TRANSITION – Guidelines

Contents

Subject A. Blockchain fundamentals – Blockchain architecture and principles.

1. Blockchain in a nutshell

The first concept of what we now call “Blockchain” came initially from Stuart Haber and W.Scott Stornetta. Although they did not coin the term "blockchain", in 1991, they published a document called "How to time stamp a digital document" which contained all the information and notions about what we call the Blockchain.

The technology can be easily defined as "**an ever-growing list of records, called blocks, that are linked and protected using cryptography**"[1].

Fundamentals: blocks and hash:
A block is a record and it has data inside it. It also has a value called **previous hash** and then a value which is **hash**.

A digital hash can be defined as the **fingerprint of the block** that makes it much clearer. Data and the previous hash are represented, encrypted, in a **number**. This shortened version of the data is specifically **sixty-four characters in length.**

Different blocks:

[1] Wikipedia

The first block is called "**genesis block**" because the chain starts with it and it will be there blocked forever it will be always the origin. T**he genesis block will always be the first** of the chain and cannot be substituted. **The genesis block does not have a previous hash** among its values because it is the first one. Therefore, in technical writing, the previous hash will be indicated as many zeros.

The second block
Then there is the second block that contains data, previous hash, and hash. **The previous hash is the hash of the genesis block**. The **blocks are cryptographically linked together through the hashes**. So, if anything changed block number one, the next fingerprint would also change, and it would no longer match. The fingerprint of the block will then show that tampering happened.

Going on, the third block, therefore, will have the hash of block number 2 as its previous hash value, and so on and so forth.

2. Hash Cryptography

Cryptography codes unique markers for data. A simple way to explain it would be through an example: everyone has a unique fingerprint, all different from each other. There is a possibility that two people share a fingerprint, but it is very unlikely (the probability is 1 in 60 million). In the same way in which the fingerprint identifies a person, hash identifies data.

This technology **is called Shell 256 Hash** and it was developed by NSA. This hash is called Shutdown in Physics. Shell stands for secure hash algorithm and 256 is the number of bits it occupies in memory.

It's very secure, it is currently used to store passwords, to check digital documents, and in blockchain as well.

The hash is always sixty-four characters long and consists of digits and letters because it is hexadecimal. It has numbers from zero to nine and the letters **abcdef,** so there are 16 in total. A stands for 10, B stands for 11, C stands for 12, D stands for 13, E stands for 14, F stands for 15.
Each character in the resulting hash takes up four bits because the quarter-power of two is 16 and four times 64 is 256.
This algorithm works not only for words, documents, or text documents but for any digital document.

The 5 requirements for safe hash algorithms:

- **It has to go in one direction only:** so basically it cannot be reversed/go backward.
- **It has to be deterministic:** if you take the same document, later on, and you run the set the apply the hash algorithm again, you will get exactly the same result
- It has to have **first computation**
- The avalanche effect: it is an ultra-important requirement of the hash algorithm. The avalanche effect means that if you take exactly the same document and you change it like make a tiny little change, even one bit of data, then the hash will be absolutely different. The reason it's called the avalanche effect is because of how that is implemented inside the algorithm. One change triggers a few changes and they, in turn, trigger more changes.

- The algorithm needs **to be able to withstand artificial collisions**. Collisions can be artificially created and used for illegal purposes: you can intentionally create two documents with the same hash. In this way, documents can be forged.

