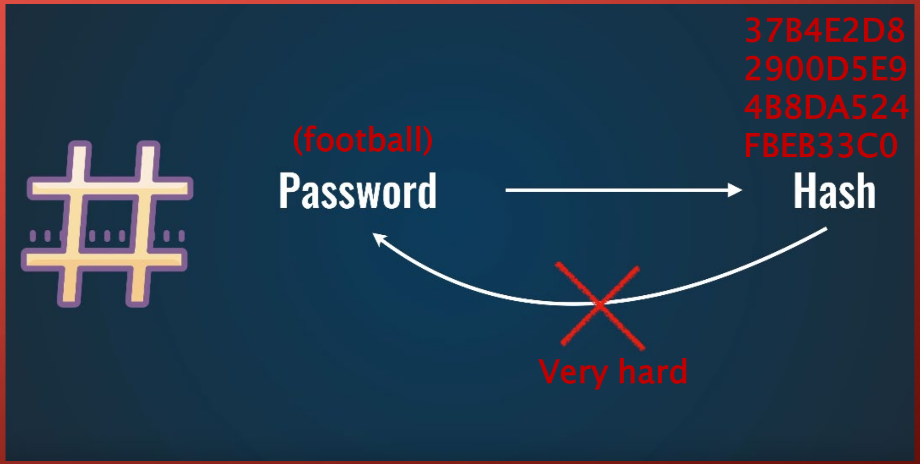


A hash function is any function that can be used to map data of arbitrary size onto data of a fixed size. The values returned by a hash function are called “hash” values or “hashes”

Encryption



Hashing

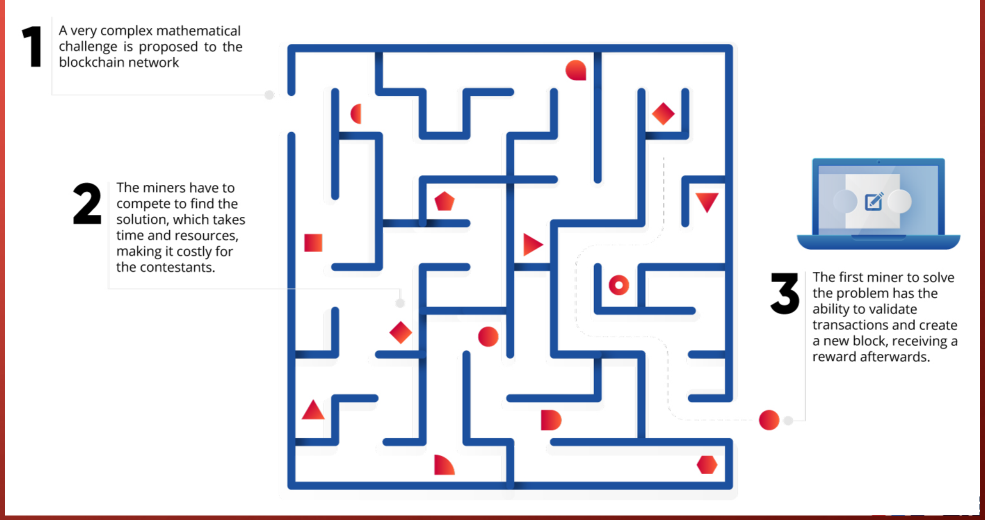


No need to store the Password

PROOF OF WORK (POW)

A special type of participants in the network called miners compete on searching for the solution to a cryptographic puzzle that will allow them to add a block of transactions to Bitcoin’s blockchain.

> “ 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”



1 A very complex mathematical challenge is proposed to the blockchain network

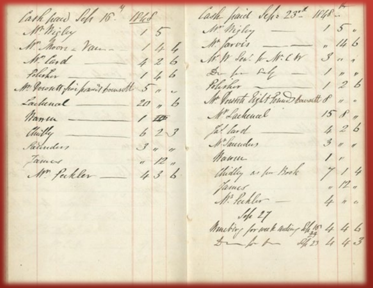
2 The miners have to compete to find the solution, which takes time and resources, making it costly for the contestants.

3 The first miner to solve the problem has the ability to validate transactions and create a new block, receiving a reward afterwards.

BLOCKCHAIN

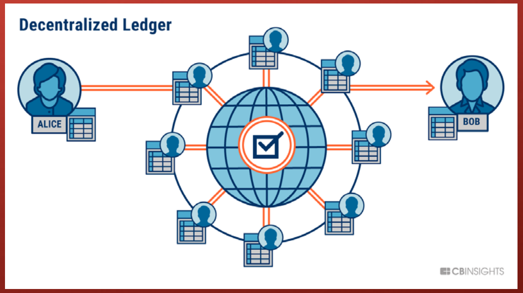
To solve the double-spending problem, Nakamoto proposed a public ledger to keep track of all transactions in the network the Blockchain:

- **Distributed:** The ledger is replicated across a number of computers. Any computer with an internet connection can download a full copy of the blockchain.
- **Cryptographic:** To make sure that the sender owns the bitcoin that she's trying to send, and to decide how the transactions are added to the blockchain.
- **Immutable:** transactions can only be added to the blockchain but cannot be deleted or modified.



