Blockchain A-Z:
Learn how to build your first blockchain

INSTRUCTORS: HADELIN DE PONTEVES & KIRILL EREMENKO

SuperDataScience Podcast - #139

The Rise of Blockchain: a disruptive super-technology much more than bitcoin

- Why learn blockchain?

1st: it will be all around the world, so if you don't know blockchain, you won't understand the world around you.

2nd: the amount of opportunities you have on "blockchain world", you can apply to industries, logistics, start a business, food, health, finance, there's a lot to do.

Blockchain is the new wild west, it's like the internet in 1993 (a lot going on, not 100% set for sure, some moves are good and some are not). You don't need mathematics, just some good sense.

- What's blockchain? What's its definition?

Blockchain was invented to make db transactions securier, so no one can hack it.

The most famous example is bitcoin: the transaction is between a sender and a receiver, it is stored into some blocks that are added one after the other, and you cannot change the history of the transaction. It works through a cryptographic path (the key for blockchain).

Cryptographic hash = fingerprint of a collection of data (it's always 64 hexadecimal digits long). You can take your name and transform it into a hash:

> Ana Beatriz = 416e61204265617472697a
>
> https://www.online-toolz.com/tools/text-hex-convertor.php

Any collection of data has its own fingerprint, which is the hash (it's almost unique).

- How does blockchain work?

Blockchain works as linked blocks, where each block has its own hash and references the previous block. If you change anything in any block, all the next blocks will change it too, because of the "links". So, the deeper in the chain you try to modify, the more consequences are and the more difficult it is to change.

Hack a chain is basically impossible, because you'd need to hack at least half of the chains at the same time, because you have a distributed situation where every computer participating on the chain has one copy of the chain (for example: if you have 100 computers on a chain, you'd have to hack and change the chain of 50 computers at the same time). That's why it's so secure - the more participants, the harder (everybody has a copy and it's always synced). Blockchain is a distributed decentralized ledger *(registro distribuído e descentralizado)*.

- The blockchain components:

Every block has 5 components: block number, data (transactional or any other) and the hash of the previous block. The 4th component is all the three together put into the hashing function, that is: the hash of the current block. The 5th component is "the nonce", which are numbers used only once.

Mining is about calculating this 4th component: the hash. A miner can change the nonce and by it, change all the content of the block. The goal of mining is to find a nonce that will generate a hash, that starts with an amount of zeros (the leading zeros, the more zeros, the more difficult to mine it). This is done to create a cryptographic hash to add this block to the network, so it's kinda a race where the miner which has more computational power will have the power to add the next block. To become a miner, you need several powerful computers, just dedicated to mining.

- Dis-intermediate:

The principle of blockchain is that you remove the intermediary in the process, you don't need anyone between the sender and the receiver. In an example of purchase, the site that

“sells” a product takes a fee because it is intermediating the purchase process, so if the buyer doesn’t receive the product he bought, the site is there to deal with that. With blockchain, you replace this site by a smart contract (kinda an escrow thing). So the money goes into an escrow and it will only be released for the seller after the buyer receives its purchase.

- Web 2.0 & 3.0:

The difference between web 2.0 and 3.0 is that in web 2.0, you have centralized servers with all the information they collect, so they can have control over everything and make money from your activities. In web 3.0, through the blockchain technology, everything will be linked up, so there’s no company holding this information: if it’s hosted in blockchain, it belongs to people.

Part 1 - Blockchain

1.1 - Blockchain Intuition

Plan of attack

Blockchain is not as complex as I.A., but its components have varying complexities, so putting these components in the right order is kinda hard.

What is a Blockchain

Understanding SHA256 - Hash

Immutable Ledger

Distributed P2P Network

How Mining Works - Pt. 1 + Pt. 2

Byzantine Fault Tolerance

Consensus Protocol: Defense Against Attackers and Proof-of-Work

What is a Blockchain

Every block has: data, previous hash and its own hash.

The genesis block: [...], 00000000, 034DFA357

The 2nd block: [...], 034DFA357, 4D56E1F05

The 3rd block: [...], 4D56E1F05, 7364AEB2F

Understanding SHA256 - Hash

A fingerprint is an identifier of a person. A hash is an identifier of a block.

Hash was created by the NSA. It's always 64 digits long (numbers 0-9 and letters A, B, C, D, E, F). It's always 4 bits (4^4). You can put any information in it (photo, text, video, a whole operational system).

https://tools.superdatascience.com/blockchain/hash

5 Requirements for Hash algorithms:

1. One-Way: you can not restore the document by its hash;
2. Deterministic: the result is always the same;
3. Fast-Computation;
4. Avalanche Effect: any change in the document, even a little one as a "+1" or a dot, will change the whole hash and the posteriors;
5. Must withstand collisions: naturally has more documents than hash combinations.

Immutable Ledger

A tradicional ledger can be changed anytime, by anyone.

The immutable ledger implies that any change in any block will change all the blocks after this one, because they're all linked in the chain. So it's impossible to hack and change it after other transactions.

Distributed P2P Network

The blockchain has a copy in all the computers involved in the transaction. When you add one block at the end of the chain in one computer, it will duplicate to the rest.

If someone tries to hack one block, all the following blocks will “break” because it will change the hashes. But the person can hack this block and the following ones, so in one concentrated network the hacker will succeed. However, in a distributed network where the computers sync all the time and check if the blockchains match up, the original hash is copied to the unsettled chain if they don't match each other.

The hacker would have to attack at least half of the blockchains at the same time to be successful in his attack - and in less than 1 minute.

How Mining Works - Pt. 1 + Pt. 2

Mining is about the nonce (a number used only once). If you change the nonce, it will change the whole block hash.

Block + nonce + data + previous hash = actual hash.

A hash is a number. It uses 0-9 and A-F, where A=10, B=11, ..., F = 15. So you can convert it into a hexadecimal number.

When you express a target for the hash (from smaller to largest), it determines the amount of leading zeros. The less zeros, the larger the hash.

To reach the target, the miner changes the nonce until he finds one that satisfies the target (the golden nonce), so the block is accepted in the blockchain. The “avalanche effect” is important so the miner can’t predict a nonce, because it has no logic in it, it’s always variable. And the miner can’t do the “reverse engineering” too, predict a hash by a nonce.

Basically, the miner keeps iterating the nonce ‘till he finds the golden nonce.

Byzantine Fault Tolerance

The Byzantine generals problem is a math problem where 4 generals are trying to attack a castle. 1 general is the “main general” and 1 or more generals are a “corrupt general”; the

main general can be corrupt too. The goal is to attack at the same time and conquer the castle, however this decision has to be agreed by the majority of the generals. To solve this problem, the amount of corrupt generals can't be more than ⅓ of the total.

Consensus Protocol: Defense Against Attackers and Proof-of-Work

It has to solve 2 main problems:

1. Protect network from *Attackers* - what happens when a hacker tries to put a block after all the blocks?
    a. Before a block is added to all the computers involved in the transaction, every single node will conduct a series of checks, very strictly and detailed.
    b. If the network accepts the block that's being added, after the check, then all the computers will have it added to its chain - cryptographic puzzles: hard to solve, easy to verify.
2. Competing Chains - what happens when two blocks are added at the same time in different computers?
    a. Similar to Byzantine Fault Tolerance, but it doesn't decide by majority, it waits for the next block to be added and compares which of the chains that are competing is longer - the longest chain wins.
    b. By probability, the chain that is in more than 50% + 1 of the network will have another block added and will "dominate" the other chain that is competing.
    c. The block that has "lost" is now a orphaned block, all the transactions are lost and there's no more use of it - that's why it's important to wait after a block is added, to guarantee that it isn't competing with another block.

Consensus Protocols:

- Proof-of-Work (PoW): the hash generated for the block is the proof of work.
- Proof-of-Stake (PoS)

Part 2 - Cryptocurrency

2.1 a - Cryptocurrency Intuition

Plan of attack

What is bitcoin

Bitcoin's Monetary Policy

Understanding Mining Difficulty

Virtual Tour of a Bitcoin Mine

Mining Pools

Nonce Range

How Miners Pick Transactions

CPUs x GPUs x ASICs

How do Mempools Work

Orphaned Blocks

51% Attack

Bits to Target Conversion

What is bitcoin

Terms: Blockchain x Bitcoin x Protocols x Coins x Tokens

Layers:

- Technology: blockchain
- Protocols/Coins: bitcoin - rules that guide how participants over the internet communicate with each other; ex: dictates how should they come to consensus,

how public it is, how to use assignatures for authentication, how they agree on updates;
    - Others protocols: ethereum, waves, neo, ripple - each one has its own coin (one protocol has one coin)
- Token: facilitates the creation of smart contracts - bitcoin doesn't have a token, ripple doesn't have too, but waves, neo and ethereum all have tokens

Bitcoin was invented in 2008 by "Satoshi Nakamoto" (was a person? a group?). Basically, it's a protocol.

Bitcoin is about taking the blockchain theory and putting it into practice. Is about creating a protocol that helps people transact (exchange value).

The Bitcoin Ecosystem:

- Nodes - stands for devices that people use but not mining the internet, just people who wants to transact to each other
- Miners
- Large Mines - lots of power, devices, equipment
- Mining Pools

Bitcoin's Monetary Policy

It consists of 2 parts: the halving (the main one) and block frequency.

It's controlled by the internet.

The Halving: it basically controls the quantity of bitcoins released. As the time passes, more people will adopt bitcoin and therefore more transactions will be required to be processed and miners will have more demand (because of the quantity of transactions).

Block Frequency: how often the blocks come in - depends on the cryptocurrency (bitcoin 10min, ethereum 15seg, ripple 3.5seg, litecoin 2.5min).

Article 1: https://bombthrower.com/articles/this-time-is-different-part-i-what-bitcoin-isnt/

What bitcoin isn't

- Backed by nothing
- A ponzi (aka esquema de pirâmide)
- Tulipmania (“primeira bolha especulativa”)

Article 2: hackernoon.com/this-time-is-different-part-2-what-bitcoin-really-is-ae58c69b3bf0

What bitcoin really is

- Inelastic (there’s a set amount of coins that will ever be created)
- Deflationary
- Transparent (blockchain + open source)
- Antifragile (gets stronger when it encounters volatility)

Understanding Mining Difficulty

- What is the Current Target and how does that feel? - If you’re a miner, how would it feel to have that difficulty imposed upon you? What would that imply? > A different perspective from the logical thing;
- How is “Mining Difficulty” calculated?

What is the Current Target and how does that feel:

The target is the leading zeros. If it’s only one zero, the pool is reduced to 1/16 of its total. If it’s two zeros, the pool is reduced to 1/16 of 1/16 of the total. And so on.

Example: if a hash has 5 digits (XXXXX), you have from 0 to 99.999 (100.000 options). If it has 5 digits, but the first one is 0 (0XXXX), you have from 0 to 9.999 (10.000 options). It is reducing the pool size by ten. The hexadecimal is reduced by sixteen because every digit can have 16 values (A-F, 0-9).

Example:

Current target = 18 zeros + 5d97dc + 40 zeros (64 digits)

Probability:

Total possible 64-digit hexadecimal numbers: $16 \times 16 \times \ldots \times 16 = 16^{64} = 1{,}1579 \times 10^{77}$

Total valid hashes (with 18 leading zeros): $16^{64 - 18} = 2{,}4519 \times 10^{55}$

Randomly pick a valid hash: $2 \times 10^{55} / 1 \times 10^{77} = 2 \times 10^{-22}$

= 0.00000000000000000002% - that is the probability to pick a nonce that is the golden nonce.

How is "Mining Difficulty" calculated and is it adjusted:

The difficulty equals the current target/ max target:

The current target is the same as the first question.