Collisions

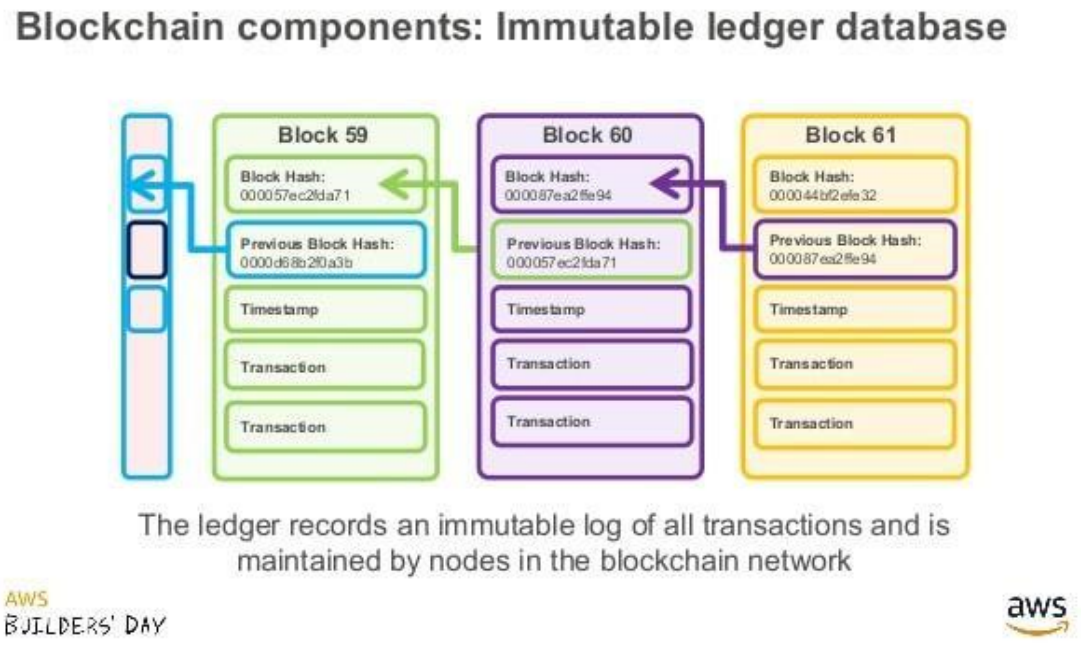
There may be two people with the same fingerprint, even if the probability is negligible, in the same way, it is possible for the hashing algorithm.

Everything is based on the **pigeon hole principle**. This means that if there are 10 pigeons and only nine holes, two pigeons will have to be put in one of those holes. So, if there is more quantity A than there are slots of quantity B, then inevitably **there will be what we call collisions**. This is however a very rare possibility

3. Immutable Ledger

If someone tries to tamper with the data in that specific block, the hash for this block will change. So that cryptographic link will no longer work because the hash is different from the hash recorded here for the previous block. The previous hash will no longer match this one. It would then be necessary to would have to change the block as well.

Due to the cryptographic link, as soon as they change one block, all the subsequent blocks will no longer be valid. They will no longer be linked to the chain and it will be very easy to tell and very difficult for the person to tamper with the record.



Source:
https://www.slideshare.net/AmazonWebServices/reinvent-roundup-time-stream-quantum-and-managed-blockchain

Unlike a physical ledger where you can only change one entry, here you would need to change all subsequent entries. This is what is meant by an immutable ledger: you cannot change entries as soon as they enter the block.

Property Ledgers is one of the biggest examples mentioned when talking about blockchain outside of finance, Bitcoin, cryptocurrency, etc. For example, during the financial crisis in 2008, even Bank of America was foreclosing on mortgages. Relying on unreliable ledgers they made many mistakes, confusing homes where there were no mortgages taken out for those actually to be foreclosed and trying to foreclose on properties they didn't even own.

Blockchain technology can add more protection and make the whole ledger immutable. In this way, it is very difficult for someone to change previous records in the ledger and thus make it more reliable.

