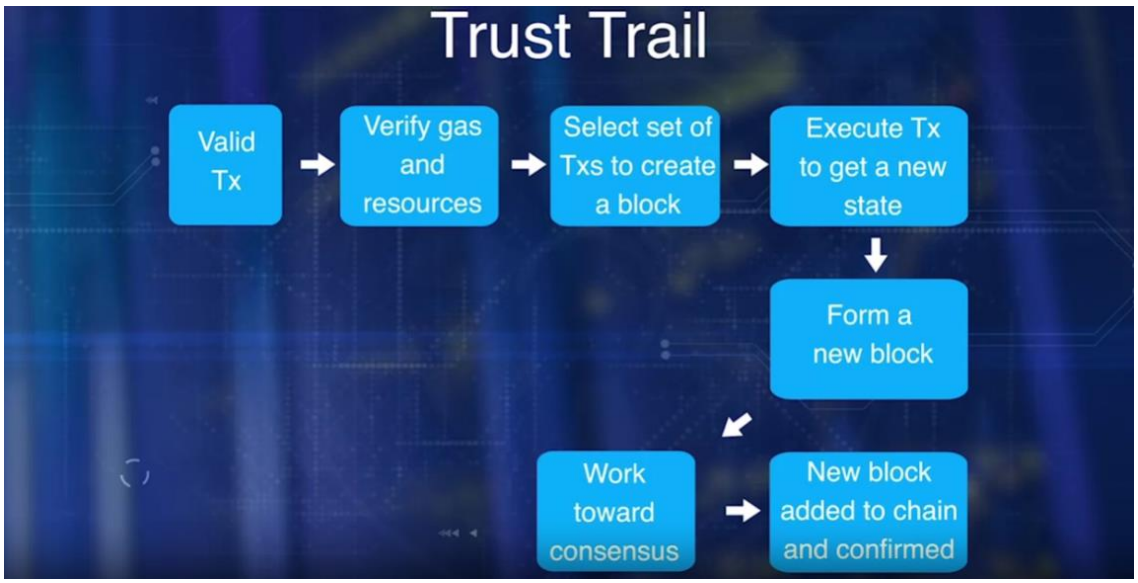
Trust Essentials

Trust in a decentralized blockchain is also about securing, validating, verifying, and making sure resources needed for transaction execution are available.

Establishing trust in a Blockchain:

- Secure chain using protocols
- Valid transactions & blocks
- Verify the availability of resources
- Executing and confirming transactions



Things required for validating transaction in Ethereum:

- Syntax
- Tx Signature
- Time Stamp
- Nonce
- Gas Limit
- Sender account balance
- Gas points & other resources
- Tx Signature hash

Merkle tree hash of the validated transactions is computed. This is in Ethereum.

All miners execute the transaction for either transfer, as well as for execution of smart contracts.

How Proof of work works?

Is there a method or a protocol to choose the next block? Yes, there is. It is called Proof of Work. *Proof of Work uses hashing.*

This is from the point of view of the miner.

- Compute the hash of the block header elements that is a fixed value, and a nonce that is a variable.
- If hash value is less than 2 par 128 for bitcoin, and less than function of difficulty for Ethereum, the puzzle has been solved.
- If it has not been solved, repeat the process after changing the nonce value.
- If the puzzle has been solved, broadcast the winning block that will be verified by other miners.
- The winner gets an incentive for creating the block.

Proof of Work is a consensus protocol used by bitcoin block chain and also by the current version of Ethereum.

The protocol may be the same, the implementations in these two block chains are different.

Many other approaches such as Proof of Stake, Proof of Elapsed Time have been proposed. This is a hotly debated area among the developers of blockchain.

Robustness

Trust us not only about executing regular operations correctly but also about managing exception satisfactory.

Robustness is the ability to satisfactorily manage exceptional situations.

An Exception:

What if more than one miner solves the consensus puzzle where it close in time to each other?

How this exception is handled?

- We start with the secure chain indicated by three blocks and we want to add to this chain.
- Here, two miners have solved the consensus puzzle very close to each other.
- Bitcoin protocol allows this chain split or two chains for the next cycle. One led by each of the competing blocks.
- The probability that the next block will happen at the same time in both these chains is extremely low.
- So, the winner of the next cycle for block creation consolidates one of the chains and that chain becomes the accepted chain.
- Now this chain is the longest and the valid main chain.
- The transaction in the other blocks is returned to the unconfirmed pool.

Ethereum handles more than one person we know by allowing Omar or Runner-Up blocks and allocating a small incentive for these Runner-Up blocks.

This incentive model helps in keeping the chains secure.

Another Exception:

What if more than one transaction references as input the same digital asset? This situation is called double spending.

We need a policy and an automatic deterministic way to handle this situation.

A policy for handling transaction and double spending in Bitcoin is to allow the first transaction that reference the digital asset and reject the rest of the transaction that reference the same digital asset.

In Ethereum, a combination of account number and a global nonce is used to address the doublet spending issue.

Every time a transaction is initiated by an account, a global nonce is included in the transaction. After that, the nonce is incremented.

Time stamp on the nonce in the transaction should be unique and verified to prevent any double use of digital asset.

FORK

Forks are just normal processes in an evolutionary path of the nascent technology enabling a blockchain.

Fork, hard fork and soft fork, are most common phrases uttered in the context of a blockchain.

Soft fork and hard fork in the blockchain word are like the release of software patches, and new versions of operating systems respectively.

Forks are mechanisms that add to the robustness of the blockchain framework.

If robustness and trust is about managing exceptional situations, hard forks and soft forks are indeed at the front and centre.

We discussed change split in the last topic, that is a minor perturbation in the chain. Such situation is handled as a naturally expected occurrence within the block chain. On the other hand, occasionally, a minor process adjustment has to be carried out typically by **bootstrapping a new software to the already running processes**. This is **soft fork**.

Hard fork implies a major change in the protocol.