The max target is 8 zeros + FFFF + 52 zeros - it is the target that was in place at the very inception on bitcoin, they started out with this target. Although it's not the max hash (it would be if all the numbers were 'F', a 64-F sequence), they started with it because if it already started with 'F', then every time a miner picked an odds, it would be a golden lauch.

The difficulty is adjusted every 2016 blocks, or 2 weeks, as every block should be released every 10 minutes.

Basically, the difficulty shows how much harder it is to mine a bitcoin now, compared to what it was at the beginning.

Virtual Tour of a Bitcoin Mine

Seeing inside a real bitcoin mine.

Mining Pools

In a bitcoin network there are nodes that are simple computers and nodes that are industrial mines with tons of employees with huge rigs. The chances that a simple computer solves the cryptographic puzzle before these industrial mines are like 1 million to 1 trillion, so it's almost impossible to find the golden lawns before them.

However, the computers that work "solo" aren't working in vain. These miners aren't mining for themselves, they combine their hashing power (their processing) into a mining pool, so they have more power together.

The mining pool provides a service where the mining pool distributes the work among the miners, so they aren't doing double work, they aren't solving the exact same block. The mining pool distributes the cryptographic puzzle by bounding the golden nonce to each node, so one computer will look for values from 1 billion to 2 billion, another one will look from 2 billion to 3 billion, and so on. As soon as one of them finds the golden nonce, this mining pool wins the reward and the fees for that block. The reward is split proportionally to the hashing power that they introduced into the mining pool.

The industrial mines can participate in these mining pools, and if they find the golden nonce before another mining pool, the reward is still splitted according to the hashing power.

Another thing the mining pools do is that they remove headaches from individual miners. There is a machine sold in ebay, a cryptocurrency gpu mining rig, that are some mining pools that you buy, download the software, choose one pool and they do all the work for the miner, they solve the cryptographic puzzle without the miner know how.

Article 1: **Bitcoin Mining Pools** https://www.buybitcoinworldwide.com/mining/pools/

What is a mining pool?

Mining pools are groups of cooperating miners who agree to share block rewards in proportion to their contributed mining hash power. Miners can choose to redirect their hashing power to a different mining pool at any time. Find a good mining pool is important to save money.

Pool Concentration in China:

The most mining pools are in China (65%), followed by Malta (10%), USA (3%), Czech Republic (3%) and Japan (0,5%). These numbers represent the hashing power each pool has.

Bitcoin Wallets:

Before joining a mining pool, a miner needs a bitcoin wallet, because all mining pools will ask for a bitcoin address that will be used to send the mining rewards and payouts.

Get Mining Hardware:

Mining hardware is a specialized computer, created for the purpose of mining bitcoin. The more powerful the hardware is, and the more energy efficient it is, the more profitable it will be to mine bitcoins.

Examples: AntminerS19 (Hash power: 95 TH/s), AntminerS19 Pro (Hash power: 110 TH/s), WhatsMiner M30S+ (Hash power: 100 TH/s).

Get a Bitcoin Wallet and Mining Software:

Before joining a mining pool, it will also need Bitcoin mining software and a Bitcoin wallet. It will also need an ASIC miner, since GPU mining will likely never be profitable again going forward.

Mining Pools vs Cloud Mining:

Mining pools are for people who have mining hardware to split profits. Cloud mining is where you pay a service provider to mine for you and you get the rewards.

Why miners are important:

Bitcoin miners are crucial to Bitcoin and its security. Without miners, Bitcoin would be vulnerable and easy to attack. Although most Bitcoin users don't mine, miners are responsible for the creation of all new bitcoins and a fascinating part of the Bitcoin ecosystem. Mining, once done on the average home computer, is now mostly done in large, specialized warehouses with massive amounts of mining hardware, where they usually direct their hashing power towards mining pools.

When you become a member of a mining pool, there are a number of ways your rewards for contributing hashing power can be calculated. All of the payout methods use the term "share". A "share" is awarded to members of the mining pool who present a valid partial proof-of-work. Essentially, the more hashing power you contribute to the pool, the more shares you are entitled to.

Pay Per Share

Is a payout scheme, the most simple one, where it guarantees the miner a payout regardless of if the pool finds the next block or not. The value of a share is determined by the amount of hashing power that is needed to find a block divided by the reward of finding it.

Full Pay Per Share/ Pay Per Share +

Is a payout scheme, similar to the Pay Per Share, except transaction fees are also paid to the pool members on top of the block reward.

Pay Per Last N Shares

Is a payout scheme that shifts more risk to pool members, but also more rewards. The pool members are only paid once a block has been found, so when the block is found, the pool looks at your share contributions for all previous blocks where the pool didn't find the block (this is a time window, and all the blocks in a time window are known as a round). Using these numbers, the pool determines your total share contributions over the round to determine your payout.

The idea behind this payout scheme is that it removes all luck and only pays members based on their contribution to actual revenue earned by the pool. This scheme also incentivizes members to continue mining in the pool even as the profitability of mining different coins rises comparatively. This is because disconnecting from the pool before a block is found will pay you nothing.

Article 2: **Bitcoin mining and energy consumption**
https://blog.bitcoin.org.hk/bitcoin-mining-and-energy-consumption-4526d4b56186

How much electricity does Bitcoin consume?

In some places, Bitcoin mining is unregulated and in many jurisdictions even illegal, so there's no hard data on how much total electricity it uses. In other places, where Bitcoin is regulated and legal, power companies don't necessarily know the purpose of the used electricity, and their statistics don't account for cryptocurrency mining.

However, it is possible to estimate the power by looking at Bitcoin's difficulty (the number calculated by the Bitcoin protocol, a measure of how many hashes it takes for a miner to find a valid block on average). As this is the currently most energy efficient miner on the market, it allows us to calculate a lower boundary for how much electricity is consumed.

Dividing 1.14x10^19 by 14x10^12, we can calculate that there are a maximum number of 800,000 S9 miners currently in operation, consuming roughly 1,100 MW in total. The statistics provided by the International Energy Agency do not use MW or GW. Instead, they use "Mtoe", or "Million tons of oil equivalent". 1 toe is 11.63 MWh. The total energy estimated to be used globally in 2017 is 13,647 Mtoe, or 158,714,610 GWh.

This is a lower boundary, but it is a relatively good estimate. Some miners currently active on the Bitcoin network might be older and less efficient, but the second most efficient Bitcoin miner generation consumes 1.5 times as much energy per hash as the S9. This lower boundary also serves as a natural equilibrium. If all other variables (as price and technology) remain constant, Bitcoin's energy consumption will converge to this boundary.

Why does that seem so small?

There are many ways to make that number look big or small in comparison. It looks big if compared to:

- Bitcoin uses as much energy as 520,000 Canadians every day;
- Bitcoin uses as much energy as the Democratic Republic of Congo;
- Bitcoin uses more energy than 116 countries each;
- Bitcoin uses enough energy to power 6 Nimitz-class aircraft carriers.

But it looks small if compared to:

- The energy that Bitcoin consumes in a year would only last the U.S. for 19 hours;
- Bitcoin uses only 20% of the energy from a single coal power plant in Taiwan;
- The Three Gorges Damn in China produces three times as much electricity as Bitcoin consumes;
- The U.S. produces more electricity from a single Geothermal plant than Bitcoin requires;

- 17 NSA Data centers together consume more electricity than Bitcoin;
- Google used about double as much electricity in 2015 than Bitcoin does today.

What does the future of Bitcoin mining look like?

Bitcoin currently consumes mostly very cheap electricity. Miners race to the bottom of who can find the cheapest electricity, and everyone consuming electricity significantly larger than the average is forced to shut down their unprofitable operations. As a result, Bitcoin mostly consumes electricity in places where it is abundant, and cannot be stored or transported. Because oil, gas and coal are often trivial to transport, you very rarely find Bitcoin mining operations that consume these resources, because it would be more profitable to ship the energy to a place where it can be sold for more.

Bitcoin will continue to seek those cheap and otherwise unused forms of electricity, while it will probably never be profitable to mine in urban or industrial centers. You are willing to pay more for your air-conditioning or water heating than a Bitcoin miner can afford.

Where will that electricity come from?

Bitcoin will continue to unlock largely inaccessible electricity in the form of hydro, geothermal and solar. Few Bitcoin mines will rely on oil, gas or coal, as these resources are largely exhausted, can easily be transported to areas with higher prices, and are expensive to extract.

Also, one can argue that Bitcoin actually saves energy. The world's financial system requires many resources beyond the electricity to run servers. Banks house themselves in tall buildings with air-conditioning, private jet companies fly gold and cash around the world for discreet clients, while printing cash requires cotton farming or even the slaughter of animals.

Nonce Range

Nonce is a field in the block which allows miners to participate in the cryptographic puzzle challenge. Miners can't change any other field except the nonce one, and by changing the nonce, they change the hash and hope to get one below the target. As soon as this happens, they win the puzzle and add the block to the chain.

The nonce isn't infinite, the miners can't increment the nonce forever. It's a 32 bit number, so there's only 32 bits of memory allocated in every single block for the nodes. The nonce is an unsigned integer which has a range and is between zero and four billion.

Probability that a random number picked is valid: $2 \times 10^{55} / 10^{77} = 2 \times 10^{-22}$ = 0,00000000000000000000002%

The nonce is a 32-bit number, the max nonce = $2^{32} = 4 \times 10^{9}$

Assuming no collision, this means that there are 4×10^{9} different hashes. The probability that one of them will be valid = $4 \times 10^{9} \times 2 \times 10^{-22} = 8 \times 10^{-13}$ = 0,0000000000008%

The conclusion is that one nonce range is not enough, even if the miner goes through the whole nodes, it's still unlikely to find a golden node.

A modest miner does 100MH/s (100 Million Hashes per second). So 4 billion hashes, they would solve in 40 seconds, more or less. If the miner doesn't find the nonce it's done? Not really. The block has a field called timestamp, that's a timestamp of when this block is being mined and the timestamp updates every single second.

The timestamp creates a loop hole, a solution to the problem, meaning that every single second the timestamp and the block will be updated because the timestamp will increase by one second. This update causes the avalanche effect and changes the hash. So even if the miner tried 100MH in one second, when the timestamp changes, these 100MH are valid again.

What about the mining pool? This timestamp solution works for one miner, but it's not the solution for the mining pool, as they will be done in less than 1 second and will have to wait till the timestamp changes.

How Miners Pick Transactions

How does a miner get the list of transactions that actually goes into the field "data"? The transactions come from the mempool, or memory pool, which is attached to every node and are basically all the unconfirmed transactions that get stored before they get included inside the block. Blocks are created every 10 minutes but transactions happen all the time.

So the miner will need to include some of these transactions in the block. In the real blockchain it's about 2000 transactions, but in this example it will set a limit of 5 transactions. And how does the miner pick these transactions?

Every transaction has the transaction ID and the fees the miner will get including each one. The fees are non-compulsory and specified by the users themselves. As the miners will get the fees, they will choose to include in the block the transactions with the highest fees.

Knowing that the miners can choose which transactions are included in the block, the problem with the mining pool is solved. If the pool tries all the hashes in less than 1 second, then they can change the transactions that are being included in the block, and as the data changes, the hash changes too.

Article 1: **How does bitcoin mining works**
https://www.coindesk.com/learn/how-bitcoin-mining-works-2/

Key takeaways

- Bitcoin mining is the process of discovering new blocks, verifying transactions and adding them to the bitcoin blockchain;
- Each time a new block is discovered, the successful miner is granted the right to fill that block with a new transaction data;
- In return for dedicating time and resources to performing this task, winning miners receive a free amount of newly minted bitcoin known as a "block reward" as well as any fees attached to transactions they store in the new blocks;
- The process of giving successful miners newly minted bitcoin is exclusively how new coins enter circulation.

Why mine bitcoin?

To stand a chance of earning bitcoin block reward and to participate in securing and maintaining the decentralized bitcoin network.