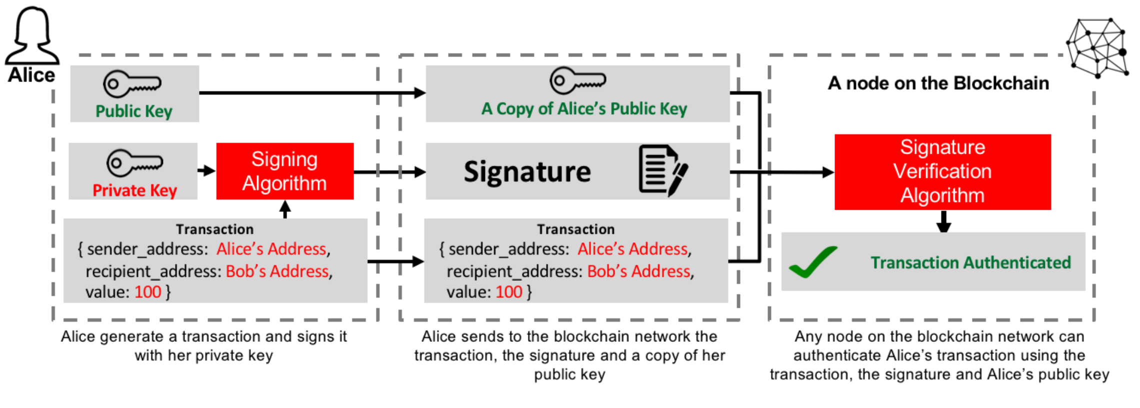
Cash paid Sept 16th 1848			
Mr Wigley	1	5	
Mr Moore – Van	1	4	4
Mr Card	4	2	6
Polisher	1	4	6
Mr Dowsett five pound [illegible]	5	"	"
Lachenal	20	"	6
Warren	1	[illegible]	
Chidley	6	2	3
Saunders	3	"	"
James	"	12	"
Mr Peckler	4	3	6

Cash paid Sept 23rd 1848			
Mr Wigley	1	5	"
Mr Jarvis	"	14	6
Mr W Sen: for Mr C W	3	"	"
Do for [illegible]	1	"	"
Polisher	1	2	6
Mr Dowsett Eight Pound [illegible]	8	"	"
Mr Lachenal	15	8	"
Jas. Card	4	2	6
Mr Saunders	3	"	"
Warren	1	"	
Chidley as per Book	7	1	4
James	"	12	"
Mr Peckler	4	"	"
Sept 27			
Numbering for week ending Sept 16/23	4	4	6
Do for do Sept 23	4	4	3



SENDING BITCOIN MONEY

1. **Create a wallet:** Stores Public/Private Key but not the bitcoins
2. **Create a transaction:** connect to wallet with Private Key, define Amount and Public Key of the receiver
3. **Broadcast to the network**: the transaction is sent to the entire network (including adress)
4. **Confirm the transaction**: a Miner the transaction autenticates the transaction using the sender public key, confirms the amount is OK and adds the transaction to the block
5. **Broadcast the change:** the miner should broadcast the blockchain change to all Miners to make sure that their copies of the blockchain are all matching.


Alice
Public Key
Private Key
Signing Algorithm
A Copy of Alice's Public Key
Signature
A node on the Blockchain
Signature Verification Algorithm
Transaction Authenticated

Transaction
{ sender_address: Alice's Address,
recipient_address: Bob's Address,
value: 100 }

Transaction
{ sender_address: Alice's Address,
recipient_address: Bob's Address,
value: 100 }

Alice generate a transaction and signs it with her private key

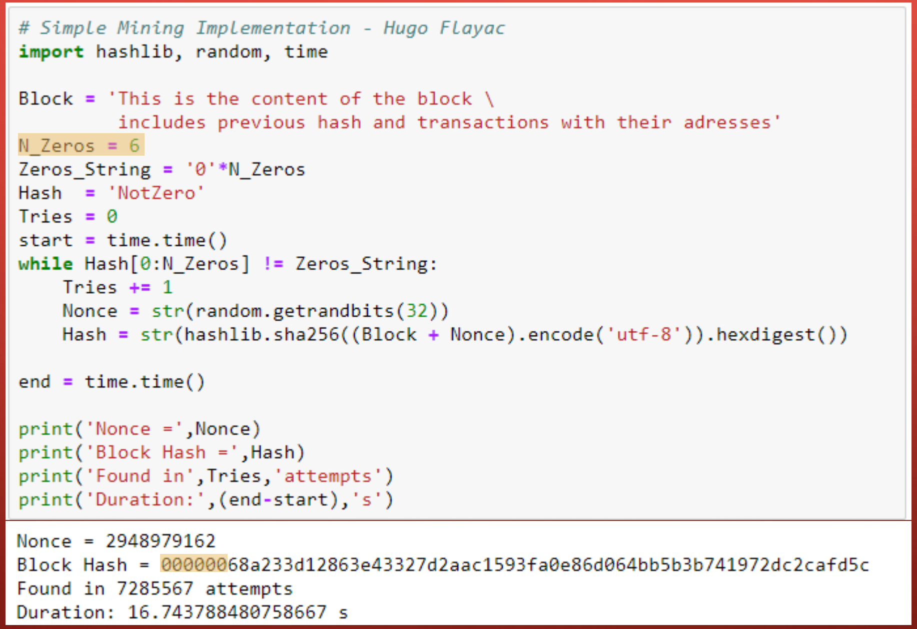
Alice sends to the blockchain network the transaction, the signature and a copy of her public key

