Blockchain 101 - Blocktrain.info

✅ Complete courses on youtube: https://www.youtube.com/blocktrain

Inception of a Revolution
Why do we need a decentralized system or a Blockchain?
Double Spending Problem
Security Pillars of Blockchain
- Cryptography
- Symmetric Cryptography :
- Asymmetric cryptography or Public Key Cryptography
- Cryptography in Blockchain

Hashing
- Transaction Hash
- Block Hash
- Blockchain Hash

Blockchain Anatomy
Data
Blockchain Trilemma
- Decentralization
- Security
- Scalability
- The Trade Offs

Consensus Protocol
- 1. Proof Of Work (PoW)
- 2. Proof of Stake (PoS)
- 3. Proof of History (PoH)

Blockchain Layers
- Layer 0
- Layer 1
- Layer 2
- Layer 3

Crypto Wallets
- Public Key
- Private Key
- Hot Wallet

Cold Wallet
EVM vs Non-EVM
Resources
Watch

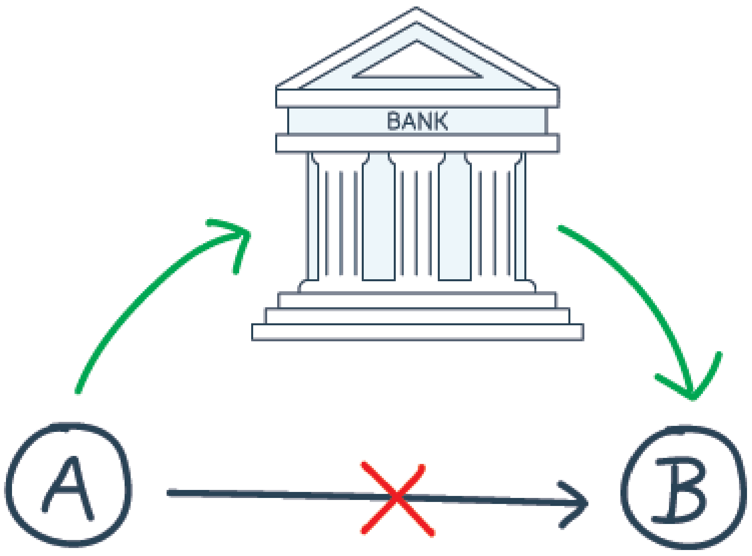
Inception of a Revolution

In 2008 the United States of America witnessed one of its biggest financial crises that impacted the global market. The primary reason being, unorthodox practices of banks that lead to a cascading effect on the overall financial system. Few Banks were on the verge of bankruptcy, they blocked people from withdrawing money which reduced the purchasing power in the market, citizens went on to protests as they were denied access to their own money. It’s said that in the midst of every crisis, lies great opportunity.

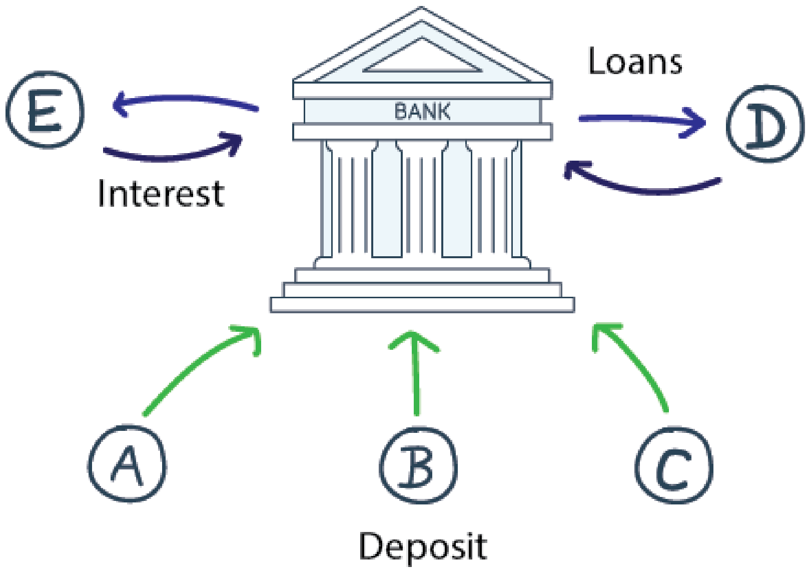
On 1 November 2008, an anonymous individual/group named Satoshi Nakamoto wrote in an email “I’ve been working on a new electronic cash system that’s fully peer-to-peer, with no trusted third party. The paper is available at bitcoin.org/bitcoin.pdf." And that is how bitcoin was born to bring revolution. The Bitcoin whitepaper states that bitcoin is an electronic cash that is peer-to-peer and works on cryptographic proof eliminating the need of third parties.

Why do we need a decentralized system or a Blockchain?

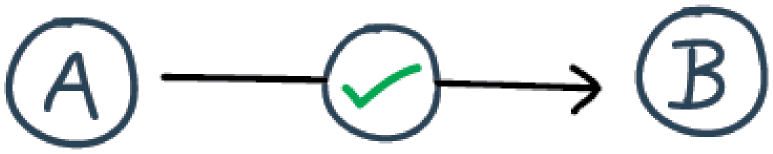
- Let's say A wants to transfer $500 to B, how will he do it? Via Bank? He cannot transfer the money directly, right? Any monetary transaction requires a mediator that establishes trust which is a bank.



- Basic Bank Functioning : For every transaction that happens through banks it maintains a book of record called ledger. It mentions that $500 was debited from A's account and credited to B's account at this particular time. The ledger is kept private.
- A,B & C want to deposit money in the bank because they trust the bank. Now, E & D wants to loan money. Bank gives them a loan and charges interest on the borrowed amount.



- It's very clear, for every operation that we do through the bank we need to trust it to keep our money safe. As discussed above regarding the 2008 crisis. Banks miss handled people’s money and it crushed people's trust in the banking system.
- And that’s when Satoshi Nakamoto proposed the idea to eliminate banks from the system, making the ledger public and transactions backed by cryptographic consensus.
- Now A can transfer money directly to B, as the ledger is public and all the transactions are visible & secured in blocks that link together making it tamper proof.



There's specific mention about the Double Spending Problem in the Bitcoin Whitepaper, let's understand what it is and how blockchain is the solution.

Double Spending Problem

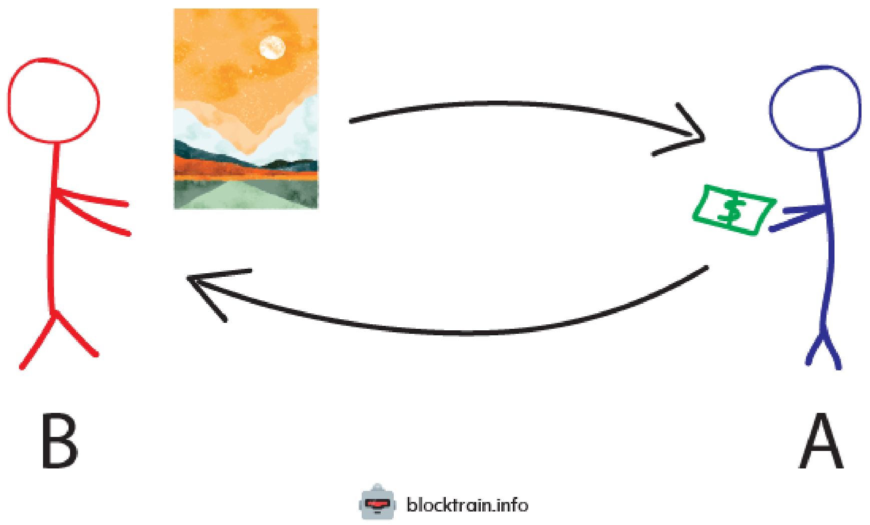
Scenario 1

➜ B wants to sell a physical artwork to A for $100.