4. Peer to Peer Network

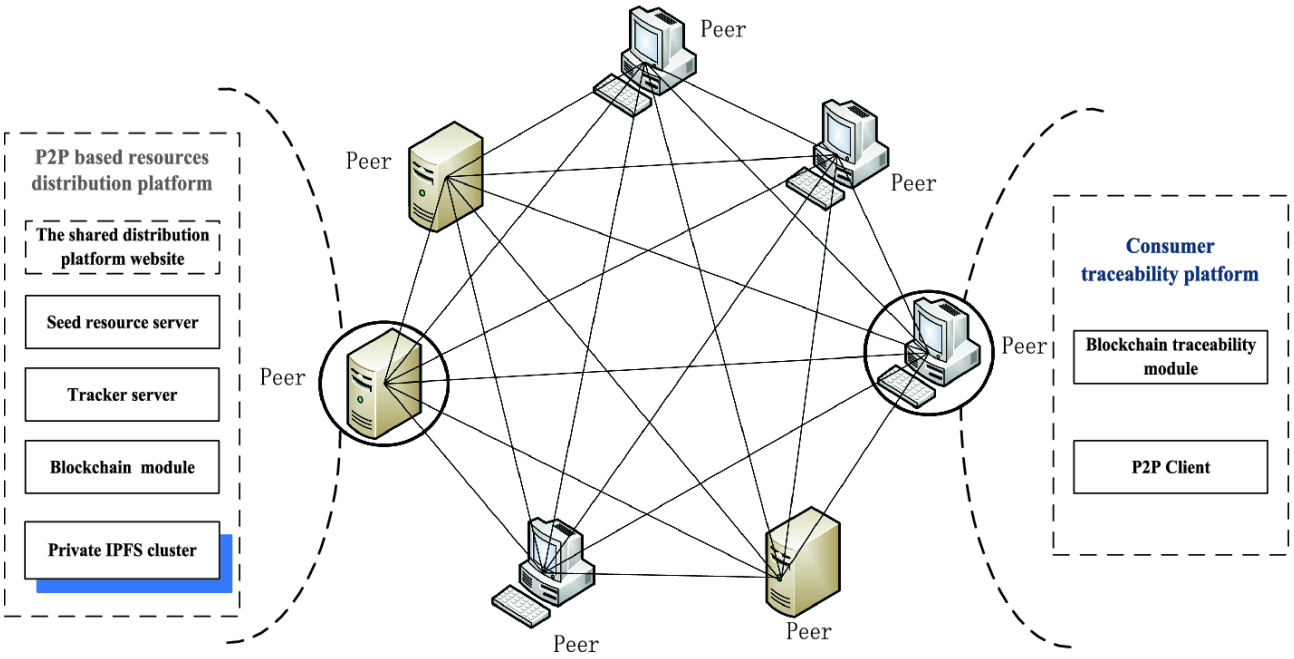
Using the blockchain allows the immutable ledger to be protected, preventing forgeries that could be facilitated by traditional means. However, two questions remain:

- if the 'scam' is potentially worth a lot of money and you have enough time, someone could make the effort to change all the blocks and hashes: what would prevent them from doing so?
- or if a system or input error occurs and the changes cause the data to be lost? How can this be remedied?

These problems can be solved by **distributed peer-to-peer networks.**

In a distributed P2P System there are lots of computers, all interconnected. Ideally, the more they're connected, the better. The blockchain is copied across all of those computers.

The problem occurs when someone tries to hack into the system or there is a technical problem. Taking the example of the attack: the attack takes place on a block, the data of which is changed. As said before, it is possible to change all the other blocks following the attacked block, modifying the chain, especially if it is a worthwhile operation. But the situation changes in P2P Network context. The network constantly checks peers to see if their blocks match, so any changes or problems are immediately reported.



https://link.springer.com/chapter/10.1007/978-981-15-2777-7_50

The rest of the computers 'realize' that there has been an attack because the encrypted chains no longer match. Immediately the anomalous values are recognized and the system replaces them with the original values on the other computers, restoring the original data. So to attack successfully, the hacker should be able to attack all the blockchains simultaneously. The hacker has to take more than 50 percent of the computers at the same time to successfully replace the chain. If you have 10000 computers, you would have to hack into five thousand and one computers at the same time and do it within a couple of minutes and that's practically impossible.

5. How Mining works – The Nonce

One question must be asked at this point: if it is so easy to take the block number, the data and the previous hash put into the hashing algorithm and get a hash in no time at all, why is there so much focus on mining? How come there are so many mining platforms around the world and so much computing power dedicated to them? This is because it is not that simple.

There is, in fact, another element within the block: the field is called **Nonce** (number used only once). This is what mining is all about. **This value**, together with the data and the previous hash, combined with the algorithm **defines the hash of the block** and thus determines the chain.

The nonce provides extra control and flexibility: you can manipulate the hash value by controlling the nonce. We must not forget that:

- The block number cannot be changed
- You cannot change the hash directly, because it would invalidate the chain
- You cannot change the previous hash because it is defined by the previous block
- You cannot change the data because it would mean tampering with it, which would defeat the purpose of a block change (It has to be an immutable ledger, to prevent tampering)

By changing the nonce, which is a number, the **hash changes substantially**. **This is due to the avalanche effect**.