Any node on the blockchain network can authenticate Alice's transaction using the transaction, the signature and Alice's public key

Authentication Process for Transactions on the Blockchain

THE MINING PUZZLE

- Bitcoins uses a Hash function called SHA-256 (64 characters)
- Applied to Block Data (BTC transactions) + Nonce (32B number)
- Change Data and/or Nonce → Different Hash
- Block “valid” or “mined” if Nonce → Hash starts with 000…0
- Difficulty changes every 2016 blocks: $t = difficulty \times 2^{32} / hashrate$



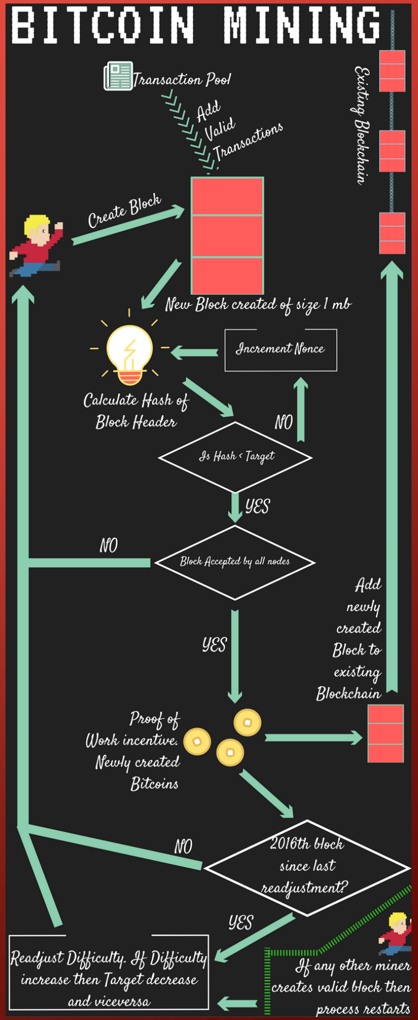
```python
# Simple Mining Implementation - Hugo Flayac
import hashlib, random, time

Block = 'This is the content of the block \
         includes previous hash and transactions with their adresses'
N_Zeros = 6
Zeros_String = '0'*N_Zeros
Hash  = 'NotZero'
Tries = 0
start = time.time()
while Hash[0:N_Zeros] != Zeros_String:
    Tries += 1
    Nonce = str(random.getrandbits(32))
    Hash = str(hashlib.sha256((Block + Nonce).encode('utf-8')).hexdigest())

end = time.time()

print('Nonce =',Nonce)
print('Block Hash =',Hash)
print('Found in',Tries,'attempts')
print('Duration:',(end-start),'s')
```