How do bitcoin miners discover new blocks?

In order to validate and add new transactions to the blockchain, miners must compete with each other using specialized computing equipment, to generate hashes. In order to discover the next block, miners must generate a hash smaller than the target hash.

As a starting point, all miners take the data from the previous block, known as the "block header"– which contains things like a timestamp of the block, the hash of the previous block data, and an empty space known as a "cryptographic nonce." Most of the data in the block header is fixed, meaning it cannot be changed, apart from the nonce, and just changing a single bit of the input produces a totally different hash.

The tricky part is, hashes are generated completely at random, meaning it's impossible for miners to know what the hashes will be before they generate them. So it's simply a case of trial and error until someone finds the right nonce value – known as the "golden nonce."

What is a hash?

A hash is a cryptographic mathematical function that converts any message or data input into a fixed-length code. The outputs have set lengths to make it impossible to guess the size of the input. These hash functions are irreversible, meaning that it's impossible to revert the hash back to its original input. The same input will also always generate the same sequence of letters and numbers. In the case of Bitcoin, the blockchain uses Secure Hash Algorithm 256 or SHA 256 to generate a 256 bit or 64 characters long output, regardless of the size of the input.

Bitcoin mining difficulty

It should take approximately 10 minutes for a miner to successfully create the winning code to discover the next block. The Bitcoin protocol has the ability to automatically increase or decrease the complexity of the mining process depending on how quickly or slowly blocks are being found. Every two weeks, the Bitcoin protocol automatically adjusts the target hash to make it harder or easier for miners to find blocks.

Why does bitcoin mining use so much energy?

At press time, Bitcoin's hash rate – the measure of all computational power dedicated to mining new coins – stands at 183 exahash (Eh/s.) This means bitcoin miners collectively

attempt to crack the target hash of the next new block 183 quintillion times per second. According to the Cambridge Bitcoin Electricity Consumption Index (CBECI,) this activity consumes approximately 131 TeraWatt hours (TWh) of electricity per year.

The main reason for this extreme consumption is because each time bitcoin rises in price, it encourages new miners to join in the battle to win new coins and forces existing outfits to purchase more rigs or upgrade their equipment to remain competitive. When this happens, the amount of computational power used to mine bitcoin increases and causes the bitcoin protocol to ramp up the difficulty.

CPU's vs GPU's vs ASIC's

The different hardwares used to solve the cryptographic puzzle.

CPU (Central Processing Unit): executes all the computer commands, like a center of operations, it's very versatile to take charge of different things, it's general (not specialized to one specific thing), has a lot of power and can do very sophisticated things. Can solve the SHA256 hash, can calculate it faster than a human but it's limited at about 10 million hashes per second (<10MH/s).

GPU (Graphics Processing Unit): basically the computer graphic card, it's specialized, designed to work with graphics and the operations specialized for the matrix operations (operations required in order for videos to work, for graphics to appear on screens, etc). Can calculate hashes and as it has less components and functionalities, it works faster than the CPU, solving 1 billion hashes per second (<1GH/s).

ASIC (Application-Specific Integrated Circuit): are totally and utterly specialized for one thing only: to calculate the SHA256 hash. The integrated circuit of the device is designed in a way that the device doesn't have to perform any operations: as the electricity runs through the device, the calculation is performed at a physical level and solves over 1 trillion hashes per second (>1000GH/s).

Cloud Mining: miners rent the equipment and pay a fee in order for that equipment to participate in mining (kinda like cloud computing/storage).

Article 1: **Ethereum's Memory Hardness Explained**

https://www.vijaypradeep.com/blog/2017-04-28-ethereums-memory-hardness-explained

As crypto-currencies increase in value, so does the payout from mining them. This creates a substantial economic incentive to not only deploy more mining hardware, but to also develop faster, more efficient mining hardware.

How the Ethereum Ethash Hashing Algorithm Works

The Ethash algorithm relies on a pseudorandom dataset, initialized by the current blockchain length (DAG) and regenerated every 30000 blocks or 5 days. The flow of the ethash hashing algorithm can be summarized as follows:

1. The Preprocessed Header (derived from the latest block) and the Current Nonce (the current guess), are combined using a SHA3-like algorithm to create our initial 128 byte mix, called Mix 0 here;
2. The Mix is used to compute which 128 byte page from the DAG to retrieve, represented by the Get DAG Page block;
3. The Mix is combined with the retrieved DAG page. This is done using a ethereum-specific mixing function to generate the next mix, called Mix 1 here;
4. Steps 2 & 3 are repeated 64 times, finally yielding Mix 64;
5. Mix 64 is post processed, yielding a shorter, 32 byte Mix Digest;
6. Mix Digest is compared against the predefined 32 byte Target Threshold. If Mix Digest is less than or equal to Target Threshold, then the Current Nonce is considered successful, and will be broadcast to the ethereum network. Otherwise, Current Nonce is considered invalid, and the algorithm is rerun with a different nonce (either by incrementing the current nonce, or picking a new one at random).

Why Is This Memory Hard?

Every mixing operation requires a 128 byte read from the DAG. Hashing a single nonce requires 64 mixes, resulting in 128 Bytes x 64 = 8 KB of memory read. The reads are random access, so putting a small chunk of the DAG in a cache isn't going to help much, since the next DAG fetch will very likely yield a cache miss. Since fetching the DAG pages from memory is much slower than the mixing computation, we'll see almost no

performance improvement from speeding up the mixing computation. The best way to speed up the ethash hashing algorithm is to speed up the 128 byte DAG page fetches from memory.

Conclusion

The sequential, DAG page fetches in the ethash hashing mining algorithm hits the memory bandwidth limits of modern day hardware, limiting their theoretical maximum hashrate. Will we be seeing custom ethereum miners? Maybe. But when this happens, they probably won't be ASIC or FPGA based. They'll likely still be based on off-the-shelf chips (mobile GPUs or VPUs), and not in the traditional graphics card form factor we're so used to seeing in modern computers.

How do Mempools Work

A peer-to-peer network is composed by nodes, that can be people wanting to do some transactions or miners. Each node has its own mempool, its own staging area for transactions.

Imagine a person A wants to do a transaction, it will be added to her mempool, then it will be broadcasted or relayed across the network (to the closests nodes) and then the transaction will be added to theirs mempools. This process goes on until all the nodes on the network have this transaction. Then, person B wants to do a transaction too, that will be added to her mempool and all the mempool's nodes on the network. Then, a miner C wants to do a transaction, usually they just mine, but this one wants to do a transaction and it will be added to his mempool and all the others on the network. This goes on as the mempools get filled up (> 8000 a day).

When the miner mines the block, he chooses the 2000 transactions that will be on that block, and if he finds the golden nonce, he mines the block, these selected transactions are removed from his mempool and the block is added to the blockchain. This blockchain is relayed across the network to the closests nodes and the transactions that are in this block are removed from the mempools that received the block. This process goes on until all the nodes on the network have this block and have removed the transactions.

Article 1: **An in-depth guide into how the mempool works**
https://blog.kaiko.com/an-in-depth-guide-into-how-the-mempool-works-c758b781c608

To submit a transaction to the miners, nodes have to relay it to each other until it has propagated across the entire network. The mempool is the node's holding area for all the pending transactions. It is the node's collection of all the unconfirmed transactions it has already seen enabling it to decide whether or not to relay a new transaction.

As the Bitcoin network is distributed, not all nodes receive the same transactions at the same time so some nodes store more transactions than others at some time. Plus, everyone can run its own node with the hardware of his choice; so all nodes have a different RAM capacity to store unconfirmed transactions. As a result, each node has its own rendition of the pending transactions, this explains the variety of Mempool sizes & transactions counts found on different sources.

Before letting a transaction into its Mempool, a node has to complete the series of checks. If the transaction matches the criteria, it is allowed into the Mempool and the node starts broadcasting it. If it doesn't, the transaction is not re-broadcast by the node.

When a node receives a new valid block, it removes all the transactions contained in this block from its mempool as well as the transactions that have conflicting inputs.

Unlike mining, there is no financial incentive for running a node. Therefore, the hardware dedicated to it tends to be limited and so a node's Mempool often max out its RAM. When this happen, in former versions of bitcoin, the node would just crash and restart with an empty Mempool. In recent versions of bitcoind (0.12+), if the Mempool size gets too close to the RAM capacity, the node sets up a minimal fee threshold. Transactions with fees per kB lower than this threshold are immediately removed from the Mempool and only new transactions with a fee per kB large enough are not allowed access to the Mempool.

Orphaned Blocks

Orphaned Blocks are valid blocks which are not part of the main chain. They can occur naturally when two miners produce blocks at similar times or they can be caused by an attacker with enough hashing power, attempting to reverse transactions.

At blockchain.com we have a vertically blockchain diagram, where the top is the most recent block and the bottom is the least recent. The dotted arrow indicates that it skipped a lot of blocks that weren't interesting for this specific analysis, that weren't orphaned blocks.

The diagram shows the orphaned blocks and their timestamp, number of transactions, who relayed it and its id. In this example, the block 503948 was added on 12/01/2018 at 23:10:32 by BTC.com and it had 1766 transactions. Before this block, everything was ok, so 503947, 46, 45, 44, etc were added to the blockchain too. After this block, two mempools claimed that they found the block 503949, one at 23:28:07 by GBMiners with 2991 transactions and the other at 23:28:33 by SlushPool with 2874 transactions. This creates two different chains, as the blocks are different too (we know this as the number of transactions aren't the same). Both mempools continue to mine blocks and try to grow their own chain. The nodes close to GBMiners will receive its block by the network and the nodes close to SlushPool will receive its block by the network, causing two competitive chains in the network at the same time.

The winner chain is the longest one, so whoever solves the next block first and adds it to the network, wins the competition. If it's a node that's close to GBMiners, then this is the winner chain. If it's a node that's close to SlushPool, then this is the winner chain.

What happens to the loser chain? The chain stops and the transactions that were in this block go back to the mempool of all the nodes that accept the block. If there is a transaction that was in the loser chain and in the winner chain, the transaction exits the mempool again. If the transaction was only in the loser chain, it goes back to the mempool and stays there until a miner puts it in another block.

By this, we know why we have to wait for a few confirmations before considering the transaction as successful. This means that we have to wait for several blocks on top of the current block - the rule is to wait for six confirmations before considering that the transaction was successful, otherwise it might end up a double spend problem.

The 51% Attack

It's a hypothetical attack that could possibly happen, discussed by blockchain owners and miners because of the consequences it can have.

It’s not an attack designed to tamper with an individual or select blocks, so it’s not about attacking 50% + 1 nodes of the network - that’s impossible, the hacker would have to hack not just miners' computers but also normal people's computers.

It’s about attacking 50% + 1 of the hash rate. A group of attackers goes into the network, get a blockchain, “close” themselves from the rest of the network and keep mining. It’s not a real attack, they don’t break into anybody’s computer, it isn’t even illegal - they just keep mining and don’t broadcast their information with the network.

While the rest of the network mines 2 blocks, this “closed group” mines 5 blocks at the same time. And as the time goes by, they’re always mining more blocks than the rest of the network. Then all of a sudden, this closed group goes back to the network and starts broadcasting their chain to the other nodes, creating two competitive chains. And the rule is that the longest chain wins, so the “real chain” that was in the network is invalidated and the group that was working on it receives the attacker’s chain.

But if the attacker group was really smart, they could have taken advantage of the double spend problem. When they broadcast their chain, all the transactions that were in the blocks that were substituted go back to the mempool, and they stop being valid. So they could’ve gone to the last block accepted in the blockchain and validated a list of transactions, which would become invalid at the end of the day, and buy a lamborghini, trade bitcoin in bitcoin cash, spend a lot of money and so on. And as the network is all about keys, p2p and all those advantages of blockchain, nobody could track them down and find who they really are. And even if the transactions could’ve been accept after on the chain, as the attacker group controls 51% of the hash rate, they wouldn’t let it happen, they could mine empty blocks or just add other transactions except these ones and after a certain time (something like 72 hours), those transactions would be released out of the mempool and return to their owners.