6. How Mining works – The Cryptographic Puzzle

The blockchain system or the algorithm will set a target for miners to accomplish a certain hash. The target is set arbitrarily, without economic or other reasons. The hash must meet this target and be below the set limit. A good way of thinking about the target is in terms of leading zeros: the lower it is, the smaller the number and therefore more leading zeros there will be.

Miners change the nonce in order to try to guess a value of the nonce that will generate a hash **below the target**. When, by trial and error, they find the nonce that allows the hash to be placed below the target (and therefore to have a certain number of zeros) they call it **Golden Hash**.

Once defined, they can create a new block to add to the blockchain. The block is accepted by the blockchain **only when the hash is below the target.**

To define a hash that satisfies the required characteristics, there is no linear process: it is completely unpredictable and that is a very important feature. The process you need to follow to find the hash is called **Cryptographic Puzzle. Without the avalanche effect**, which substantially modifies the hash by making very small changes, **this puzzle would not exist**. It would be enough to decrease or increase the Nonce number to be sure of meeting the target. However, precisely because of this effect, the search requires attempts and not defined and unambiguous steps**.**

7. Byzantine Fault Tolerance

It is a very important characteristic, not only for blockchain but also for any type of decentralized system. To explain the concept there is a story. Four Byzantine generals surround a castle and want to conquer it.
They can only win if the majority of them come to a consensus of what to do. Whether they attack or retreat, the majority of these generals have to come to an agreement:

- If three out of four say "we are attacking" and they attack, they will win;
- If three or four out of four say "we are retreating" and they retreat, they'll be all safe and fight;
- However, if they don't come to a consensus, they will be destroyed by the enemy.

Between them there are two figures:

- the supreme commander
- a probable traitor (but the commander himself may be the traitor)

No one knows who is who. It is possible to come up with an algorithm that will help decide who is the traitor.
They can communicate with each other, but they can only deliver oral messages. The algorithm is looking at the majority of the messages that they get to base their decision on that.

For example:

1) The commander orders each one separately to attack, but the three generals do not know whether he is the traitor. Then they will have to look at the majority in the content of what the commander said. The traitor will tell the other two that he received the order to retreat, lying. The other two will tell the truth, saying that the commander ordered each of them to attack.
2)

Can **consensus be reached**?

- The commander will attack because he ordered it.
- General 1 has two positive responses for the attack (commander and general 2)
- General 2 has two positive responses for the attack (commander and general 1)

So the majority reaches an agreement and attacks.

3) **What happens if the commander is also the traitor?**

- If he told everyone to attack it would be pretty stupid of him, because then they would tell each other to attack and they would attack and take the castle
- The same for retreating
- He could tell one to retreat and two to attack: but even then all would have two positive responses to the attack and one negative, so all three would attack

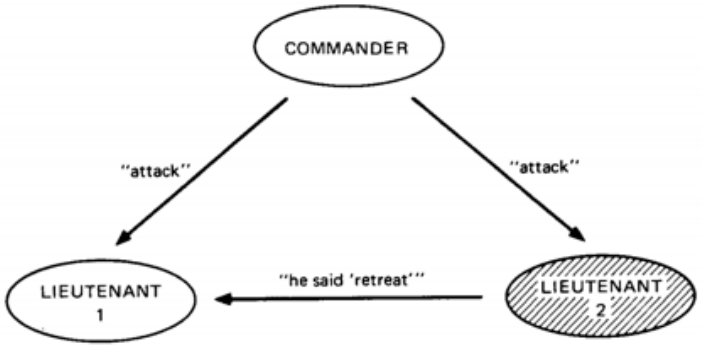


Fig. 1. Lieutenant 2 a traitor.