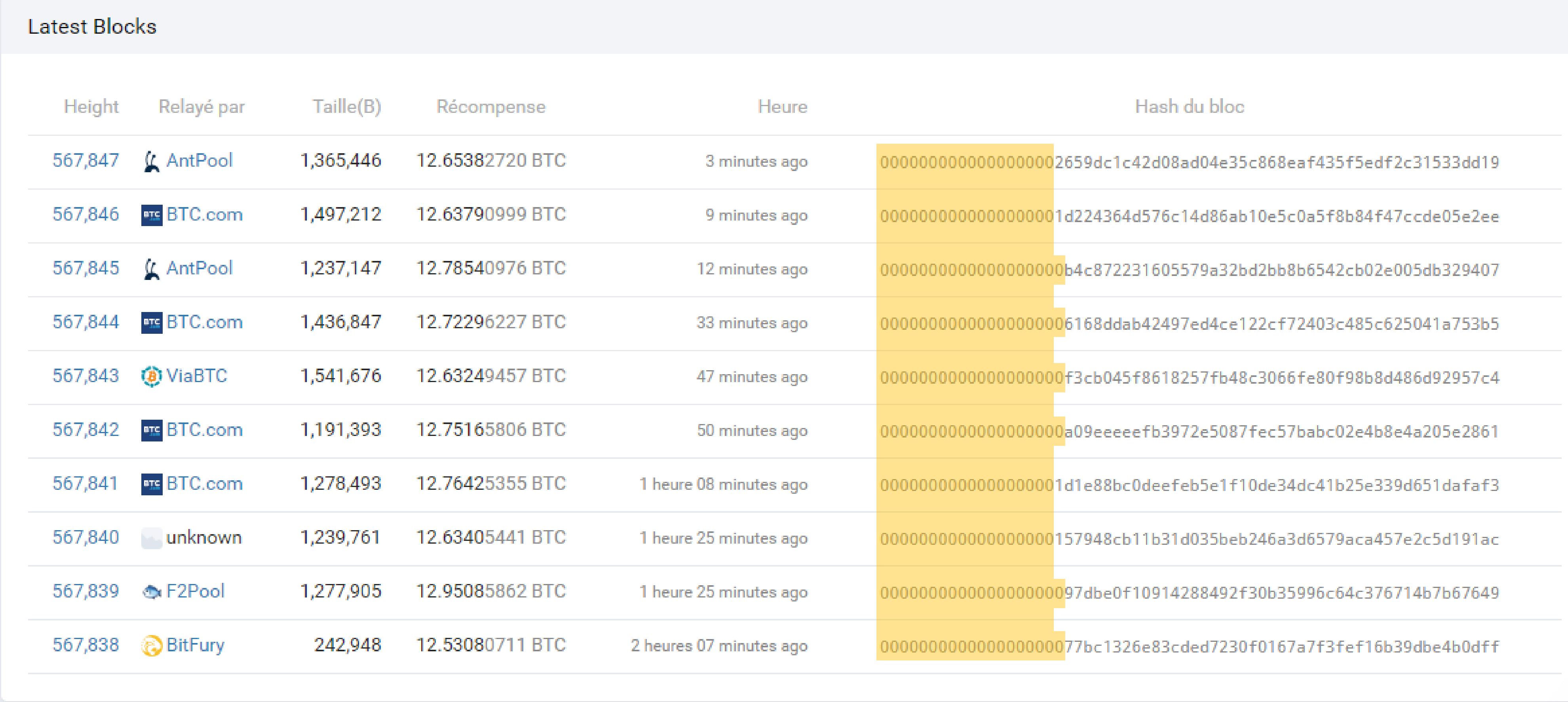
```
Nonce = 2948979162
Block Hash = 00000068a233d12863e43327d2aac1593fa0e86d064bb5b3b741972dc2cafd5c
Found in 7285567 attempts
Duration: 16.743788480758667 s
```



BITCOIN MINING
Transaction Pool
Add Valid Transactions
Existing Blockchain
Create Block
New Block created of size 1 mb
Increment Nonce
Calculate Hash of Block Header
NO
Is Hash < Target
YES
NO
Block Accepted by all nodes
YES
Add newly created Block to existing Blockchain
Proof of Work incentive. Newly created Bitcoins
NO
2016th block since last readjustment?
YES
Readjust Difficulty. If Difficulty increase then Target decrease and viceversa
If any other miner creates valid block then process restarts