➜ A pays him $100 in cash and owns the artwork.

Can B sell the same artwork to another person? No he cannot.

What if he can?



Scenario 2

➜ B has a digital artwork.

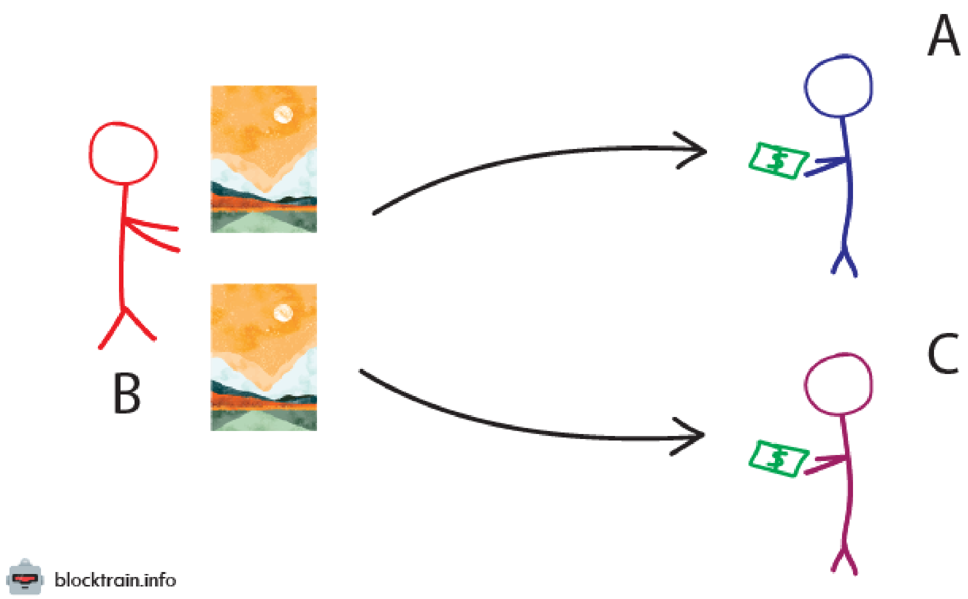
➜ He sells it to A for $100.

➜ He makes a copy of it and sells it to C

➜ Here, A & C both spend $100's for the same artwork.

This is called Double Spending

Let's see how blockchain solves the problem

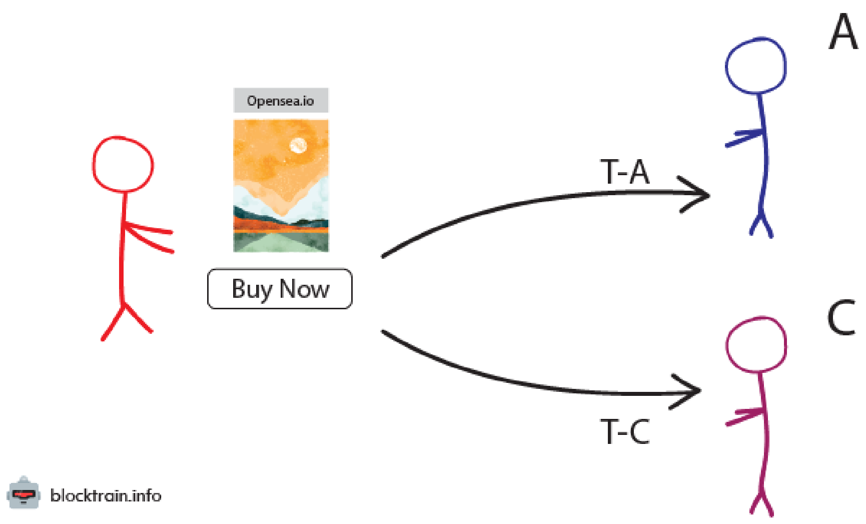


Smart Move

➜ A & C are smarter than B, they ask B to mint the artwork on a blockchain as NFT.

➜ Both of them try to buy at the same time.

➜ Let's say transaction of A was T-A and for C was T-C

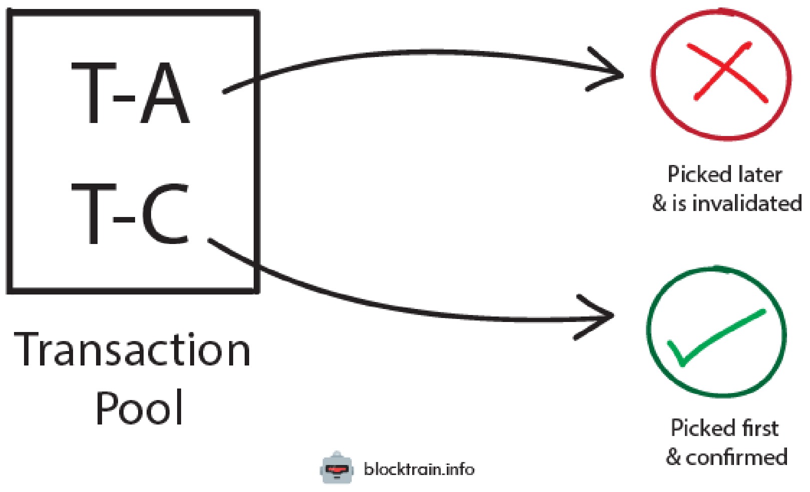


Transaction

➜ Both the transactions are added to a pool of transactions.

➜ The miner picks T-C first and achieves finality

➜ C's transaction is approved and the NFT is transferred to him, while A's transaction will be invalidated.



What if two different miners pick T-A & T-C at the same time?

➜ Then, the transaction that is confirmed first will be approved and added to a block,

➜ If T-A is approved first, T-C will be rejected.

What if both the transactions are approved at the same time?

➜ Then, a race starts

➜ Where the transaction with maximum confirmations is accepted and the other one is removed from the block.