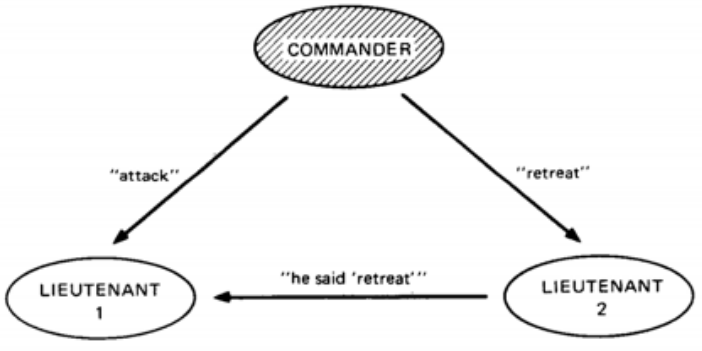


Fig. 2. The commander a traitor.

https://blog.goodaudience.com/the-byzantine-generals-problem-d979c5d8c467

Deciding based on the majority of information is the algorithm and it is Byzantine fault-tolerant. The question is to what level is it tolerant or to what level is intolerant? If there were two traitors, the

mechanism could not work. For instance, for this algorithm to work, you have to have no more than 33% traitors: if there are four traitors out of ten generals, it will not work.

That is the level of tolerance of this system in the sense of traitors. How does this go back to blockchain or like other systems that are decentralized or more technological?

What happens in the blockchain? If anyone tries to attack the system, it is necessary to put in place a consensus protocol that will allow to protect the system and to make it as tolerant as possible. **That's the whole concept of Byzantine fault tolerance**.

https://blog.bitnovo.com/en/what-is-byzantine-fault-tolerance-a-quick-guide/

Pros and Cons

Pros:

- The network does not need multiple confirmations, nor a waiting period to ensure that a transaction is secure or valid after it is included in a block,
- Consensus can be reached without requiring excessive energy usage for miners.

Cons

The system is vulnerable to Sybil attacks, that are executed by the same entity that controls the network entities and therefore corrupt the system.

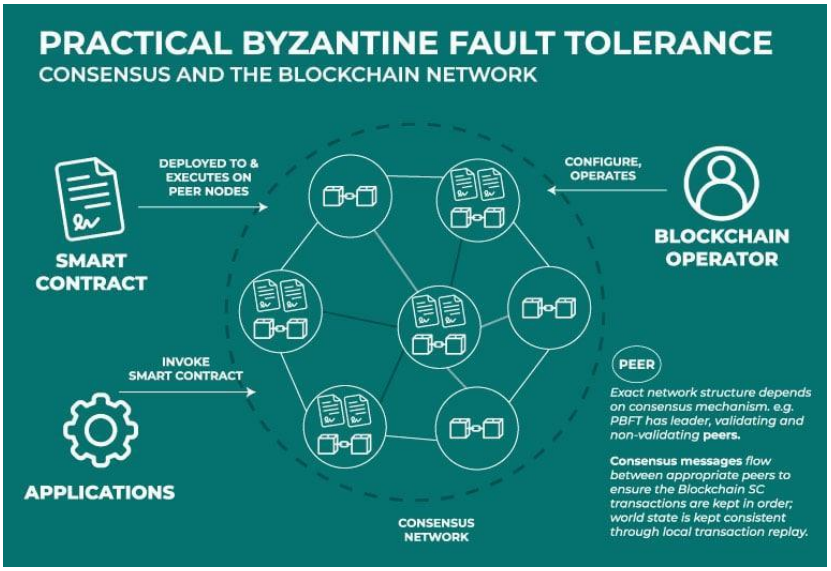
Overall, the Byzantine fault tolerance is considered an attractive alternative to other algorithms such as PoS (proof of stake), PoW (Proof of Work) consensus and PoI (Proof of Importance).

8. **Consensus Protocol - Defense Against Attackers**

The consensus protocol for a blockchain has to solve two main challenges:

1) **protect the network from attackers:** what happens if an attacker tries to put a malicious block at the end of the chain
2) **the challenge of competing chains**

In a large blockchain, because it is distributed across the world, it can be a lag between nodes, especially those that are far away from each other. It could also happen that two nodes that are far away from each



other can successfully find a block at the same time. This is not an attack, but a delay of information that arrives just after the creation of the block.
For the blockchain, this is a problem because it needs to be in consensus on how to keep growing. If there is no consensus, there will be conflicts and then they will split up into two and then later on the block channel split up into four and eight and so on.