THE MINING PUZZLE

12,5 BTC = $50’000

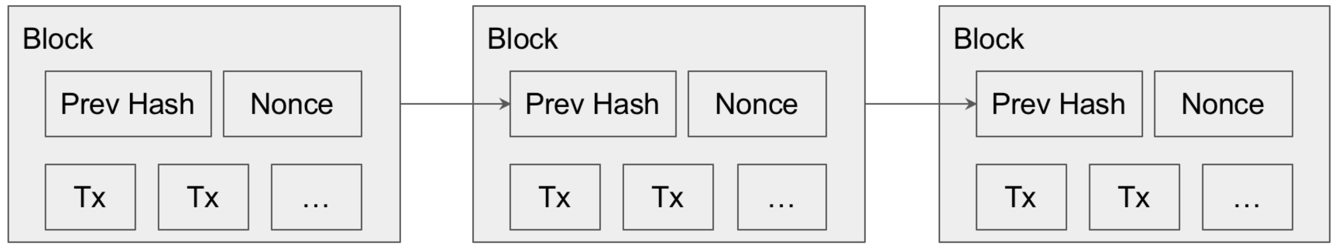


Latest Blocks

Height	Relayé par	Taille(B)	Récompense	Heure	Hash du bloc
567,847	AntPool	1,365,446	12.65382720 BTC	3 minutes ago	0000000000000000002659dc1c42d08ad04e35c868eaf435f5edf2c31533dd19
567,846	BTC.com	1,497,212	12.63790999 BTC	9 minutes ago	0000000000000000001d224364d576c14d86ab10e5c0a5f8b84f47ccde05e2ee
567,845	AntPool	1,237,147	12.78540976 BTC	12 minutes ago	0000000000000000000b4c87223160 5579a32bd2bb8b6542cb02e005db329407
567,844	BTC.com	1,436,847	12.72296227 BTC	33 minutes ago	00000000000000000006168ddab42497ed4ce122cf72403c485c625041a753b5
567,843	ViaBTC	1,541,676	12.63249457 BTC	47 minutes ago	0000000000000000000f3cb045f8618257fb48c3066fe80f98b8d486d92957c4
567,842	BTC.com	1,191,393	12.75165806 BTC	50 minutes ago	0000000000000000000a09eeeeefb3972e5087fec57babc02e4b8e4a205e2861
567,841	BTC.com	1,278,493	12.76425355 BTC	1 heure 08 minutes ago	0000000000000000001d1e88bc0deefeb5e1f10de34dc41b25e339d651dafaf3
567,840	unknown	1,239,761	12.63405441 BTC	1 heure 25 minutes ago	000000000000000000157948cb11b31d035beb246a3d6579aca457e2c5d191ac
567,839	F2Pool	1,277,905	12.95085862 BTC	1 heure 25 minutes ago	000000000000000000097dbe0f10914288492f30b35996c64c376714b7b67649
567,838	BitFury	242,948	12.53080711 BTC	2 heures 07 minutes ago	000000000000000000077bc1326e83cded7230f0167a7f3fef16b39dbe4b0dff

FROM BLOCKS TO BLOCKCHAIN

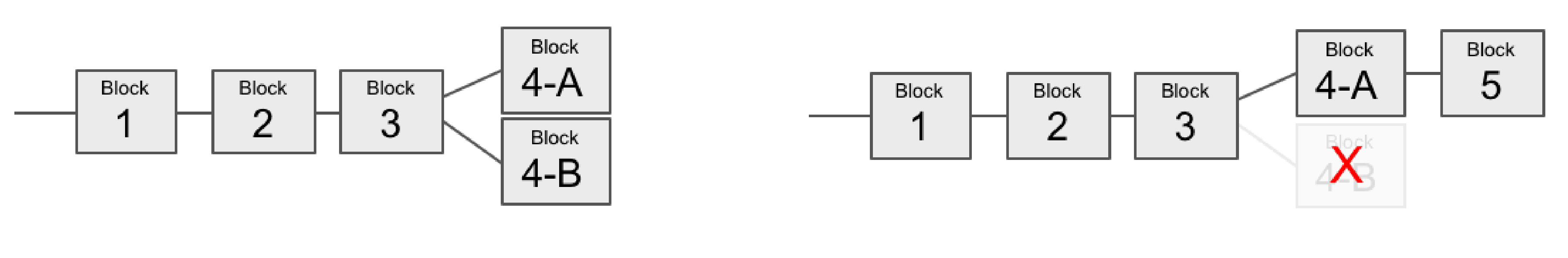
- Transactions are grouped in blocks and blocks are appended to the blockchain
- Each new block uses the previous block's hash as part of its data
- To create a new block:
  - Miner selects a set of transactions
  - Adds the previous block's hash
  - Tries to mine the block
- Changes to the data in a block will affect all the hash values of the following blocks



Blocks are chained together using the previous block's hash to form a Blockchain.

FROM BLOCKS TO BLOCKCHAIN

- All the miners in the Bitcoin network compete with each other to find the next valid block
- Given the number of miners, the probability of a miner in the network validating a block is high
- What happens if two miners or more submit their blocks at the same time?