Security Pillars of Blockchain

Blockchain has two levels of security

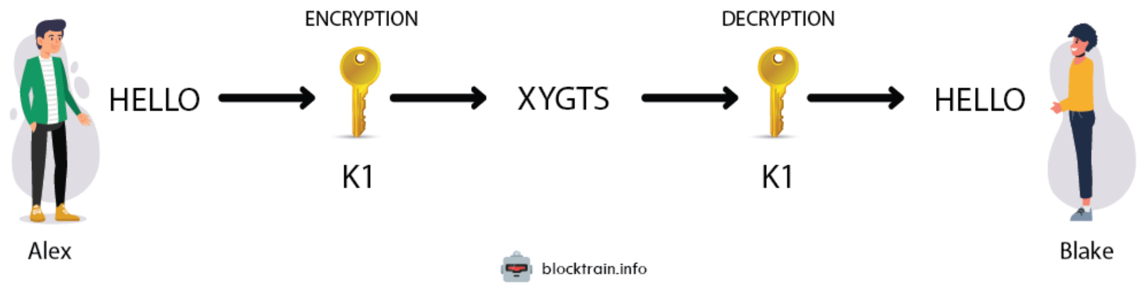
- *External Level* : Cryptography (Works between node to node & in Wallets)
- *Internal Level* : Hashing (Works in blocks)

Cryptography

Before we understand how cryptography is implemented in blockchain let's understand its working in general.

Symmetric Cryptography :

Alex wants to send some confidential message to Blake. In symmetric cryptography both of them will have a common key, let's call it K1. Alex encrypts the message using the key (K1) and then Bob decrypts the message using the same key (K1). The problem with this method is, if the key is compromised anyone can read the message.

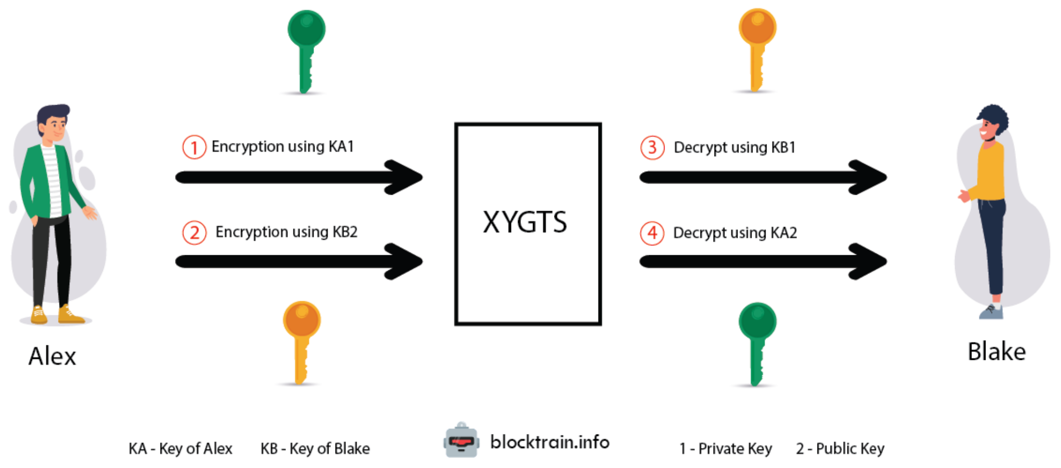


Asymmetric cryptography or Public Key Cryptography

In this method, each node has two keys, one is a public key and the other one is a private key. In the above case, Alex and Blake both will have two keys each. Whenever there's a link between them, they'll share their public keys to each other. The encryption-decryption happens as follows.

KA - Key of A, KB -Key of B, 1- Private Key, 2- Public Key

1. When Alex sends a message to Blake, It's first encrypted using his private key (KA1)
2. Proceeded by Blake's Public Key (KB2)
3. Then, Blake decrypts the first layer of encryption using his private key (KB1).
4. Blake gets the original message when it's decrypted using A's public key (KA2) in the last step.



Properties of Asymmetric Cryptography :

- It is not possible to guess the Private key from the Public Key.
- However, they are mathematically linked in such a way that anything encrypted using either a Public or Private key can only be decrypted using only these two keys.
- Two layers of encryption ensures authenticity of the message. In the last step Blake is sure that the message was from Alex as it could be decrypted using Alex public key only.
- End-to-End encryption in WhatsApp works the same way.

Cryptography in Blockchain

Symmetric Cryptography was one of the earliest implementations in blockchain which is no longer used.

Asymmetric Cryptography in Blockchain, works on two level

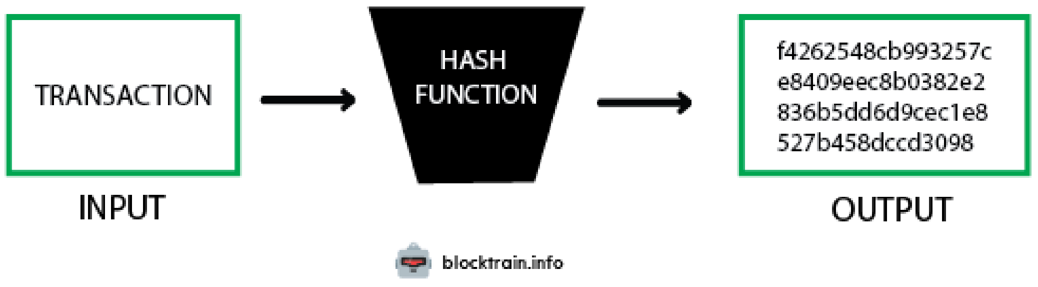
1. Wallets:
    - If you ever used a crypto wallet like Metamask or Phantom. You must have heard about the term “seed phrase''.
    - It’s a 12, 18 or 24 word long phrase that you get when you create a new wallet. This is the private key.
    - The address that we use to send or receive crypto is the public key.

- The wallet address and the seed phrase both are linked together but a seed phrase can never be guessed from a wallet address.
- If Alex sends Blake 1BTC on his address, Blake must have his private key to authenticate the ownership of that 1BTC.
- If he loses his private key, he'll never be able to access his wallet.
- For the same reason it's always recommended to save your private key.

2. Node to Node Communication or Peer to Peer Communication
    - Any end device (miner) connected to a blockchain network is called a node.
    - Consider 2 nodes A and B. Both have their own pair of Private and Public keys.
    - When A&B want to communicate, the encryption-decryption protocol starts, which works the same way as we saw in the case of Alex and Blake.

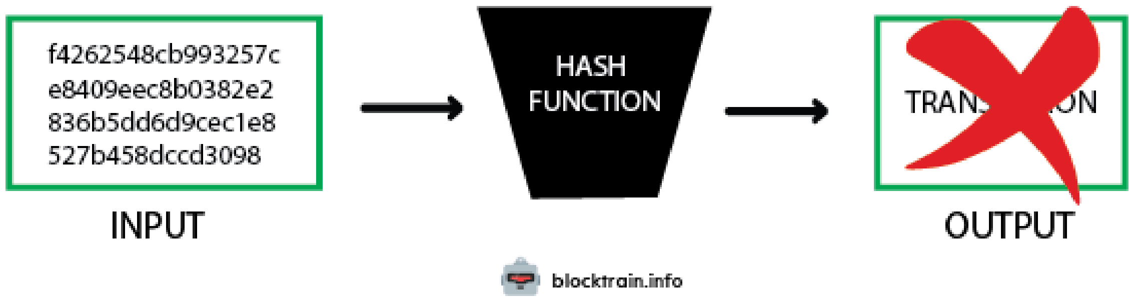
Hashing

Hashing is a function that takes arbitrary length of data as input & returns a value of fixed size, the output is called a hash.