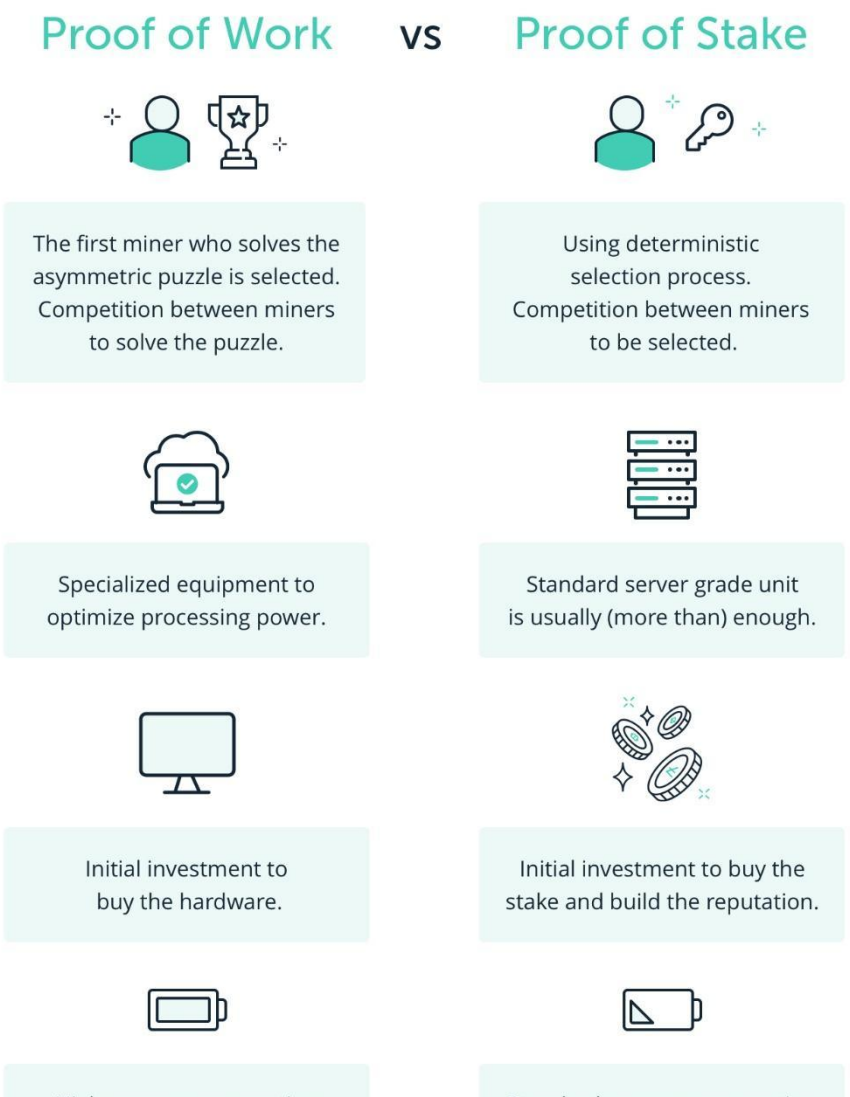
At this point is important to note that there are multiple types of consensus protocols, that we mentioned before:

- Proof-of-Work (PoW)
- Proof-of-Stake (PoS)
- Others

PoW is the original algorithm, and it is currently used, among others, by Bitcoins.

Where does the term come from? Finding the correct hash, the one that meets the target, requires a lot of work, many hours, and therefore a lot of electricity. The final hash is **proof of the work** that went into finding it, proof that the search had a purpose and that it achieved it.

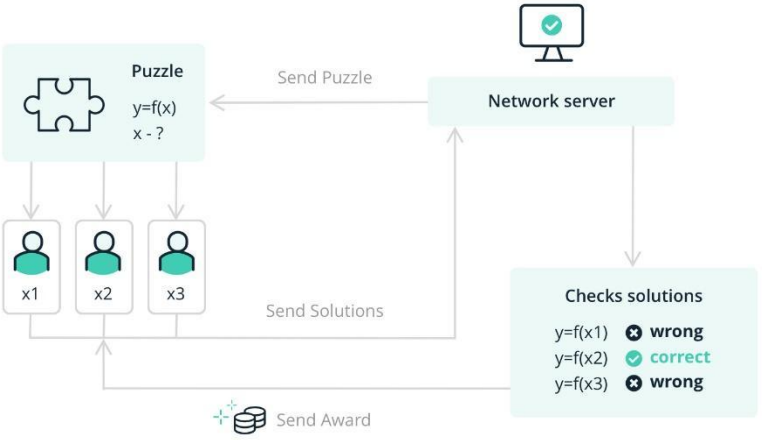
When a miner adds a new block, there comes there is a new block. The network or the blockchain will reward the miners for mining and they also will get the fees associated with the transactions that are included in that block: so there is a monetary incentive, but they have to play fair.