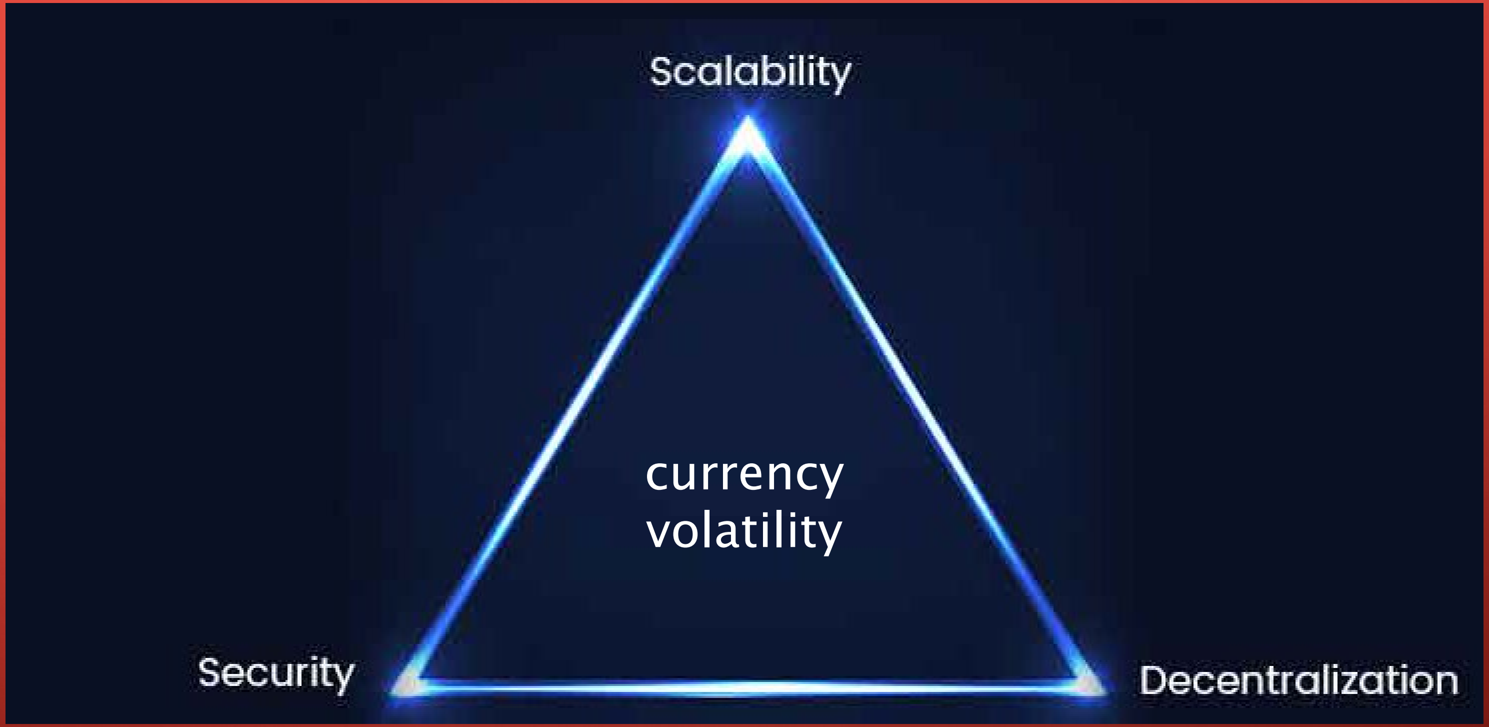


Resolving conflicts - The longest chain wins

ATTACKING THE BLOCKCHAIN

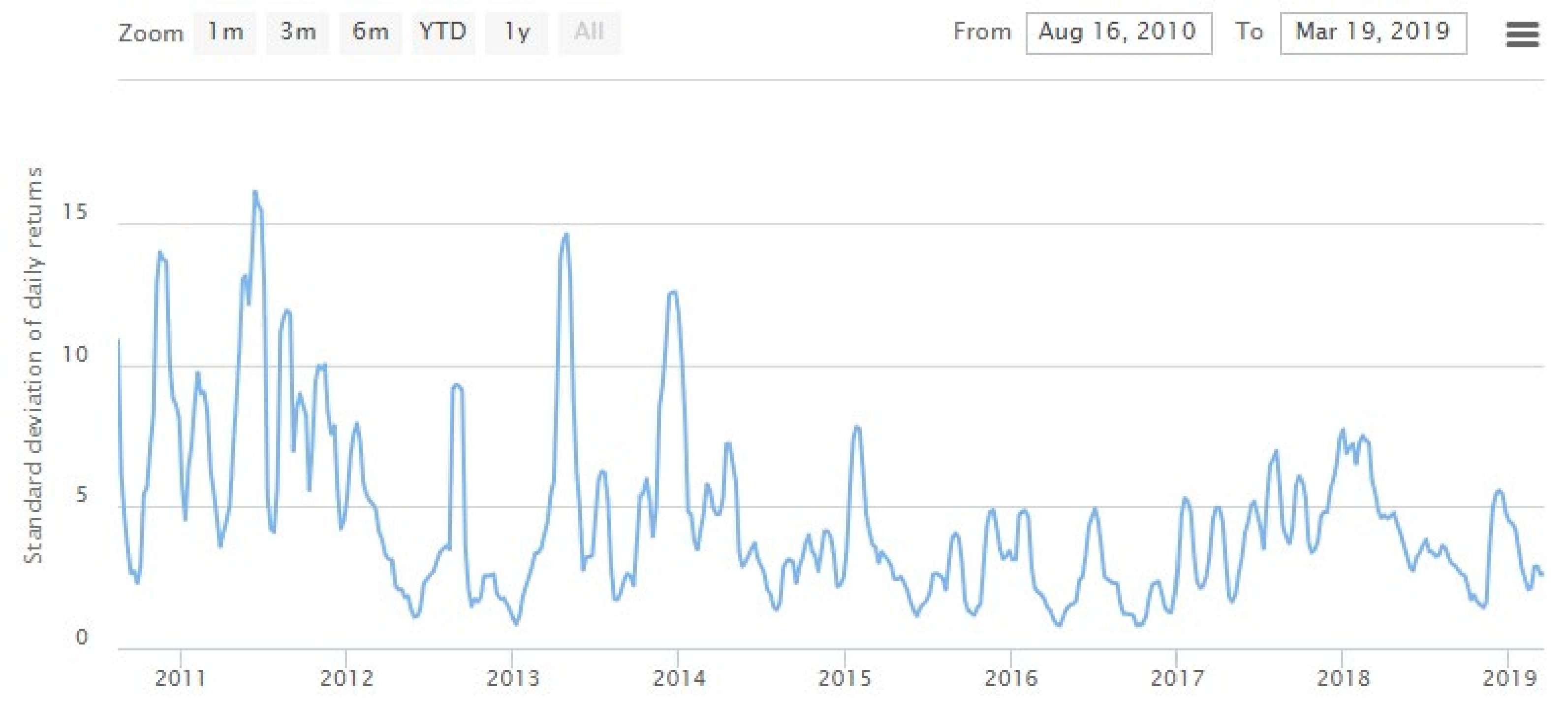
- Race Attack: An attacker sends the same coin in rapid succession to two different addresses. It is recommended to wait for at least one block confirmation (10 mins) before accepting the payment.
- Finney Attack: An attacker pre-mines a block with a transaction, and spends the same coins in a second transaction before releasing the block. In this scenario, the second transaction will not be validated. To prevent from this attack, it is recommended to wait for at least 6 block confirmations (60 mins) before accepting the payment.
- Majority 51% Attack: The attacker owns 51% of the computing power of the network. The attacker starts by making a transaction that is broadcasted to the entire network, and then mines a private blockchain where he double-spends the coins of the previous transaction. Since the attacker owns the majority of the computing power, he is guaranteed that he will have a longer chain.