In the orphaned blocks we saw that users normally wait till 6 confirmations to consider the transaction as valid, but this is in a normal context, where no one is trying to attack the network. And in the bitcoin network this is harder to happen, because of the immense hash rate (lots of miners), but in smaller cryptocurrencies it is possible to happen, even if it’s not lucrative financially.

Extra: Bits to Target Conversion

Where is the current target stored? In a field called bits, but not in an explicit way, but as a code that needs to be “converted” to the number of bits.

- Bits: 392009692

The first step is to transform this code into a hexadecimal number.

- Bits in Hex: 175D97DC

The second step is to take the first 2 numbers of the hexadecimal number and transform them into decimal numbers again - this will be the length in bytes.

- 17 -> 23 * 2 = 46 hex digits [0000000000000000000000000000000000000000000000]

The third step is to take the rest of the numbers and substitute the first zeros with them.

- 5D97DC0000000000000000000000000000000000000000

The last step is to complete the 64-digit number with zeros.

- 00000000000000000005D97DC0000000000000000000000000000000000000000

And this is the current target.

2.1 b - Cryptocurrency Transactions Intuition

Plan of attack

Transactions and UTXO's

Where do Transactions Fees Come From

How Wallets Work

Signatures: Private & Public Keys

Signatures & Keys Demo

What is a Segregated Witness (SegWit)

Public Keys vs Bitcoin Address

Hierarchically Deterministic (HD) Wallets

Transactions and UTXO's

UTXO: Unspent Transaction Outputs

In a bank, a transaction is done, it's recorded somewhere and that's it.

In Bitcoin, it's different. A transaction lives on after it's been executed until another transaction builds off. The wallet doesn't have the amount of cryptocurrency as a sum up, it has the transactions that hand the "money".

A wallet wouldn't have something like "1,7 BTC", but something like

- Person1 -> Me : 0,1 BTC
- Person2 -> Me : 0,3 BTC
- Person3 -> Me : 0,6 BTC
- Person4 -> Me : 0,7 BTC

} UTXOs

And if I want to buy something for 0,5 BTC, I have to pick the transactions to have the value. In this case, I would pick the 3rd transaction, as it's the closest to 0,5. And as it can't be no

change, I need to transfer 0,5 BTC to the store where I'm buying the product and 0,1 BTC back to myself. This will change the list of UTXOs I had in my wallet:

- Person1 -> Me : 0,1 BTC
- Person2 -> Me : 0,3 BTC
- Person4 -> Me : 0,7 BTC
- Me -> Me: 0,1 BTC

} UTXOs

Where do Transactions Fees Come From

Now, the wallet has the following transactions:

- Person1 -> Me : 0,1 BTC
- Person2 -> Me : 0,1 BTC
- Person3 -> Me : 0,4 BTC
- Person4 -> Me : 0,3 BTC
- Person5 -> Me : 0,3 BTC

} UTXOs

And I want to buy another bicycle for 0,9 BTC and an apple for 0,02 BTC. The transaction will have the following inputs and outputs:

Inputs:	*Outputs:*
Person3 -> Me : 0,4 BTC	0,9 BTC to the bike shop
Person4 -> Me : 0,3 BTC	0,02 BTC to the fruit shop
Person5 -> Me : 0,3 BTC	0,06 BTC back to me

But the inputs totalize 1 BTC and the outputs totalize 0,98 BTC. So, what isn't explicit as money you take back, is the fee. In this case, the fee equals 0,02 BTC. In theory, bitcoin is free, you don't have to pay any fees, but in reality as there's a competition going on, the higher the fee, the more likely is the transaction being accepted.

So the fees come from the input - output.

How Wallets Work

The wallet analyzes the transactions that ended up coming back to the person and are UTXO and add up these amounts, resulting in the "balance'' of the wallet.

Signatures: Private & Public Keys

In the example above, the inputs were from a person to me, but what if I put another person as she had transferred the bitcoin to me? Well, for this, bitcoin has private and public keys.

When someone starts in cryptocurrency, it is assigned a private key, an unique identifier, that the person shouldn't tell anybody, like a bank password. Then, from a private key it is possible to generate a public key, that the person can actually share with others to send and receive money.

If this person wants to send money to another person, to make it secure, her private key is combined with the message (the transaction) and generates a signature, as like the private key signs the message and only the owner knows it. And to prove that it really is this person who signed the message, the public key is combined with the message and the signature, send to the verification function and it outputs a yes or no answer.

With the public key, it's impossible to do the reverse function and generate the private key. But combined with the message and the signature, it's possible to know if the transaction is valid or not. And all these components are added to the blockchain.

Signatures & Keys Demo

How public and private keys works:

tools.superdatascience.com/blockchain/public-private-keys/keys

What is a Segregated Witness (SegWit)

A block has its id, timestamp, nonce, previous hash, hash and so on, that are called the block header combined, as they're not the transactions, not the actual content, they're just an additional info that comes along. The transactions are the main body of the block.

Bitcoin has a block's size limit of 1 megabyte, probably because of the quantity of transactions in it so it doesn't make the network too slow. But as people noticed the network became slower over time, two solutions were proposed to solve the problem: the

hard fork and the soft fork. The hard fork was to increase the block's size and the soft fork was segregated witness.

Soft fork is like an update on how blockchain works, which is not compulsory for everybody to take on board right away, it can propagate over the network with time and people can accept it slowly, so it's easy on the network in that way.

What segregated witness does is it looks at the contents of an individual transaction (the id, the address from, the address for, the amount, the signature, the public key) and separates the signature and the public key (called scriptSig, which corresponds to 60% of the whole transaction's size) from the message. It's still linked to each block in each transaction but it goes through the network separately, saving some space and decreasing the block's size. This fork causes the feeling that now bitcoin has a bigger size limit, but that's not true, the block's size limit is still 1 megabyte, but as the scriptSig goes separately from the content, you have more space available to include more data.

And it's called segregated witness because witness is used as a synonym for signature. So it's a segregated signature.

Article 1: **Understanding Segwit Block Size**
https://jimmysong.medium.com/understanding-segwit-block-size-fd901b87c9d4

Block Size and Soft Fork

Block size is simply the size, in bytes, of the serialized block. As part of the consensus rules, every node on the Bitcoin network currently checks that a block is less than 1,000,000 bytes. That is, a block that's greater than 1,000,000 bytes will be rejected by these nodes as a consensus rule.

Enter Segregated Witness

The key insight is that a big part of the transaction, the scriptSig (signature, pubkey, etc), can essentially not be sent to Legacy nodes and still be counted as valid. Non-Segwit transactions put the scriptSig in the middle of the transaction. Segwit transactions put the scriptSig at the end. When Segwit transactions are sent to Legacy nodes the witness data is stripped. The key is that these "stripped" transactions are still valid transactions on Legacy

nodes, which gives us a savings over non-Segwit transactions. Thus, more transactions can fit into the block sent to Legacy nodes without going over the 1,000,000 byte limit.

Restricting Segwit Blocks

The creators of Segwit could have let Segwit blocks be as large or as small as they wanted and Segwit would still have been a soft fork, provided the blocks sent to Legacy nodes are still 1,000,000 bytes or under. A 1MB Segwit block restriction would not increase block size at all and a 1GB Segwit block restriction would open up a very obvious attack vector. To restrict Segwit blocks, the creators of Segwit instead came up with a different restriction than size.

The Segwit blocks are restricted by something called Block Weight. Block Weight is a new concept introduced in Segwit, and it's calculated on a per-transaction basis. Each transaction has a "weight" which is defined this way:

(tx size with witness data stripped) * 3 + (tx size)

Non-Segwit transactions have zero witness data, so the weight for a non-Segwit transaction is exactly 4 times the size. Segwit transactions have some witness data so the weight is going to be less than 4 times the size. Note Segwit transactions are transmitted to Legacy nodes without witness data, so this formula will always result in blocks communicated to Legacy Nodes that are less than or equal to 1,000,000 bytes. Again, this is why Segwit is a soft fork.

So How Big Can Segwit Blocks Get?

Blocks received by legacy nodes will be less than 1,000,000 bytes. Blocks received by Segwit nodes, on the other hand, can be bigger, but how much bigger?

It turns out that if you made a pathologically large Segwit transaction, you can make a block with just the coinbase transaction and a pathologically large Segwit transaction that's very close to 4MB. Essentially, this pathologically large Segwit transaction would be mostly Witness Data with the block weight just under 4,000,000. That block would be very close to 4MB, but way under 1,000,000 bytes when stripped of witness data. This is an

extraordinary case and wouldn't be very profitable for a miner unless that transaction also had an extraordinarily high fee.

The normal case without pathologically large/giant fee Segwit transactions results in a block size of around 2MB, which is what the creators of Segwit designed for. When you hear someone say "Segwit is a block size increase", this is what they're referring to. The average Segwit block size will be roughly 2MB, though Legacy nodes will still receive blocks that are 1,000,000 bytes or lower due to stripped witness data.

Conclusion

It's a little weird talking about the block size with Segwit since the size isn't the way blocks are restricted. Segwit blocks are restricted by weight and that's a related, but different calculation. If there are no Segwit transactions in a block, this weight calculation collapses to size, but in the more general case of blocks with Segwit transactions, miner profit does not strictly increase with block size.

Segwit makes block size a less relevant concept. Block Weight is the more accurate and useful metric to judge blocks by, though it certainly has a relationship to size.

Public Keys vs Bitcoin Address

Even though most people use "Public Keys" and "Bitcoin Address" as synonyms, these terms are slightly different.

The public key is derived from the private key and it's used on the verification function to verify if a transaction is valid or not. The bitcoin address is derived from the public key applying the SHA256 hash function, which is deterministic and has low chances of collisions. So the same public key always generates the same address, and it's very unlikely that two different public keys will generate the same address.

The address is where you ask people to send you money and it's sharable, it's public. It's possible to send money to the public key also, the wallets handle it one way or another. And the reason for both exist is to make the public key safe, to not expose it too often, only to send money for running the verification function, not when receiving money.

And even being unable to know the private key by the public key, as the public key is generated by an elliptic function, it is possible that in the future someone discovers how to do the reverse and goes from the public key to the private key. So, sharing the address that is generated by "hashing" the public key, there's another layer of security.

Hierarchically Deterministic (HD) Wallets

Cryptocurrencies were made in a way that no one knows who's behind the private key, it's not anonymously because if someone accesses the private key, they can track back all the transactions that were made, but it's like the private key is an avatar and no one really knows your infos.

So everytime you receive money, you share your address, and everytime you send money, you share your public key. It's possible that someone starts to observe the pattern by your address and public key, they won't know who you are but they'll know that you are receiving money from one person and sending money to another person. And as they observe your pattern, they can discover where you're spending money, the frequency you do it and somehow have an idea of who you are.

And if a network says it's going to have privacy in the network, then it has to have and people shouldn't be able to establish patterns. One solution for that is to create different private keys, that will generate different public keys, that will generate different addresses. And if the person really wants privacy, they can create a private key and so on for each transaction. But as it's really hard to maintain it securely and know all the private keys, bitcoin introduced an improvement called BIP32.

BIP stands for Bitcoin Improvement Proposal, that proposes the hierarchically deterministic wallets. An active wallet gets a master private key, then it gets a private key, then a public key and then an address. And as we know, the first two keys (master priv and private) must be kept in secret and the public key only should be shared when really necessary. So, having this master private key and adding "1" to it, it's generated a new private key, then a public key and then an address, as the hash will change. And if you add "2", "3", "4" and so on, it will generate different keys every time.