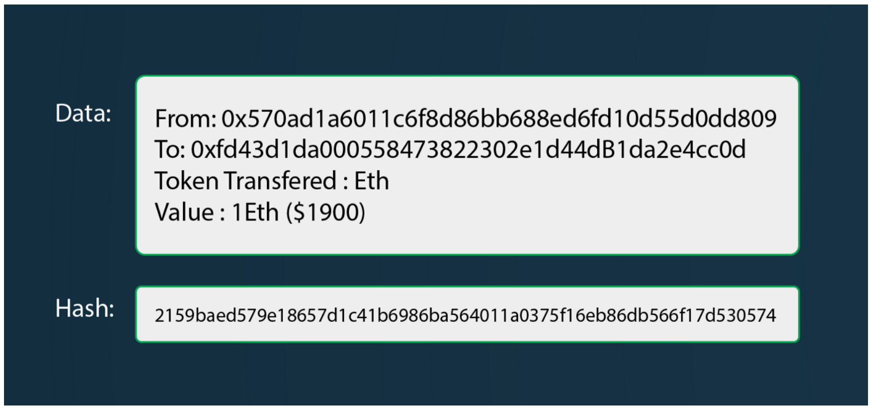
- The size of the hash depends on the algorithm used. Some Common Hashing Algorithms MD5 (128 bit), SHA-1 (168 Bits) & SHA-2 Hashing Family have 224, 256, 384 or 512 bit size.
- Bitcoin uses the SHA-256 algorithm where SHA stands for secure Hashing Algorithm and 256 is the bit size of the hash. A SHA256 hash is a hexadecimal value containing 64 characters (0–9,A-R).

- Hashing is a one way function that means the original data cannot be retrieved from a hash unlike cryptography where it is possible to decrypt the key to get the original message. So one should not confuse hashing with cryptography



Transaction Hash

Each transaction in a blockchain is identified by a hash. This hash is like a fingerprint for the transaction. You can look into the details of a transaction, just by searching it's hash on a blockchain explorer.



Data:

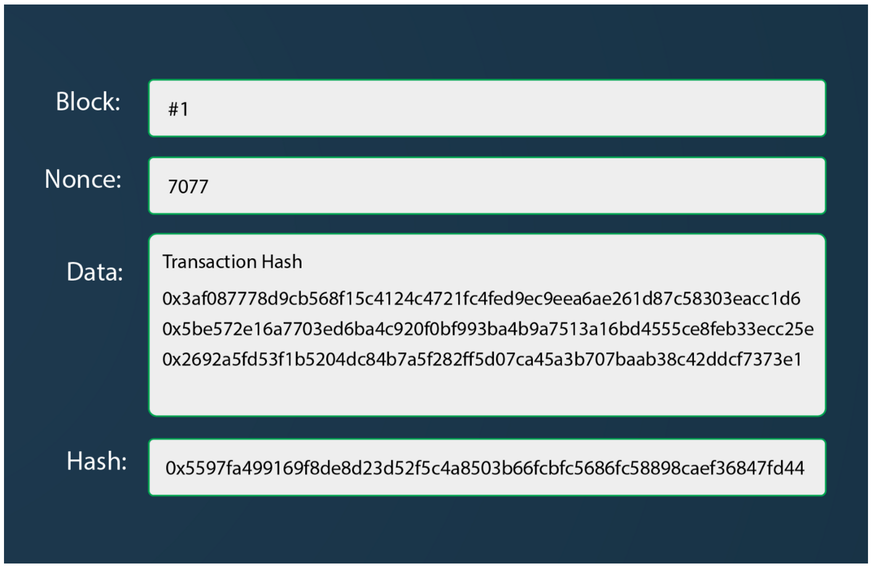
From: 0x570ad1a6011c6f8d86bb688ed6fd10d55d0dd809
To: 0xfd43d1da000558473822302e1d44dB1da2e4cc0d
Token Transfered : Eth
Value : 1Eth ($1900)

Hash:

2159baed579e18657d1c41b6986ba564011a0375f16eb86db566f17d530574

Block Hash

As we know the transactions are stored in blocks in a form of hash, the block is itself identified by a hash. And the block is valid when the hash is validated by a miner using Nonce.



Block: #1

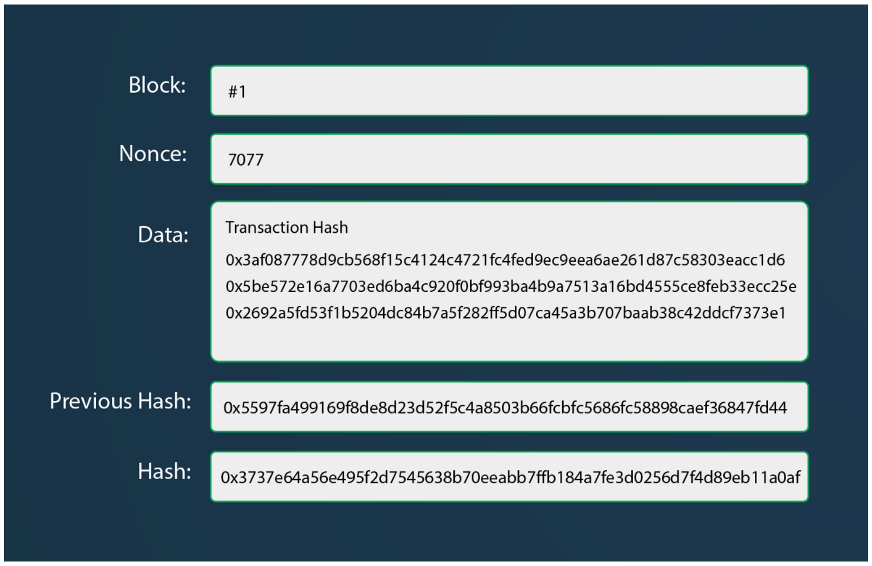
Nonce: 7077

Data: Transaction Hash
0x3af087778d9cb568f15c4124c4721fc4fed9ec9eea6ae261d87c58303eacc1d6
0x5be572e16a7703ed6ba4c920f0bf993ba4b9a7513a16bd4555ce8feb33ecc25e
0x2692a5fd53f1b5204dc84b7a5f282ff5d07ca45a3b707baab38c42ddcf7373e1

Hash: 0x5597fa499169f8de8d23d52f5c4a8503b66fcbfc5686fc58898caef36847fd44

Blockchain Hash

When a block is created, it is linked to the blockchain using the hash of the preceding block, you can imagine it like a linked list. The first block in a blockchain is called the genesis block, the bitcoin genesis block was mined on Jan 3, 2009.