BLOCKCHAIN TRI-LEMMA



BLOCKCHAIN QUADRI-LEMMA

Currency Volatility



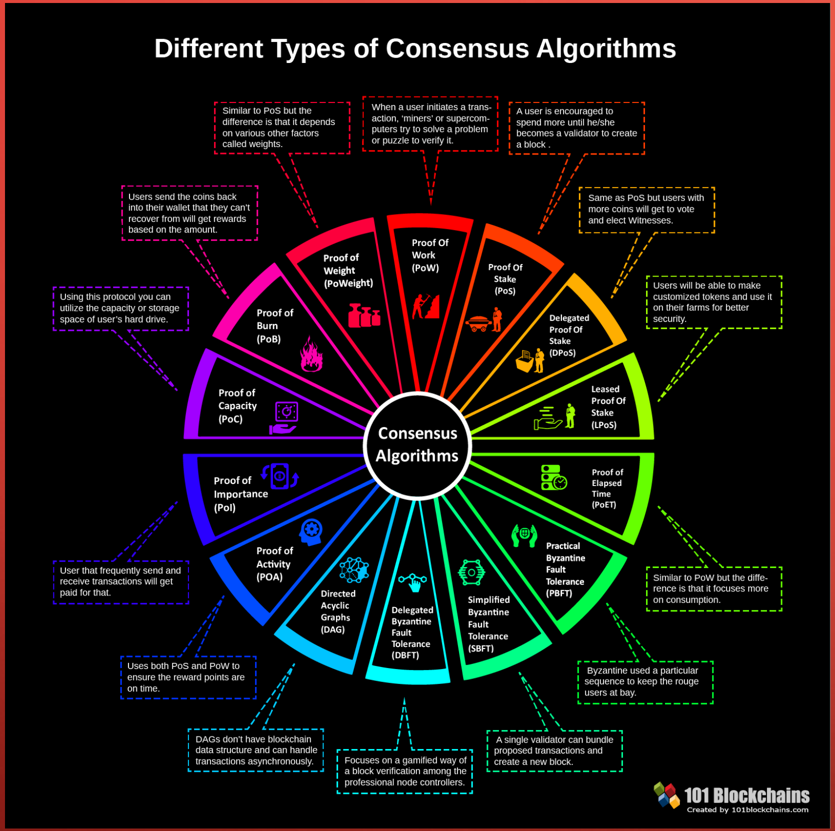
BITCOIN FACTS

- **1998**: B-Money and BitGold: Wei Dai creates B-Money, an “anonymous, distributed electronic cash system.” Nick Szabo proposes a similar idea: Bit Gold, which lay the groundwork for blockchain.
- **October 2009**: Bitcoin whitepaper
- **January 2009**: Genesis Block 0 mined by Nakamoto. Worth 50 BTC that can’t be spent. A Bitcoin is worth 0,001 USD at that time
- **May 2010:** Bitcoin Pizza Laszlo Hanyecz paid 10,000 BTC for two Papa John’s pizzas
- **July 2010:** In 5 days, the price grew 900%, rising from $0.008 to $0.08
- **August 2010:** a protocol failures allows the generation of 92 Billons BTC. The blockchain is reversed to previous state.
- **December 2018**: BTC is worth almost $20k
- **Today:** BTC is worth $4k

F.A.Q

- Why fees ?
  Because you want miners to validate your transactions quickly. Typically 0.00001BTC.
- Why limited Bitcoins?
  Because limited solution to the math problem and this is necessary for a currency.
- Why volatility?
  The total value of bitcoins in circulation and the number of businesses are still small compared to what they could be. Small events, trades, or business activities can significantly affect the price.
- Is Bitcoin anonymous?
  It’s pseudonymous as the adress is shared on the blockchain but the adress can be different for every transaction.
- What if I didn’t mine on the longest chain?
  If the block doesn't end up in the main chain, the miner effectively doesn't get the reward: Attempts to spend it will not be accepted as valid payment by others. The transactions go back to the pool and have to be re-validated.