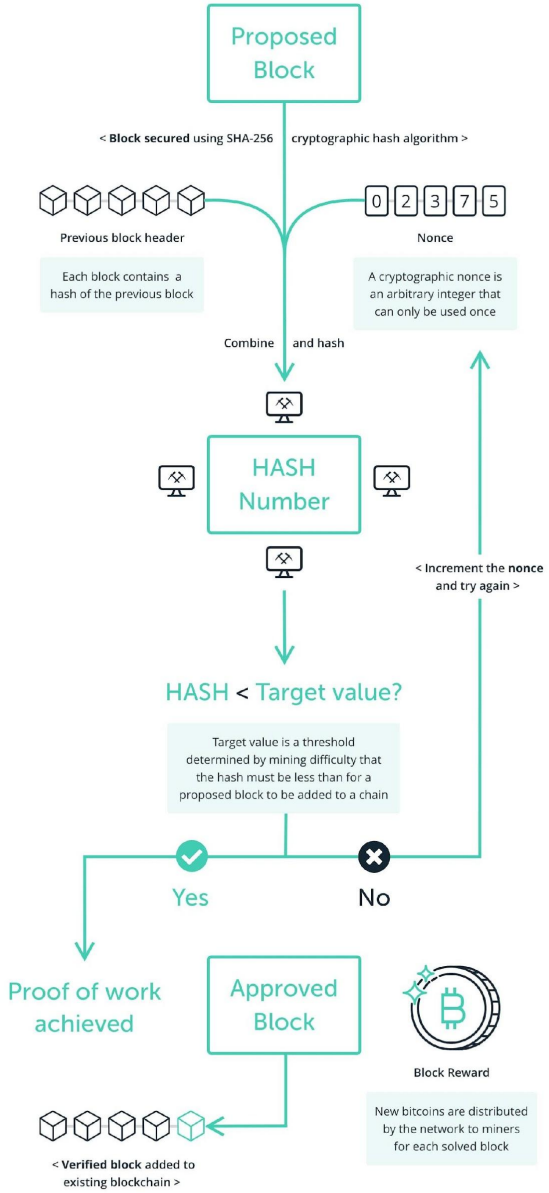


https://www.ledger.com/academy/blockchain/what-is-proof-of-stake

How will the network know if they're adding a malicious block?
Every single note before the block is propagated to the network will conduct a series of checks and this series of checks is very rigorous. If a check does not go through, then they reject the block and so basically, at the end of the day, the network will not allow malicious blocks to be added to the chain. That is why there is a financial incentive to play according to the rules.

9. Consensus Protocol – Proof of Work (PoW)





https://www.ledger.com/academy/blockchain/what-is-proof-of-work

Conflict between blocks: When a new block, free of malicious intent, is created, it is attached to the blockchain. The information may propagate at different speeds and does not reach all nodes immediately. Another block may be generated at the same time.

How do we proceed? We wait for another block to be added. Once that block is added, then we will see which of the two chains is longer: which chain basically adds a block first wins. Whichever chain has the most blocks will eventually win and replace the other chain. **The part of the network that has the highest hashing power will eventually generate the longest chain**. **Hashing power is measured by how many hashes can be checked per second.**

In a blockchain, the consensus protocol predicts that 50 of those with 51% of the hashing power, or more than 50% of the hashing power, will win.
When the conflict is resolved, the "losing" block is "detached" and becomes an "**orphan block**". Since the remuneration is contained within the block, the creator will lose the transaction. When a conflict appears, it is always better to wait for other blocks to be added, to make sure you add your own to the winning chain and don't lose the reward.

Here the link for training yourself:
https://tools.superdatascience.com/blockchain/hash/