Block: #1

Nonce: 7077

Data: Transaction Hash
0x3af087778d9cb568f15c4124c4721fc4fed9ec9eea6ae261d87c58303eacc1d6
0x5be572e16a7703ed6ba4c920f0bf993ba4b9a7513a16bd4555ce8feb33ecc25e
0x2692a5fd53f1b5204dc84b7a5f282ff5d07ca45a3b707baab38c42ddcf7373e1

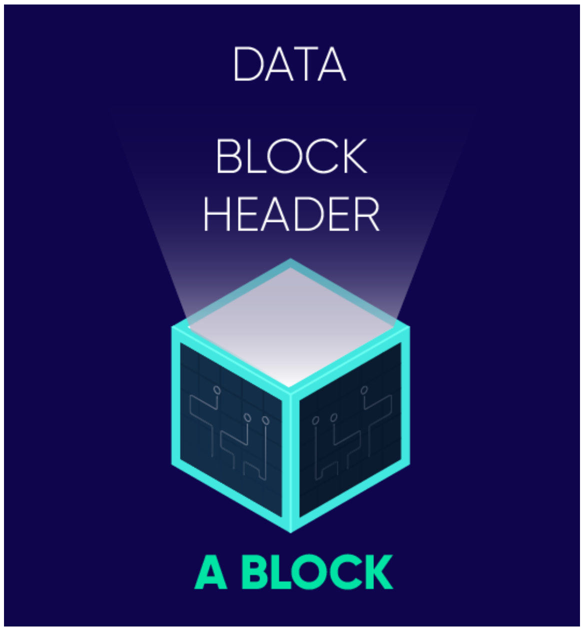
Previous Hash: 0x5597fa499169f8de8d23d52f5c4a8503b66fcbfc5686fc58898caef36847fd44

Hash: 0x3737e64a56e495f2d7545638b70eeabb7ffb184a7fe3d0256d7f4d89eb11a0af

Blockchain Anatomy

Two Components

1. Data
2. Block header

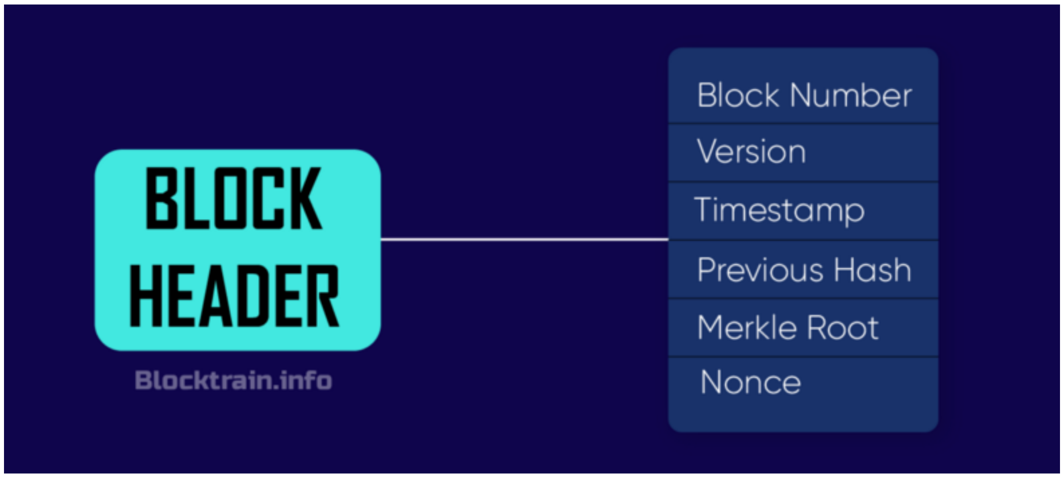


Data

The Data part of the block consists of information regarding the transaction like From, To and money. These are the same transaction details we see on a blockchain explorer like etherscan or bscscan

A Block Header contain

1. Block Number
2. Version
3. Timestamp
4. Previous Hash
5. Merkle Root
6. Nonce



Block Number

It's the identification number of the block, given to it when it's added to the blockchain. You can look in a block if you know it's block number.

Version

It stores the current version of the chain when the block was created. A different version might have more elements in the block, so it's important to keep track of the version to update the components of the block.

Timestamp

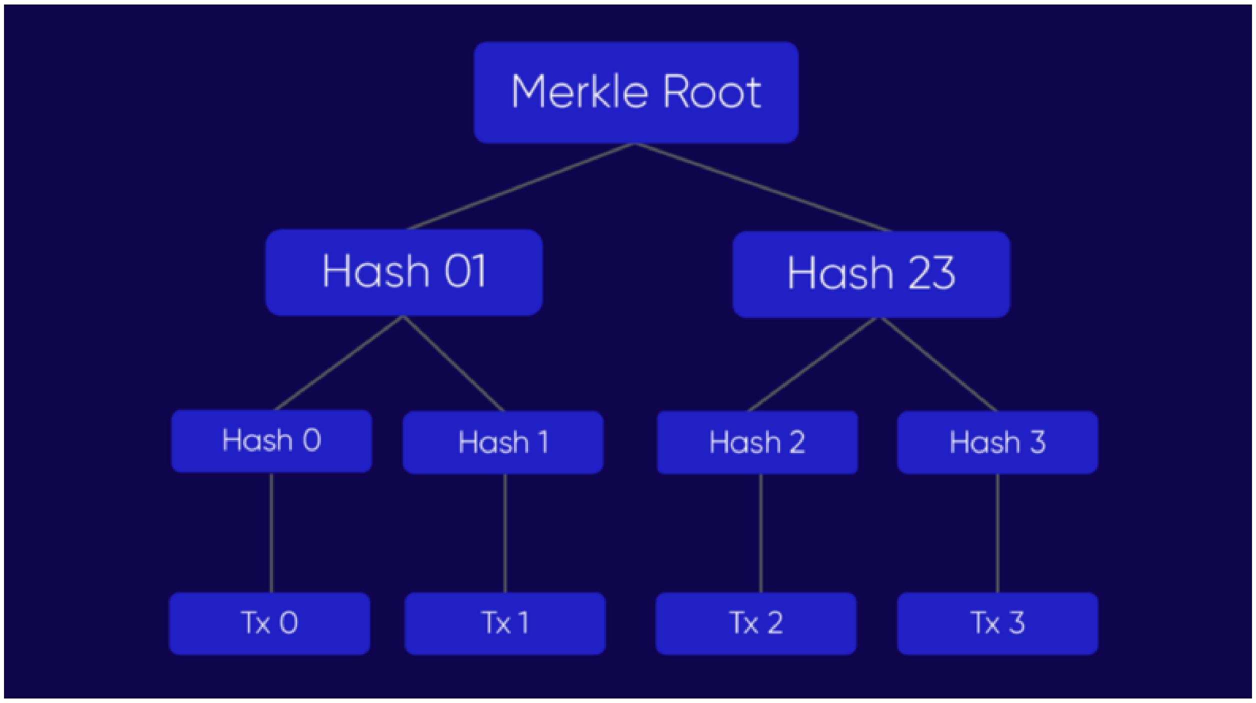
It's the time when the transactions are approved and the block was created. It's in a unix time format and it helps in mining during nonce determination.

Previous Hash

It's the hash of the previous block's header.

Merkle Root

Also called Binary Hash Tree, it is a data structure that summarizes all the transactions in a block.



Nonce

Number Used Only Once (NONCE), is a number that help miners solve the mathematical problem of generating a hash. It starts from 0 until the number where the targeted hash is achieved. Let’s understand it with a simple equation

You are given,

X+Y=10

And asked to find the value of Y but you don’t know X as well, can you find Y? No, you can’t.