MORE TO BE DISCUSSED NEXT…



Different Types of Consensus Algorithms

Consensus Algorithms
Proof Of Work (PoW)
Proof Of Stake (PoS)
Delegated Proof Of Stake (DPoS)
Leased Proof Of Stake (LPoS)
Proof of Elapsed Time (PoET)
Practical Byzantine Fault Tolerance (PBFT)
Simplified Byzantine Fault Tolerance (SBFT)
Delegated Byzantine Fault Tolerance (DBFT)
Directed Acyclic Graphs (DAG)
Proof of Activity (POA)
Proof of Importance (PoI)
Proof of Capacity (PoC)
Proof of Burn (PoB)
Proof of Weight (PoWeight)

When a user initiates a transaction, 'miners' or supercomputers try to solve a problem or puzzle to verify it.

A user is encouraged to spend more until he/she becomes a validator to create a block .

Same as PoS but users with more coins will get to vote and elect Witnesses.

Users will be able to make customized tokens and use it on their farms for better security.

Similar to PoW but the difference is that it focuses more on consumption.

Byzantine used a particular sequence to keep the rouge users at bay.

A single validator can bundle proposed transactions and create a new block.

Focuses on a gamified way of a block verification among the professional node controllers.

DAGs don't have blockchain data structure and can handle transactions asynchronously.

Uses both PoS and PoW to ensure the reward points are on time.

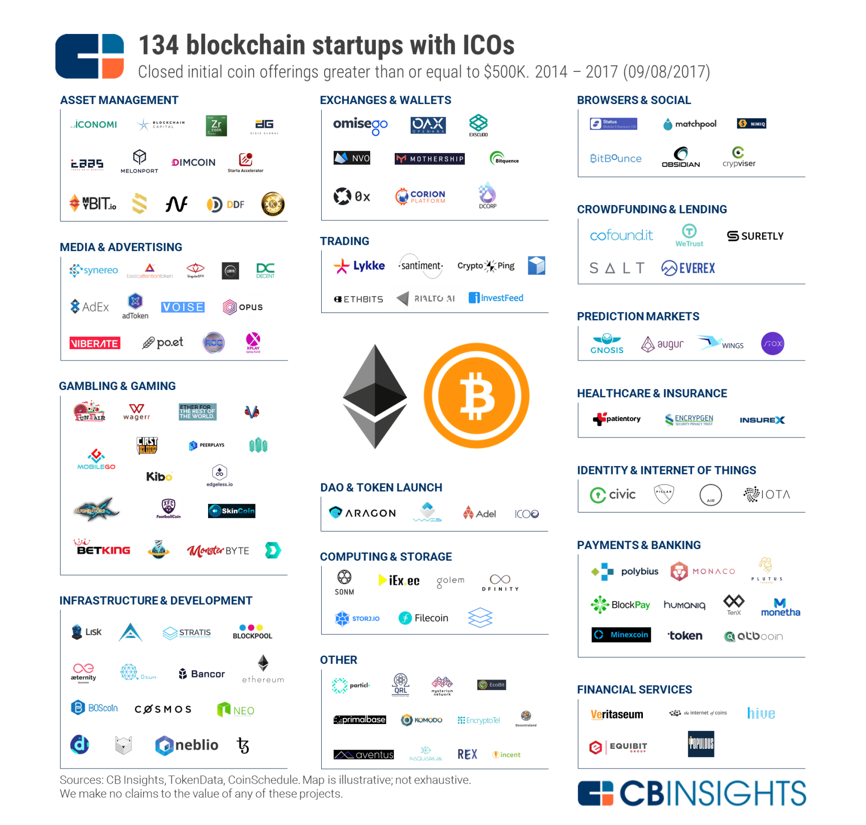
User that frequently send and receive transactions will get paid for that.

Using this protocol you can utilize the capacity or storage space of user's hard drive.

Users send the coins back into their wallet that they can't recover from will get rewards based on the amount.

Similar to PoS but the difference is that it depends on various other factors called weights.

101 Blockchains
Created by 101blockchains.com



134 blockchain startups with ICOs

Closed initial coin offerings greater than or equal to $500K. 2014 – 2017 (09/08/2017)

ASSET MANAGEMENT

MEDIA & ADVERTISING

GAMBLING & GAMING

INFRASTRUCTURE & DEVELOPMENT

EXCHANGES & WALLETS

TRADING

DAO & TOKEN LAUNCH

COMPUTING & STORAGE

OTHER

BROWSERS & SOCIAL

CROWDFUNDING & LENDING

PREDICTION MARKETS

HEALTHCARE & INSURANCE

IDENTITY & INTERNET OF THINGS

PAYMENTS & BANKING

FINANCIAL SERVICES

Sources: CB Insights, TokenData, CoinSchedule. Map is illustrative; not exhaustive.
We make no claims to the value of any of these projects.

CBINSIGHTS