It is deterministic, so it is possible to regenerate them at any point in time, just having access to the master private key.

And how is that advantageous?

- For wallets, that are always generating different keys and addresses so it's impossible to track down or observe patterns;
- For organizations, so the director has the master private key and each department has separated private keys, and the good part is that only the master private key gives access to all the transactions done in all the private keys;
- A master public key can be created by the master private key, it doesn't have any private keys in between and it can recreate any of the public keys - it's used by an auditor, for example, that doesn't need to send money, so doesn't need a private key, but they need the access to the public keys to go through the blockchain to understand where funds have been sent.

Article 1: **Deterministic Wallets, Their Advantages and Their Understated Flaws**
https://bitcoinmagazine.com/technical/deterministic-wallets-advantages-flaw-1385450276

Unlike old-style Bitcoin wallets, which generate new Bitcoin addresses and private keys randomly as needed, in a deterministic wallet all of the data is generated using a specific algorithm from a single seed. If you write down the seed to your deterministic wallet, and then after six months your hard drive gets corrupted and the wallet unrecoverable, you can simply create a new wallet using the same seed and all of the addresses and private keys from your old wallet will come back again exactly as they were before.

The latest deterministic wallets have two key properties that are heavily advertised by their developers. The first of these properties is the concept of a "master public key". A master public key is a key that can be generated from the wallet's master private key that has the power to generate all of the addresses in a Bitcoin wallet, but none of the private keys. Thus, someone with access to a master public key can look at the balance of a deterministic wallet, but cannot actually spend the balance because they have no way of generating the private key corresponding to each address. The second property is hierarchy: the private keys that you generate from a master private key are themselves master private keys and can in turn be treated as deterministic wallets in their own right.

Part 3 - Smart Contract

3.1 - Smart Contract Intuition

Plan of attack

What is Ethereum

What is a Smart Contract

Decentralized Applications (Dapps)

Ethereum Virtual Machine & Gas

Decentralized Autonomous Organizations (DAO)

The DAO Attack

Soft and Hard Forks - pt. 1

Soft and Hard Forks - pt. 2 (Advanced Tutorial)

Initial Coin Offerings (ICOs)

ICO Case Study

Blockchain Startups: White Papers

Blockchain & Web 3.0

What is Ethereum

Ethereum is a project created by Vitalik Buterin in 2013, when he was 19yo. He wanted to do a scripting language for bitcoin, but the community didn't agree with it, so he went along and created his own project.

Ethereum and Bitcoin are like friends, as they both exist in the protocol layer and have different purposes: Bitcoin creates a cryptocurrency to disrupt banks, by allowing people to

trade or transfer money borderless and permissionless; Ethereum creates a platform for others to create projects on top of it.

The idea behind the Ethereum Protocol is to interconnect everyone and everything, as the Internet, but using the blockchain technology, that not only allows people to store transactional data but also allows people to store programs that facilitate the execution of other programs, decentralizing any application built in there.

Imagine the Facebook app, where to people use it, they need to go to Meta Server and store their data there; with Ethereum protocol, people would be able to run the application at the same time but without storing their data in a server. This would work as a world's super computer in a distributed manner, it won't be like in one location or one very strong computer, it would be all computers connected together, working to execute apps and programs in this environment, facilitated through blockchain, where the programs will be described, the transactions will be stored, any changes will be recorded. And the copy of blockchain will reside on every single computer as well.

Article 1: **What is Ethereum? | The Ultimate Beginners' Guide**
https://coincentral.com/what-is-ethereum-the-ultimate-beginners-guide/

Ethereum is an open-source blockchain-based platform that essentially enables hundreds of decentralized cryptocurrencies and projects to be built and deployed without having to build their own blockchains.

Ethereum vs. Bitcoin

Bitcoin launched in 2009 as the world's first cryptocurrency, with the single goal of creating a decentralized universal currency. This currency would not require any intermediary financial institutions, but would still ensure safe and valid transactions. This was made possible by a revolutionary technology called the "blockchain."

The blockchain is a digital ledger, continuously recording and verifying records. It's used to track and verify Bitcoin transactions. Since the global network of communicating nodes maintains the blockchain, it's pretty much incorruptible. As new blocks are added to the network, they are constantly validated.

Similar to Bitcoin, Ethereum is a distributed public blockchain network. While both Ethereum and Bitcoin are cryptocurrencies that can be traded among users, there are many substantial differences between the two.

Bitcoin, for example, utilizes blockchain to track ownership of the digital currency, making it an extremely effective peer to peer electronic cash system. Ethereum, on the other hand, focuses on running the programming code of an application. Application developers largely use it to pay for services and transaction fees on the Ethereum network.

Both Bitcoin and Ethereum are "decentralized," meaning they have no central control or issuing authority. Respective miners run each network by validating transactions to earn either bitcoin (for Bitcoin) or ether (for Ethereum).

What is Ethereum?

Simply put, Ethereum is a blockchain-based decentralized platform on which decentralized applications (Dapps) can be built. Ethereum's appeal is that it's built in a way that enables developers to create smart contracts. Smart contracts are scripts that automatically execute tasks when certain conditions are met.

These smart contracts are executed by the Turing-complete Ethereum Virtual Machine (EVM), run by an international public network of nodes. The cryptocurrency of the Ethereum network is called ether. Ether serves two different functions:

1. Compensate the mining full nodes that power its network. This keeps things running smoothly at an administrative level;
2. Pay people under smart contract conditions. This is what motivates users to work on the Ethereum platform.

Welcome to a Wild Ride: Ethereum

The Ethereum white paper goes into detail for some of the potential use cases, all of which could be built through decentralized apps on the Ethereum network. The list goes on and on:

- Token Systems

- Financial Derivatives
- Identity and Reputation Systems
- File Storage
- Banking
- Centralized Autonomous Organizations
- Insurance
- Data Feeds
- Cloud Computing
- Prediction Markets

By building these apps on the Ethereum network, these dapps can utilize Ethereum's blockchain instead of having to create their own.

The Ethereum Virtual Machine

Unlike these early blockchain projects, Ethereum allows users to create their own operations. The Ethereum Virtual Machine (EVM) makes this possible. As Ethereum's runtime environment, the EVM executes smart contracts. Since every Ethereum node runs the EVM, applications built on it reap the benefits of being decentralized without having to build their own blockchain.

Smart Contracts

Smart contracts are strings of computer code capable of automatically executing when certain predetermined conditions are met. Instead of requiring a single central authority to say "yay" or "nay," these contracts are self-operated. This not only makes the entire process more effective, it also makes it more fair and objective. The takeaway: Smart contracts can automate a variety of tasks, without requiring intermediaries. All a smart contract needs is the arbitrary rules written into it.

Ethereum's Challenges and Initiatives

Handling financial transactions alone presents hugely complex problems in terms of reliability and security. And since the Ethereum network comprises a general purpose blockchain that handles assets other than money, more complex challenges arise beyond

mere financial transactions. Moving into the future, Ethereum confronts issues of scalability, energy consumption, security, privacy, and decentralization.

Decentralized Apps (Dapps)

Most of us have a pretty good understanding of what an application (app) is. An application is formally defined as a program or piece of software designed and written to fulfill a particular purpose of the user. We use apps every day: Apps allow us to check our bank balance, scroll through a live feed of pictures, or even launch a Flappy Bird into oblivion.

Now take this definition and decentralize it. Dapps serve similar functions, but run on an entire network of nodes rather than a central source. The fact that they are decentralized gives dapps an enormous advantage over traditional apps.

You know when Instagram is down because the server is down? This doesn't happen with dapps. How about when Zomato got hacked and exposed the information of 17 million people? This doesn't happen either.

Moreover, Dapps are:

- Open Source – Dapps allow users to view the app code on both the frontend and backend. No sketchy "allow us to use your location" nonsense unless otherwise stated;
- Autonomous – Dapps automatically act by the rules encoded into them. No room for outside corruption;
- Secure – Data and protocols are stored on the blockchain cryptographically. No hacks;
- 100% Uptime – The blockchain is always running, meaning zero downtime for dapps. No crashes;
- Easier to Implement – Developers wanting to take advantage of blockchain technology do not need to create a new blockchain. The framework is there, saving dapp creators a ton of time and effort spent creating a potentially subpar framework. In order to run on this decentralized network, dapps just pay transaction fees.

In many cases, front-end users can't even distinguish dapps from regular apps. Dapps typically use HTML/JavaScript web applications to communicate with the blockchain, appearing the same to users as many applications you're already using today.

Ethereum vs Bitcoin: Continued

While the two cryptocurrencies serve different purposes, Ethereum provides a number of benefits over Bitcoin:

- Shorter Block Times – On Ethereum, blocks are mined roughly every 15 seconds compared to Bitcoin's 10-minutes rate. This shorter time allows the blockchain to more quickly start confirming transaction data, although it also means more orphaned blocks;
- More Sophisticated Fee Structure – Ethereum transaction fees are based off storage needs and network usage. Bitcoin transactions are limited by block size and compete with each other;
- More Sophisticated Mining – Bitcoin mining currently requires ASICs (Application-Specific Integrated Circuits), necessitating a large amount of capital investment to mine. Ethereum's mining algorithm was designed with ASIC-resistance in mind, thus leveling the playing field and aiding in the decentralization of mining.

Ethereum arguably currently functions better than Bitcoin as a currency. With Ethereum, you can reliably send transactions faster, pay lower transaction fees, and mine at a more profitable rate (although it still has its downfalls for miners).

However, Bitcoin does have a relatively more stable price—and therefore functions as a better value storage option—from a trading and value storage perspective. Ethereum is much younger but has covered a substantial amount of ground in recent years. Although Ethereum certainly shows promise as a currency, its true potential lies in features nonexistent in Bitcoin's code.

The DAO: Trouble in Paradise

The most famous DAO was simply known as The DAO. The nearly identical name causes a lot of confusion for people and gives DAOs a bad reputation.

The DAO was a decentralized autonomous organization primarily functioning as its own investor-directed venture capital fund. It didn't have the conventional management structure or board of directors, was not tied to any particular government, and instead ran on open source code. The DAO was set up to give funders the power to vote for which dapps deserved investment through DAO tokens.

Dapps had somewhat of an approval process:

- Get whitelisted by reputable figureheads in the Ethereum community;
- Get voted on by those who held DAO tokens;
- Get an approval of 20% in the vote in order to receive a share of DAO funds they required to get started.

The DAO is most famous for the largest crowdfunding campaign in history, raising over $150 million in ether from more than 11,000 investors. The DAO is also most infamous for getting hacked for $50 million. This hack inevitably caused a split in the Ethereum community, creating what we now know as Ethereum (ETH) and Ethereum Classic (ETC).

The hack happened because of The DAO's "Split Function." Funders who wanted to exit The DAO could use its "Split Function," which would give them back the ether they had invested. The only stipulation was that existing funders had to hold their ether for 28 days before they could withdraw them.

On June 17th 2016, an unknown person or group of people took advantage of a lapse in the Split Function's security with a simple recursive function. This frustratingly easy hack allowed the hacker(s) to repeat their request to withdraw the same DAO tokens multiple times before the system registered it as $50 million.

The news of this hack created chaos in the Ethereum community. While this hack had nothing to do with the Ethereum platform and everything to do with The DAO platform, many members of the Ethereum community were invested in The DAO. The community as

a whole had 28 days to come up with a solution, which ended up being to "fork"—stop the current blockchain entirely and create something new from scratch.

What is a Smart Contract

Smart Contracts are programs (codes) that run on the blockchain.

Bitcoin has a Bitcoin script that allows people to code things on the blockchain. Ethereum uses the programming language Solidity, which allows people to code things also. The difference between them is that the Bitcoin script isn't Turing-Complete and the Solidity is Turing-Complete.