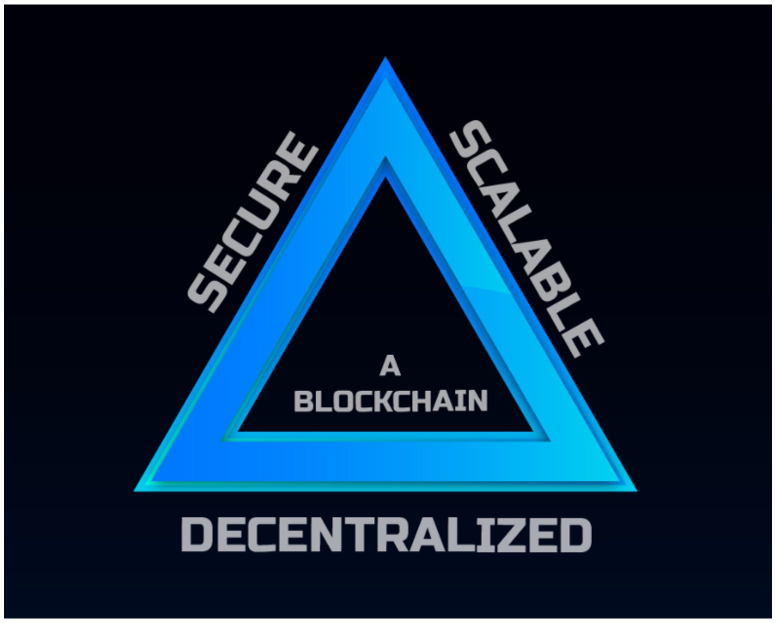
Now, if X is given as 4 then Y is 6, so

Let’s say X is the Nonce, Y is the hash that the miner needs to find and 10 is the targeted hash value. When X is known we can easily find Y, same way Nonce(X) helps find the hash(Y) that’s valid (10).

Blockchain Trilemma

Theoretically an ideal blockchain, is built on three foundational pillars i.e Decentralization, Security and Scalability but according to Vitalik Buterin (Founder of Ethereum) in the current structure of Blockchain, if a network has to grow it cannot do so without compromising any one of the pillars and this is called as Blockchain Trilemma.

It's more like a general problem for the Blockchain developers to build a network in which all the three pillars compliment each other making it an ideal blockchain. Let's understand the terms



Decentralization

The primary reason for the existence of Blockchain technology is the need of a system that is not controlled by a single authority and eliminates the 3rd party dependency.

The Bitcoin Network is decentralized but it's not scalable because of the block size, block time and power consumption.

Security

Security & Decentralization are the two sides of the same coin. If a blockchain is completely decentralized it means it's fundamentally secured. Security means the data & transactions are immutable and the network is safe from attacks.

Scalability

Scalability of a blockchain is its capability to perform ideally and handle the growing demand. For example, Ethereum gas fee rises exponentially when transactions are high. Solana is highly scalable but it’s considered more centralized which is evident by the fact that it went down 4 times in a month.

The Trade Offs

If a blockchain network is completely decentralized, it compliments security while compromises on scalability.

If a network is scalable then it’s more likely to be centralized to give more throughputs which indeed makes it less secure. And security should never be compromised for anything.

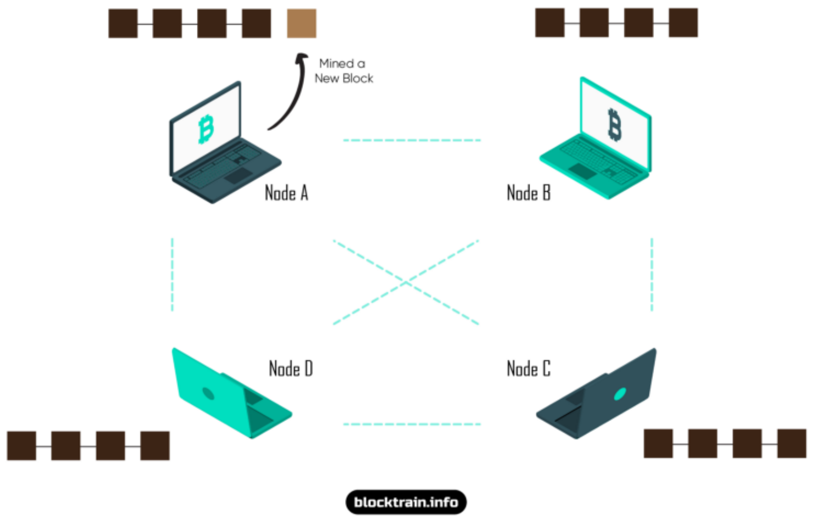
Solution?

In the past 2–3 years developers have approached the problem in many ways, like implementing Blockchain Layers (Layer 1, 2,3), Consensus Mechanism (Proof of Stake, Proof of History), Sharding or Beacon Chain

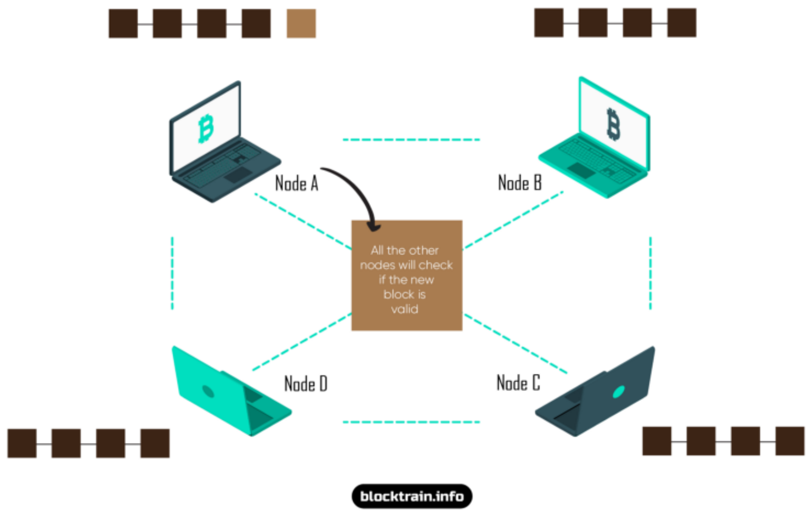
Consensus Protocol

The general definition of consensus is, coming to an agreement and protocol are a set of rules that need to be followed to perform actions. So in blockchain, Consensus Protocols are fixed rules in which all the nodes in the network agree to the new state(copy) of the blockchain.