If a program language is Turing-Complete it means that people can code absolutely any logic into that language. Bitcoin Script isn't Turing-Complete because it doesn't have loops, as the Smart Contracts run in every node and it slows down the network. Solidity is Turing-Complete because they included loops in their language.

A blockchain can have smart contracts with data blocks, which means that everyone in the network will also have a copy of that program. Every node has:

- History of all Smart Contracts;
- History of all transactions;
- Current state of all Smart Contracts.

Article 1: **Blockchain: the solution for transparency in product supply chains**
https://www.provenance.org/whitepaper

Demand for transparency is increasing

Without understanding the impacts of goods and services, we buy into systems that deplete natural resources, worsen environmental and social problems and endanger humans and ecosystems. Supply chains are conventionally held secret, limiting the stakeholders who can prevent environmental, social and health and safety problems.

There is a growing rallying call by customers and governments demanding more transparency from brands, manufacturers, and producers throughout the supply chain.

The market for products of proven origin is growing. In the future, regulations will require companies to transparently disclose reliable information about their business footprint.

Pioneering companies have long realized the competitive advantage of open, transparent supply chains and sustainable manufacturing. Sustainability standards and certification have been an important tool to enable choice differentiation and conscientious consumption, yet in the end the outcome of certification is often just an image file or printed label on the packaging whose actual meaning is difficult to know and hard to verify. Guaranteeing the integrity of certificates is a costly process that, despite laborious audits, still struggles to assure the validity of the claims being made. Worldwide expansion of certification schemes in regions with levels of high corruption further endangers credibility.

Centralized systems can't power transparency

Despite various efforts, full "chains of custody" that tell the stories of products remain largely rudimentary and difficult to verify. Fragmentation of these efforts make them open to fraud. To connect the dots, nominally neutral, not-for-profit or governmental entities are commissioned with the task of creating a centralized data storage to enable a flow of trusted information.

In the face of these efforts, we must ask ourselves: can one organization be trusted to broker all data about every product's supply chain? The truth is that no single organization can, and that relying on one party (or even a small collection of cooperating parties) creates an inherent bias and weakness in the system. If the party were the brand itself, or the most powerful actor in the supply chain, then it would be responsible ultimately for only its own bottom line; this could lead to selective disclosure or, worse, extortion. If the supply chain data were gathered by a third party, it would have to be both totally unbiased and properly incentivized to deliver the technical capability of running the system.

Despite these difficulties, the idea of using a centralized system with a governing third party was, until recently, the only conceivable way to achieve data and transaction transparency along supply chains. Today, however, a new technology called the blockchain presents a whole new approach. The blockchain is a recent development in the field of computer science, which uses a global peer-to-peer network to provide an open platform that can deliver neutrality, reliability and security.

Non-localization: A truly global computer running by consensus

Personal computers are limited by the physical world. Even though it may seem that modern applications run on several devices, to keep consistency an application's core program is in fact executed on a single, centralized server, with the client device serving merely as a powerful display.

In contrast, there is no single machine that governs the business logic or the data on which a blockchain operates. Instead, the data on a blockchain is determined by consensus, which is a defined convention for how to execute and administer the business logic. The magic of the blockchain and its surrounding incentive structure is such that users can then unambiguously discover the state of the system, not from a single particular authority but rather by independently applying common rules and publishing data openly.

A machine of unparalleled digital security

With the blockchain, security is different: it doesn't matter who or where the user is, because all information provided to the blockchain is accepted only if it is authenticated. This authentication is provided in the form of an unforgeable digital signature: a cryptographic mechanism that allows someone to prove their identity without enabling someone else to impersonate them in the future. Thus, it does not matter what your job is or what your access capabilities happen to be, as you simply cannot interact with the blockchain unless you provide the digital "key" required for the interaction that you own. This means that elevated privilege levels are curbed or removed entirely, and the security risk of the weakest link is drastically reduced.

A perfectly auditable system

A blockchain is different, as by design it is perfectly auditable. Each individual operation or interaction, such as the provision of a new employee or the recording of outgoing stock, is perfectly recorded and archived. Auditing is thus as simple as joining the blockchain network, as this allows one to "replay" the operations of the past in order to build a correct model of the present. Combined with the absolute guarantees of authenticity for every interaction, strong and agile data systems can be facilitated that are at their core resilient to coercion and human factors.

Implementing supply chain certification on the blockchain

The blockchain removes the need for a trusted central organization that operates and maintains this system. Using blockchains as a shared and secure platform, we are able to see not only the final state, but crucially, we are able to overcome the weaknesses of current systems by allowing one to securely audit all transactions that brought this state of being into effect, to inspect the uninterrupted chain of custody from the raw materials to the end sale.

The blockchain also gives us an unprecedented level of certainty over the fidelity of the information. We can be sure that all transfers of ownership were explicitly authorized by their relevant controllers without having to trust the behavior or competence of an incumbent processor. Interested parties may also audit the production and manufacturing avatars and verify that their “on-chain” persona accurately reflects reality.

Information architecture for a certification and chain-of-custody system on the blockchain

1. Producers;
2. Manufacturers;
3. Registrars, which are organizations that provide credentials and a unique identity to actors;
4. Standards organizations, which define the rules of a certain scheme;
5. Certifiers and auditors, which are agents that inspect producers and manufacturers and verify certain standards;
6. Customers, the buyers of products all along a supply chain, including the end consumer.

Registration program

It is this program alone that forms the fundamental trust relationship between the customer and the system as a whole. All other programs derive their “trustability” through their own reputation (which may be imported through their real-world name). This program will initially be deployed by the registrar, who implements a process for the registration of named participants. Such participants may request registration of their

digital identity, which links their real-world identity with their blockchain-based digital identity, thus allowing them to interact with the blockchain using their real-world identity. Upon request, the registration authority verifies their identity and records the result in the blockchain, available for all to inspect.

Standards programs

These programs represent the implementation of schemas for proper recognition of a standard. Through these programs, standards organizations provide for the creation of compliant production or manufacturing programs, allowing instances or batches of goods and materials to be added to or processed on the blockchain. Such producers or manufacturers may require inspection by a certifier or auditor of their facilities and processes to be able to obtain and operate a certified program. Successful verification results in the deployment of a production or manufacturing program that is both registered with the certification program and authenticated by an auditor, and allows a producer to create the digitally tradeable equivalent of a good, which acts as its blockchain-based avatar.

Production programs

These programs are used by producers to prove the creation of materials or primary goods. The program specifies and implements the parameters for each production facility, including:

- The certification of the production capacity for the production of the good;
- A taxonomic description of the good, which would include a detailed description of the output, together with any additional “tags” to help identify specific attributes;
- The production accounting.

These parameters can be adjusted according to desired guidelines by certifiers or following the inspection by an auditor, and in case of an unsuccessful audit, the program can be easily (temporarily) revoked if necessary. Since they are principally responsible for the creation of goods, producer programs are the root for the traceability of finished goods, which then link back to the identity provided by the registrar.

Manufacturing programs

These programs implement the transformation of input goods from production into output goods. Much as with production programs, once deployed by the certifier the programs are operated by manufacturers, but with one additional constraint: input goods must be "used" for any output to be created, just as in the physical world. For example, the registration of a certain amount of organic cotton fabric requires as input the appropriate amount of raw organic cotton, and after this usage the raw organic cotton should no longer be usable. Because of its auditability, the blockchain provides the same cast-iron guarantee as in the physical world; namely, that creation of an output good can happen if and only if the required input is used.

User-facing applications facilitate access to the blockchain

By design, every transaction along a supply chain on the blockchain is fully auditable. By inspecting the blockchain, smartphone applications can aggregate and display information to customers in a real-time manner; furthermore, due to the strong integrity properties of the blockchain, this information can be genuinely trusted. A thoughtful user interface that sheds light on the digital journey of a product can empower better purchases by giving users a true choice that they can exercise.

Public-Private Key Infrastructure

Public/private key infrastructure allows us to mimic a physical signature by way of probably registering our identity with a digital document or instruction without at any time giving others the ability to further produce such signatures for other instructions or documents.

Through the use of functions with special properties, it is possible to hold a small piece of data known as a secret (or private key), and use it to demonstrate that you have explicitly sanctioned a particular piece of information without ever uncovering that secret to another party. To do so, the secret is combined with the document in question to produce a signature. This may be freely distributed. All secrets have a counterpart public key, which may be published by the secret holder as their identity. When a third party recombines the document with the signature, they are able to retrieve not the secret, but rather the secret's public counterpart, the public key and the secret holder's published identity. This

allows them to be sure that the document was sanctioned by the secret holder without ever knowing their secret and thus compromising the fidelity of future signatures.

The blockchain brings significant operational benefits

- Interoperable: a modular, interoperable platform that eliminates the possibility of double spending;
- Auditable: an auditable record that can be inspected and used by companies, standards organizations, regulators, and customers alike;
- Cost-efficient: a solution to drastically reduce costs by eliminating the need for "handling companies" to be audited;
- Real-time and agile: a fast and highly accessible sign-up means quick deployment;
- Public: the openness of the platform enables innovation and could achieve bottom-up transparency in supply chains instead of burdensome top-down audits;
- Guaranteed continuity: the elimination of any central operator ensures inclusiveness and longevity.

Decentralized Applications (Dapps)

A dapp contains an interface for people to connect with applications on blockchain. It has a front-end and a back-end, which is a Smart Contract (or you can think of it as an API for the interface to connect with the blockchain).

If the application developed needs to connect to a server, and this server uses blockchain, the Smart Contract is the "API" that connects it and the front-end is developed for people to interact with it.

Steemit is a dapp like twitter but decentralized.

Ethereum Virtual Machine & Gas

The more the system grows and becomes complex, the more security threats it has.

The first threat is the looping concept - that's why bitcoin script never included loops in its language - because people can accidentally (or not) develop an infinite loop or a really

heavy loop that would slow down the network. But Ethereum has loops in its language (Solidity), so how is it not a potential threat in there?

- Gas Concept: any computation that runs on the blockchain, the developers of the smart contract need to pay - you have an ocean of gas and use that gas to pay for the computation that you need to run;
    - It doesn't use "eth" because eth is more volatile than gas (its value is defined by consensus);
- So if there's an infinite loop or a really heavy loop, you will run out of gas and the application will stop working;
- You'll also run out of money and be penalized for writing that code => it makes people write good and efficient codes.
- ethgasstation.info => helps developers estimate how much gas the application will need.

The second threat is the Smart Contract concept - as a Smart Contract runs on every single node, how is this convenient for a virus or someone to access your private files in your private computer?

- This is solved with an Ethereum Virtual Machine (EVM) - a computer that runs on the computer and encapsulates everything there, so it doesn't know anything about the personal computer and doesn't have access to any files in it.

The purpose of bitcoin is cryptocurrency. The purpose of ethereum is dapps/web3.0.

Decentralized Autonomous Organizations (DAO)

An organization is a hierarchical structure where you have directors, managers and employees that follow protocols/procedures in order to make the company operate. They know the protocols they have to follow because of their contracts, which says exactly what is expected from them.

So, if we replace these people by smart contracts, write all the protocols and code them into smart contracts, we have a Decentralized Autonomous Organization (DAO). It doesn't need any human interaction, everything is done by the Smart Contracts.

Article 1: **DAOs, DACs, DAs and More: An Incomplete Terminology Guide**

https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide

One of the most popular topics in the digital consensus space is the concept of decentralized autonomous entities. There are now a number of groups rapidly getting involved in the space, developing "decentralized autonomous companies", decentralized applications and decentralized autonomous organizations. All in all, it is safe to say that "DAOism" is well on its way to becoming a quasi-cyber-religion. However, one of the hidden problems lurking beneath the space is a rather blatant one: no one even knows what all of these individual terms mean.

Smart Contracts

A smart contract is the simplest form of decentralized automation, and is most easily and accurately defined as follows: a smart contract is a mechanism involving digital assets and two or more parties, where some or all of the parties put assets in and assets are automatically redistributed among those parties according to a formula based on certain data that is not known at the time the contract is initiated.