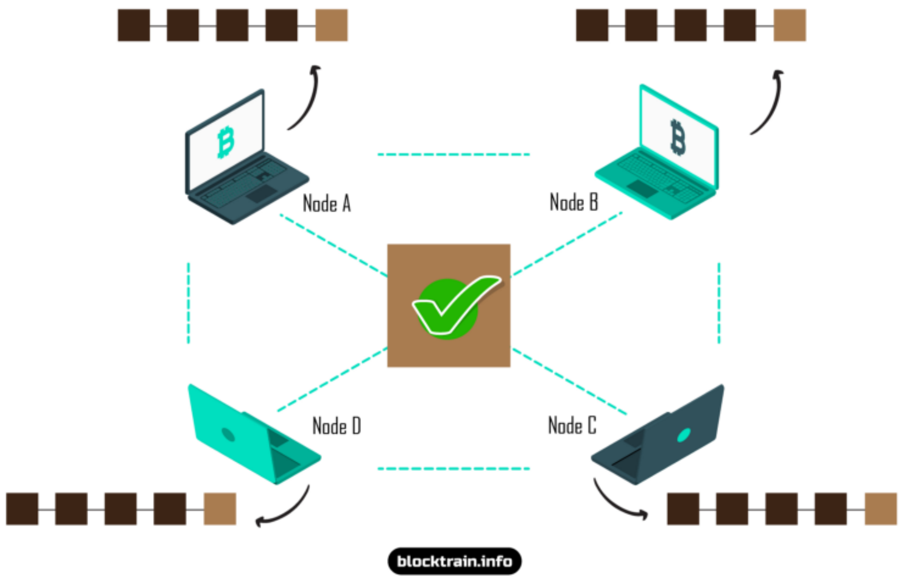
Let’s say there are 4 nodes in a network having a copy of the blockchain consisting of 4 blocks. Now, node A mines a new block and adds it to the copy of the blockchain that it holds. As we know, blockchain is a distributed ledger hence all the nodes should have the same copy of the blockchain.



All the remaining nodes will follow a protocol and come to a consensus to verify whether it's a valid block or not.



If the block is valid, it will be added to the chain and the miner will be rewarded some tokens of that blockchain.



Basically that's how a consensus protocol works in blockchain. Different blockchain's have different consensus protocol

1. Proof Of Work (PoW)

Yes, it means what it sounds like, showing the proof that you have done the work. As we know mining is basically solving a complex mathematical problem that requires high computational power. At a time when many miners compete to mine a block, the node that wins the race has a proof of work of spending energy and resources and hence it's rewarded when the block is verified and added to the chain. If the node tries to perform malicious activity, the block will not be verified by other nodes in the network and the miner doesn't get the reward, which is a total waste of its resources and time. Bitcoin works on PoW, while Ethereum is on its way to Proof of Stake. Proof Of Work requires a lot of power as many miners try to compete to mine a block, Proof Of Stake is a better alternative.

2. Proof of Stake (PoS)

In Proof of Stake, instead of miners competing in a race to mine a block, they are asked to stake (deposit) some minimum tokens of that blockchain to become validators (Miners in PoS are called as Validators). The validators are chosen randomly taking into consideration how many tokens they have staked or for how long they have been staking. If a validator performs any malicious activity he's penalized from the stake. It is a more energy efficient protocol as not all the validators compete against each other which lowers transaction cost and increases throughput. This is one of the primary reasons for Ethereum to migrate to PoS.

3. Proof of History (PoH)

Proof of History is an extension of proof of Stake, developed by Solana. And it's one of the primary reasons why Solana is very fast, having the block time of 400 millisecond compared to Ethereum (1Sec) & BTC (10Mins). PoH introduces a variable of time in the block, it's showing the proof that a historical event had occurred. In other consensus protocols the validators need to talk to each other to agree on when a block was added and how much time has passed. Whereas in Solana it has its own clock coded in a SHA256 algorithm, the validators keep mining blocks and they are sequentially added to the chain.

Blockchain Layers

Layer 0

It's the underlying infrastructure layer that forms the basic blockchain network consisting of distributed nodes forming the peer-to-peer architecture. Layer 0 enables interoperability (cross-chain communication), blockchain protocols that are built on the same layer 0 can easily transfer tokens and data.

Example : PolkaDot, Avalanche, Cosmos

Layer 1

Prominent blockchains like Bitcoin. Ethereum, Solana, Near, etc are the layer 1 chains. These are the execution protocols that provide the environment for transactions, handling cryptographic algorithms, data, consensus and tokenomics. Native tokens of the chain are the medium to interact with smart contracts, access resources, and mint NFT.

Example : Solana, Ethereum, Bitcoin, Tezos, Near

Layer 2

Built on top of layer 1 chains, L2 solves the scalability problem of the blockchain trilemma by improvising on transaction speed (faster finality) & throughputs (higher transactions per second). Layer 1 handles security, data & consensus while Layer 2 handles transactions and regularly communicates with layer1. It takes away the transaction burden from layer 1 while taking the advantages of layer1's security and decentralization.

Example : Polygon Matic L2 on Ethereum

Layer 3

This is the utility layer where all the decentralized applications are built like DeFi, games & wallets. It gives abstraction where the user is only

concerned about the application functionality and not the core

Protocols.

It opens the door for adoption & interoperability. For example, Trust Wallet & Metamask can be connected to multiple chains. Example : Aave, Curve Finance, DYDX Exchange.

Crypto Wallets

Crypto Wallet is a container for digital assets (Cryptocurrencies & NFT). It’s like a tool that helps us manage our assets in one place and allows us to do transactions. It is a gateway to blockchain. Cryptocurrencies never leave the native blockchain, they are just transferred from one wallet to another just like transfer of water from one bucket to another.

A Wallet has two components :

1. Public Key
2. Private Key

Public Key

The address that you use to send or receive the assets is the public key. For different tokens on the same chain, there is one public key. Whereas for multi chain wallets there are multiple public keys for each chain. That’s why you have the same wallet address for DAI and ETH in Metamask, whereas different addresses for Solana and Eth.

Private Key

The private key is a 12, 18 or 24 long phrase that we are asked to save when we create a new wallet. This phrase is needed to verify the ownership of the assets in the wallet, if you lose your phrase you lose the assets in the wallet.

Types of Wallet

1. Hot Wallet
2. Cold Wallet

Hot Wallet

The wallet that is connected to the internet is called a hot wallet. Like Metamask Phantom or Trust Wallet. It’s also called Software Wallet that can be either web based/extension, desktop wallet or mobile app wallet.

METAMASK

Phantom

TRUST WALLET

Cold Wallet

It’s a hardware device that stores the assets along with private and the public keys but is never connected to the internet. It’s like a store of value, people who hold for long term prefer using a ledger as it’s safe from hackers.

EVM vs Non-EVM

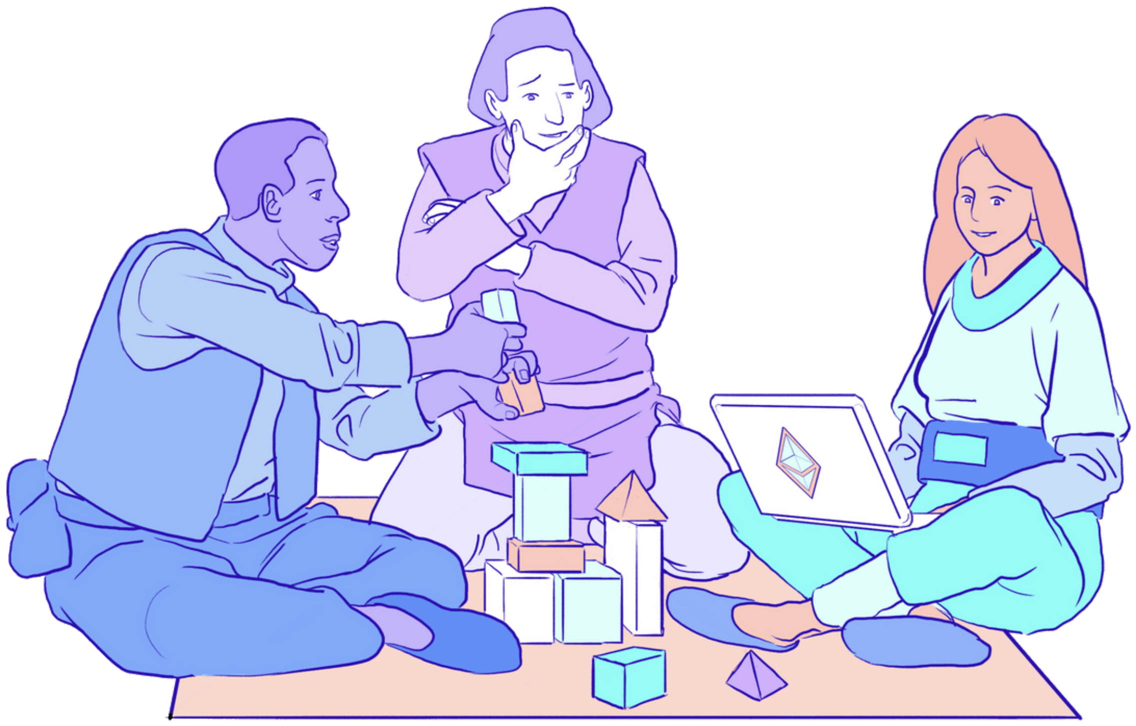
- The Ethereum Virtual Machine is the software platform that developers can use to create decentralized applications (DApps) on Ethereum. This virtual machine is where all Ethereum accounts and smart contracts live.
- The role of the EVM is to deploy a number of extra functionalities to the Blockchain to ensure users face limited issues on the distributed ledger. Every Ethereum node runs on the EVM to maintain consensus across the blockchain.
- EVM works like a large decentralized or master computer to complete all types of tasks on the blockchain.
- The EVM is Turing complete, meaning that if asked, it will find an answer. Users can write smart contracts in Solidity, Ethereum’s programming language, and send it to the EVM to interpret and execute. The Ethereum protocol works as the consensus architecture for these contracts.
- Every project is working towards making transactions faster, adding multi-chain functionality, and more. Polkadot has added Moonbeam, Near launches Aurora, and

soon (announced but not released), Evmos will be live on Cosmos and Neon will be implemented on Solana, eg: https://aurora.dev/

- Solana is an example of a non-EVM project.
- Cardano is now EVM compatible!?
- https://coinwut.com/cardano-evm-compatible/
- https://medium.com/coinmonks/cardano-determinism-e-utxo-ledger-model-ef183d1e88ce

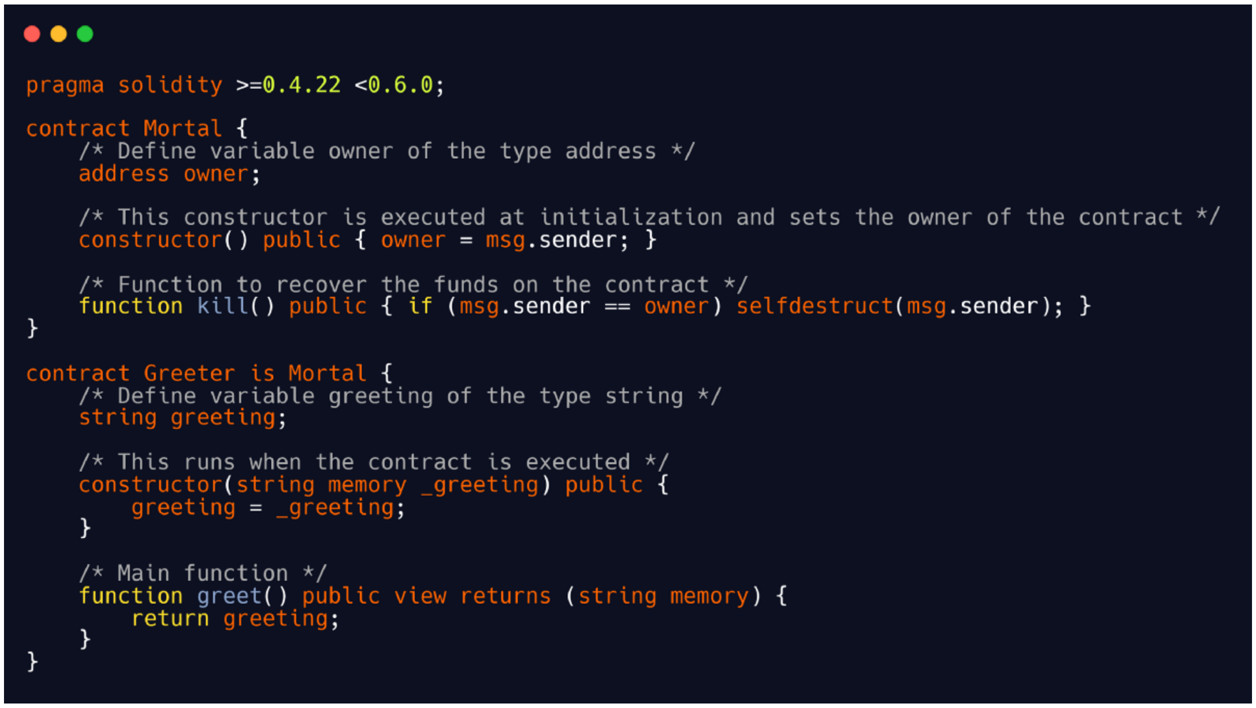
Resources

- Ethereum Virtual Machine (EVM) | ethereum.org
- An introduction to the Ethereum virtual machine and how it relates to state, transactions, and smart contracts.
- ethereum.org



The Ethereum Virtual Machine — How does it work?

If you've tried developing a smart contract on the Ethereum blockchain, or have been in the space for a while, you might have come across the term " EVM", short for Ethereum Virtual M achine. Virtual machines are essentially creating a level of abstraction between the executing code and the executing machine.



```
pragma solidity >=0.4.22 <0.6.0;

contract Mortal {
    /* Define variable owner of the type address */
    address owner;

    /* This constructor is executed at initialization and sets the owner of the contract */
    constructor() public { owner = msg.sender; }

    /* Function to recover the funds on the contract */
    function kill() public { if (msg.sender == owner) selfdestruct(msg.sender); }
}

contract Greeter is Mortal {
    /* Define variable greeting of the type string */
    string greeting;

    /* This runs when the contract is executed */
    constructor(string memory _greeting) public {
        greeting = _greeting;
    }

    /* Main function */
    function greet() public view returns (string memory) {
        return greeting;
    }
}
```

- Bridging: https://apys.medium.com/what-is-an-evm-and-why-do-you-need-bridges-fc80936cf55a
- https://www.theblockcrypto.com/data/scaling-solutions/non-evm-chains-stats

Watch

- https://www.youtube.com/watch?v=1sn3A62Fj_A