Autonomous Agents

Autonomous agents are on the other side of the automation spectrum; in an autonomous agent, there is no necessary specific human involvement at all; that is to say, while some degree of human effort might be necessary to build the hardware that the agent runs on, there is no need for any humans to exist that are aware of the agent's existence.

A full autonomous agent, or a full artificial intelligence, is the dream of science fiction; such an entity would be able to adjust to arbitrary changes in circumstances, and even expand to manufacture the hardware needed for its own sustainability in theory. Between that, and single purpose agents like computer viruses, is a large range of possibilities, on a scale which can alternatively be described as intelligence or versatility.

Autonomous agents are some of the hardest things to create, because in order to be successful they need to be able to navigate in an environment that is not just complicated

and rapidly changing, but also hostile. If a web hosting provider wants to be unscrupulous, they might specifically locate all instances of the service, and then replace them with nodes that cheat in some fashion; an autonomous agent must be able to detect such cheating and remove or at least neutralize cheating nodes from the system.

Decentralized Applications

A decentralized application is similar to a smart contract, but different in two key ways. First of all, a decentralized application has an unbounded number of participants on all sides of the market. Second, a decentralized application need not be necessarily financial. Because of this second requirement, decentralized applications are actually some of the easiest things to write.

Generally, decentralized applications fall into two classes, likely with a substantial gray area between the two. The first class is a fully anonymous decentralized application. Here, it does not matter who the nodes are; every participant is essentially anonymous and the system is made up of a series of instant atomic interactions. The second class is a reputation-based decentralized application, where the system (or at least nodes in the system) keep track of nodes, and nodes maintain status inside of the application with a mechanism that is purely maintained for the purpose of ensuring trust. Status should not be transferable or have de-facto monetary value.

Decentralized Organizations

In general, a human organization can be defined as combination of two things: a set of property, and a protocol for a set of individuals, which may or may not be divided into certain classes with different conditions for entering or leaving the set, to interact with each other including rules for under what circumstances the individuals may use certain parts of the property.

The idea of a decentralized organization takes the same concept of an organization, and decentralizes it. Instead of a hierarchical structure managed by a set of humans interacting in person and controlling property via the legal system, a decentralized organization involves a set of humans interacting with each other according to a protocol specified in code, and enforced on the blockchain. A DO may or may not make use of the legal system for some protection of its physical property, but even there such usage is secondary.

Decentralized Autonomous Organizations

The ideal of a decentralized autonomous organization is easy to describe: it is an entity that lives on the internet and exists autonomously, but also heavily relies on hiring individuals to perform certain tasks that the automaton itself cannot do.

The main difference between a DA and a DAO is that a DAO has internal capital; that is, a DAO contains some kind of internal property that is valuable in some way, and it has the ability to use that property as a mechanism for rewarding certain activities.

The obvious difference between a DO and a DAO, and the one inherent in the language, is the word "autonomous"; that is, in a DO the humans are the ones making the decisions, and a DAO is something that, in some fashion, makes decisions for itself. This is a surprisingly tricky distinction to define because, as dictatorships are always keen to point out, there is really no difference between a certain set of actors making decisions directly and that set of actors controlling all of the information through which decisions are made.

DOs and DAOs are both vulnerable to collusion attacks, where (in the best case) a majority or (in worse cases) a significant percentage of a certain type of members collude to specifically direct the D*O's activity. However, the difference is this: in a DAO collusion attacks are treated as a bug, whereas in a DO they are a feature.

Decentralized Autonomous Corporations

Decentralized autonomous corporations/companies are a smaller topic, because they are basically a subclass of DAOs, but they are worth mentioning. Since the main exponent of DAC as terminology is Daniel Larimer, we will borrow as a definition the point that he himself consistently promotes: a DAC pays dividends. That is, there is a concept of shares in a DAC which are purchaseable and tradeable in some fashion, and those shares potentially entitle their holders to continual receipts based on the DAC's success. A DAO is non-profit; though you can make money in a DAO, the way to do that is by participating in its ecosystem and not by providing investment into the DAO itself. Obviously, this distinction is a murky one; all DAOs contain internal capital that can be owned, and the value of that internal capital can easily go up as the DAO becomes more powerful/popular, so a large portion of DAOs are inevitably going to be DAC-like to some extent.

The DAO Attack

DAO (Decentralized Autonomous Organization): created in 2016 on Ethereum by Vitalik to help with the development of decentralized applications to run on Ethereum Blockchain. It's stateless (doesn't belong to any country).

There was an error on its code, on the way that the smart contracts were made, and DAO was attacked and hacked in June 2016. The attacker didn't do anything illegal, he saw this error and moved the money from DAO to his own account, and nobody could do anything even knowing what was happening, because the DAO is ruled by the smart contracts, it doesn't do what it's told to and it's impossible to change the smart contract once on the blockchain.

Thankfully, the attacker couldn't take away all the money, he would have to wait 30 days, and in this period of time, the whole community had the opportunity to discuss what should be done. They got into a dilemma: "code is law?". People who believed code is law said that they couldn't change the code because that was the way the smart contract operated and it shouldn't be immutable. People who believed code isn't law said that they should do a hard fork and change the code on smart contracts.

Eventually, they did the hard fork and changed the code. This splitted Ethereum into ETH (Ethereum) and ETC (Ethereum Classic).

The problem was on DAO and not Ethereum.

Article 1: **The Ether Thief**

https://www.bloomberg.com/features/2017-the-ether-thief/?leadSource=uverify%20wall

A Turkish professor noted that the smart contract he was looking at might have a problem on line 666 and feared that the bug could allow a hacker to make unlimited ATM-like withdrawals from the millions, even if the attacker had only 10$ in his account.

This money was inside a DAO (Decentralized Autonomous Organization), governed by a smart contract, intended to democratize how ethereum projects were funded. And Gun, the Turkish professor, had already been tracking and publicizing flaws in the DAO's design.

A few weeks before the attack, he even told the investors they shouldn't buy into DAO until the security issues were fixed, but as smart contracts are built to be entirely reliant on their code, the DAO code couldn't be fixed. The main problem was that the code was so new that no one knew what to expect or even if there was a real problem in it.

The DAO attack exposed the early frailties of smart contract security and left many in the community shaken because they hadn't found the bug in time, and eventually pit good hackers against bad ones (white hat vs. black hat) in the "DAO wars".

To control the attack, and try to save the money, a group of investors/developers gathered together and formed the "Robin Hood Group". But they didn't succeed in their goal, because of bad internet and family commitments.

The attacker used the address 0xF35e2cC8E6523d683eD44870f5B7cC785051a77D, an anonymous string of characters as any other one on the blockchain. He created a contract to interact with the DAO and started sending about $4.000 worth of ether through the attacker's account every 3-4 minutes to drain the DAO. But to do the contract work on the ethereum blockchain, it needs an amount of ether that came from 2 different accounts, the first one already mentioned and the second one that used an exchange called ShapeShift. ShapeShift doesn't collect any information on its users and turns one virtual currency into another, such as bitcoin in ether, in less than 10 seconds. After the attack contract stopped working, the thief needed to deploy it again, he tried but failed, and after a few more transactions, the hack whimpered to an end - and one possible reason the attack stopped was that the hacker's tokens became corrupted, so he had no way to exploit the bug.

After 4 days, when the hacker tried to attack the DAO again, the Robin Hood Group was ready to contain it. In short, what they could do was change the ethereum blockchain to fix the DAO, but only if they got a majority of computers running the network to agree to a software update. Pull that off, and it's as though the attack never happened. This is known as a hard fork. The decision stirred such strong reactions that it remains controversial a year later, both within the ethereum community and with bitcoin users who insist a blockchain's history is never to be tampered with.

Soft and Hard Forks - pt. 1

Hard Fork: new rules that allow to reverse the logic in the contract, as a new software or a new upgrade on the blockchain software, the fork is done on the software, not on the chain ⇒ some people didn't want to change to the "New Ethereum", so they remained in the Classical one. It means that Ethereum was splitted in two different currencies, ETH ("new one") and ETC ("classical one").

- If someone has coins before the hard fork is done, it will have the same amount of coins of the original chain and the same amount of coins of the new chain ⇒ so it will kinda of duplicate the initial amount.

Soft Fork: changes that don't split the currency in two or more currencies.

Soft and Hard Forks - pt. 2 (Advanced Tutorial)

The main difference between hard fork and soft fork is that Hard Forks have loosen rules and Soft Forks have tighten rules. It means that hard forks don't have to be followed by everyone on the network, but soft forks do, that's why hard forks generate two different chains and soft forks continue as one and only chain.

Initial Coin Offerings (ICOs)

ICO can happen in two layers, the most common is the token layer but it can be applied in the protocol/coin layer.

IPO (Initial Public Offering): a way that companies have been raising money from the stock market. Ex: a group of people wants to raise a company, in the beginning they put work and capital into it, before any IPO is considered, and in return they get shares of the company. The company creates its products and services and in return gets a profit. Then, they conduct an IPO, an initial public offering, and people can give cash to the company in return of getting shares in that company ⇒ the company raises cash in order to fund their operations further.

ICO (Initial Coin Offering): a way that companies have to induce people to buy their products with their own token (it's not their coin because to be a coin they'd need their own

protocol). Ex: a group of people wants to raise a company, in the beginning they put work and capital into it and in return they get shares of the company. The company creates its products and services and in return gets a profit. To scale their idea/business, they put their products in an enclosed environment where people will be able to buy these products/services if they have the company's token, that can be exchanged by fiat money. The consumers will still be able to buy their products and services with fiat money, but it will be more expensive than buying with the company's token.

The main difference between IPO and ICO is that in an IPO the public gets shares and control and entitlement to share the profit of the company, whereas in an ICO the public gets these tokens to spend later on the company's products and services or wait for these tokens to appreciate and value and sell them off on a higher price. Also, to do an IPO, the companies have to fulfill some requirements, but they can do an ICO anytime they want.

ICO Case Study

An amusement park's owner creates an amusement park's token. To support her and by speculation, some people buy these tokens for 2$. Then, more people want to buy these tokens, to use at the amusement park, so they can buy from the initial people for 6$ or from the amusement park for another price. That's how the cycle works, there's no mining on these tokens, because mining is particular from the second layer (the protocol/coin one), even though there's mining to keep them on the blockchain, but it's apart from this.

Article 1: **How Crypto Tokens Will Enable the Disruption of Businesses like Uber and Airbnb**

https://finnscave.com/2018/02/07/how-crypto-tokens-will-enable-the-disruption-of-businesses-like-uber-and-airbnb/

"If history teaches us one lesson, it is those who adapt will survive, and those who fight tooth and nail for the status quo will flounder when the world invariably changes. If many smart people say that something "is the future", it is generally worthwhile to try to figure out why."

Crypto tokens have potential far beyond being the "internet of money" – they can reshape how businesses form and operate. Businesses that are two-sided marketplaces, built

around network effects and transaction fees (Uber, Lyft, eBay, Airbnb), will be especially vulnerable to disruption from businesses built around crypto tokens.

The traditional way to build a two-sided marketplace business:

- Use equity financing from founders and accredited investors to create a marketplace;
- Use more equity financing from institutions to grow two sides of a marketplace as fast as possible;
- Reward early employees with equity and early marketplace participants with low, zero, or even negative transaction fees (cash rewards);
- Increase transaction fees as the marketplace increases in scale and monopoly power;
- Grow value from the appreciation of the equity because of its right to the future cash flows from these transaction fees;
- Founders, employees, and investors capture value by selling appreciated equity.

Using crypto tokens, these businesses will be built a different way:

- Use ICO (initial coin offering) financing from anyone, around the world, to create a no-transaction fee marketplace, with a specialized token as both the means of exchange and reward for investing;
- Use this same token to reward early marketplace participants, as well as founders and early employees;
- Grow value from increasing the value of the token, which happens through matching as many willing buyers and sellers as possible, not through transaction fees;
- Founders, employees, investors, and users of the marketplace capture value by selling tokens for US dollars, other digital tokens, or fiat currency.

In the crypto token version of this business, marketplace value is captured by ALL those who create value for the marketplace, including marketplace “users”, and that value is captured WITHOUT transaction fees. These are revolutionary concepts for how individuals can create and capture value together.

Hypothetical Example: Token-based Ride Sharing Service

To illustrate how this would work in practice, let's create a fake ride sharing company called Lyber, which uses LyberTokens as their means of exchange. Lyber works exactly like Lyft/Uber (matching riders with drivers for an agreed price), with some differences:

1. There are 1,000 total LyberTokens that are issued, and there will never be any additional tokens created;
    a. 100 of these tokens are owned by the people who work at Lyber, 100 tokens are reserved for rider/driver incentives, and then 800 of the tokens were sold to the public in an ICO at $10/token. Altogether, there are 1,000 LyberTokens that can be bought and sold freely on the open market. At the launch of the service, LyberTokens trade for $10/token.
    b. Key point: The number of tokens is fixed. The creation of these tokens is governed by digital contracts maintained on Ethereum, and the token's ledger of exchange is maintained on a decentralized blockchain.
2. Rides are bought with LyberTokens, and drivers are paid in LyberTokens;
    a. The price of a ride is pegged to the price of a LyberToken in US dollars. So, if a ride costs $10, and a LyberToken costs $10, then a ride costs 1 LyberToken.
    b. Key points: A user of Lyber can still price a ride in US dollars, and they do need to think of prices in terms of LyberToken. The US dollar can still be the dominant pricing mechanism, and a user does not need to know how much one LyberToken is worth to use Lyber.
3. The more rides that occur, the more demand for each LyberToken because there is a fixed number of LyberTokens;
    a. The price per LyberToken will naturally appreciate with the demand.
    b. For example, if at the start, a ride was $10, all 1,000 LyberTokens were available for purchase, and there were 1,000 people taking a ride at a given time who needed LyberTokens to pay for their rides. There were $10,000 of total ride value and 1,000 LyberTokens available to be purchased, so each LyberToken would maintain its price of $10/token. However, if there were 100,000 people taking a ride at a given time, so there was $1,000,000 of rides

that had to be paid for with only 1,000 LyberTokens, then each LyberToken would have to be worth $1,000, so each ride would cost .01 LyberToken.

   c. Key points: i) If the price of the token rises, it does not mean the underlying cost of the service rises. ii) If you have a set number of tokens, the more transaction value between buyers and sellers in a network, the more the value of the token will appreciate. Value is captured from matching buyers and sellers through the appreciation of the token, NOT through transaction fees.

4. If riders do not have LyberTokens, they can still pay with fiat currency, like a US dollar;
   a. To facilitate this, Lyber charges riders 3% more than the US dollar ride price.
   b. So, if the price of a ride was $10, and a LyberToken cost $10/token, the rider can choose to pay $10.30 instead of using 1 LyberToken to pay for her ride. Lyber would then take that $10.30 and purchase 1 LyberToken on the open market, which would then be paid to the driver.
   c. The driver would then have the option to keep the 1 LyberToken (either to use themselves as a rider or sell it at a later date) or to be paid in US dollars, for a 3% fee. If they choose to be paid in US dollars, the company simply takes that 1 LyberToken, sells it on the open market for US dollars, and pays the driver.
   d. Key point: Neither the rider nor the driver needs to deal with tokens if they don't want to, and the transaction cost is still far cheaper (3%) than what Lyft/Uber charges (20%). The company can essentially "force" a token economy on the business, so long as there is enough liquidity on the buy and sell side (which there should be if enough people are using the service because the marketplace is two-sided).
5. There is a ~0% transaction fee for each ride;
   a. If the ride costs 1 LyberToken, then the rider pays 1 LyberToken, and the driver receives .999 LyberTokens. The tiny transaction fees cover the cost of mining, which maintains the decentralized blockchain and gives riders and drivers security that tokens and transactions are not being faked.
   b. Key point: Whether you are rider/driver number 10 or 10 billion, you are getting a better deal than you're currently getting with Lyft/Uber because you

pay no transaction fees. There is no reason to ever use a service like Lyft/Uber that has transaction fees if a token-based service works just as well.

6. Lyber rewards drivers and riders who invite friends to use Lyber with LyberTokens;
    a. The earlier you were involved in the marketplace, the more valuable your potential LyberToken appreciation, and the more incentive you have to join early and help build the marketplace.
    b. Key point: The everyday person who provides value in the marketplace now has the opportunity to participate in its upside. The should lead to stronger marketplaces that grow more quickly and spread the value capture across more participants. For example, instead of receiving a measly coupon for inviting your friends to also drive for Uber, you could actually make serious money by earning tokens in Lyber that appreciate over time.

Foreseen Benefits and Risks

- Founders, employees, investors, and customers are all incentivized by the same thing – create and maintain a product that matches as many willing riders and drivers as possible;
- Riders and drivers no longer have to pay transaction fees;
- The earliest rider and driver participants can benefit from helping grow the network;
- Riders and drivers can further participate in the "upside" of the business by earning and holding tokens;
- Tokens are always liquid and value can be captured more easily.

Potential downsides that need to be resolved:

- The equity model strongly incentivizes employees and investors to think and act long-term – the payday comes at the end (and only at the end). In the token model, without proper governance, the payday could come at any time, reducing the motivation of the team.
- Because tokens will tend to appreciate in a deflationary system like this, there will be strong temptations for all participants to always be jumping to a new thing where they can get low price tokens. It will be like the current startup ecosystem, but on crack, because of the early liquidity.

- Not all value from the tokens will come from actual usage in the system. Because they are tradeable on the open market, they are susceptible to rampant speculation. This will shift some of the value capture away from the everyday user of the service to professional speculators. Right now the everyday user captures absolutely no value in the network they help build, so it is hard to imagine this still not being an improvement.
- Getting token economics correct will be extremely difficult. If there is no token "inflation", then the price of a token can appreciate too much, and there will be a temptation to horde.
- Businesses and employees still have real expenses that need to be paid in real currency – the will have to figure out how to pay these expenses by selling off tokens in some systemic fashion or otherwise
- Crypto-enabled businesses would be inherently less flexible than traditional equity businesses – it is much harder to pivot or recapitalize the business because the success of crypto is based on trust and rules. Equity investors with protections can be patient, token investors with no protections will not be so forgiving.
- If people start using Lyber less, and LyberToken liquidity decreases, then the price will plummet. Again, similar to the price of a startup company's stock, but on crack.

Blockchain Startups: White Papers

White Papers are a way to analyze if a project is interesting/ good or if it's just a person wanting to "surf" on the blockchain's wave/ it's a bad project.

A white paper is a description of how the project will work and if there's a coin or token, how it will work, what's the purpose and why people should use it.

Examples: SIA, Power Ledger Ecosystem (POWR Token), Provenance, Kodak, Useless Ethereum Token

Blockchain & Web 3.0

Web 1.0: 90's; plain text; some links and images; some external videos.

Web 2.0: 00's; more colors, different designs, less links (most things are internal to the website); carousels, widgets, new forms of interaction, machine learning, personalized ads.

Most things changed on the front-end, but the core of the back-end is the same.

Web 3.0: we don't know how exactly the front-end will change/ is changing, maybe some more interaction, virtual reality, "metaverse conception"; but the back-end is changing the most with blockchain technology ⇒ it's a decentralized network, where everyone is interconnected, there's no main server as the facebook one, everyone runs a copy of the blockchain/ smart contract on it's own device.

Article 1: **Why the Web 3.0 Matters and you should know about it**

https://medium.com/@Matzago/why-the-web-3-0-matters-and-you-should-know-about-it-a5851d63c949

Just like the Middle Ages, the **Web 1.0** wasn't given its name until it bit the dust. The 'World Wide Web' as it was known, was just a set of static websites with a load of information and no interactive content. Connecting meant dialing up through rickety modems and blocking anyone in the house from using the phone. It was the web of AOL chat rooms and MSN messenger, of AltaVista and Ask Jeeves. It was maddeningly slow. Streaming videos and music? Forget it. Downloading a song would take at least a day.

And then there was **Web 2.0**. The memory of bleepy modems and boring interfaces has largely floated away. Faster internet speeds paved the way for interactive content, the web wasn't about observing anymore, it was about participating. The global sharing of information spawned the age of 'Social Media'. Youtube, Wikipedia, Flickr and Facebook gave voices to the voiceless and a means for like-minded communities to thrive.

Publishing this blog post will take me a hassle-free 30 seconds, an immeasurable improvement from when it took a concerted effort between designers, developers and administrators just to make a simple website edit. We could call this the Read-Write-Publish' era — where the spread of information is as simple as those three words. So it begs the question, the web 2.0 is great, what went wrong?

- **Information is money:** personal information is an enormously valuable asset; people sacrificed security for the convenience of these services; whether they knew

it or not, their identities, browsing habits, searches and online shopping information was sold to the highest bidder.

The 3.0 revolution

The next web, they envisaged, would take nostalgic turn to the vision of the web 1.0: more 'human' and more privacy. Rather than concentrating the power (and data) in the hands of huge behemoths with questionable motives, it would be returned the rightful owners.

The vision of a fairer and more transparent web dates back to around 2006, but the tools and technologies weren't available for it to materialize. Bitcoin was still three years off, bringing with it the notion of a distributed ledger, or blockchain, for peer-to-peer digital storage. Decentralization was the idea; blockchain was the means. Now we have what is described as human-centered internet.

- **The pro-privacy, anti-monopoly web:** While the Web 2.0 democratized many power structures and created new opportunities, the economic engine is largely privatized and monopolized. The Web 3.0 is the antithesis of this, it's about multiple profit centers sharing value across an open network. It's easy to envision a not-too-distant future where crypto-based phones, VPN's, decentralized storage and cryptocurrency wallets are widespread. A future without the need of network and cellular providers that suspend or surveil our information. If we're to avoid sleep-walking into a Black Mirror style privacy dystopia, these are the tools we require.
- **No central point of control:** Middlemen are removed from the equation, blockchains like Ethereum provide a trust-less platform where the rules are unbreakable and data is fully encrypted. No government or entity will have the ability to kill sites and services; and no single individual can control the identities of others.
- **Ownership of data:** End users will regain complete control of data and have the security of encryption. Information can then be shared on a case-by-case and permissioned basis.
- **Dramatic reduction in hacks and data breaches:** Because data will be decentralized and distributed, hackers would need to turn off the entire network, while state-sponsored tools such as Vault7, used by the three-letter agencies, would

be rendered obsolete.At present, internet companies are compelled to hand over user data or succumb to having the entire database scrutinized. These data intrusions aren't just limited to major security threats such as terrorism.

- **Interoperability:** Applications will be easy to customize and device-agnostic, capable of running on smartphones, TVs, automobiles, microwaves and smart sensors. At present, applications are OS-specific, and are often limited to a single operating system. It adds expenses for developers tasked with issuing multiple iterations and updates of their software.
- **Permissionless blockchains:** Anyone can create an address and interact with the network. The power to access permissionless chains cannot be overstated. Users will not be barred on account of geography, income, gender, orientation or a host of other sociological and demographic factors. Wealth and other digital assets can be transferred cross-border, quickly and efficiently, anywhere in the world.
- **Uninterrupted service:** Account suspension and distributed denial of service are dramatically reduced. Because there's no single point of failure, service disruption will be minimal. Data will be stored on distributed nodes to ensure redundancy and multiple backups will prevent server failure or seizure.