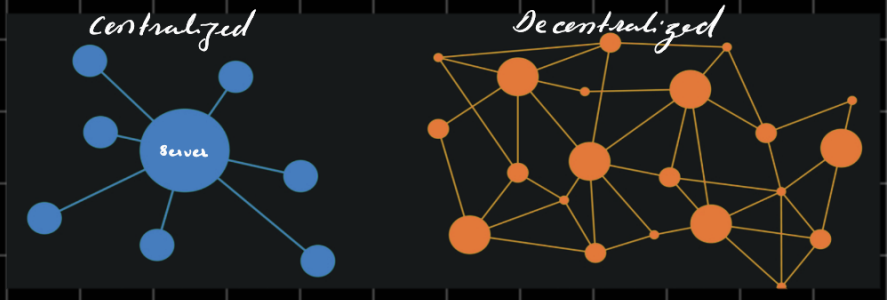
Block chain →

Block chain is a destributed decentralised system that enable secure and transparent transactions and information sharing.

A digital database that is distributed among the node of a peer-to-peer network.



Peer to Peer →

- Group of devices at same heirarchy.
- Any system can become a server and a client.
- Equal power and rights.
- Destributed network.
- No central authority.

Some key aspects of block chain technology →

- Decentralization →

  Block chain operate on decentralized network of compute, known as nodes, which maintains and evaluate/validate the block chain.

- Transparency and security →

  The data stored on the block chain is transparent and available to all the participants in the network. Each information is maintained and validated by advance cryptographic technique.

- Consensus Mechanism →

  To achive agreement on block chain, various machinism are used Proof of work (POW), Proof of Stake (POS), Delegation Proof of Stake (DPOS).

These mechanisms ensures that the majority of participants agree on the validity of transactions and the order in which they are added to the block chain.

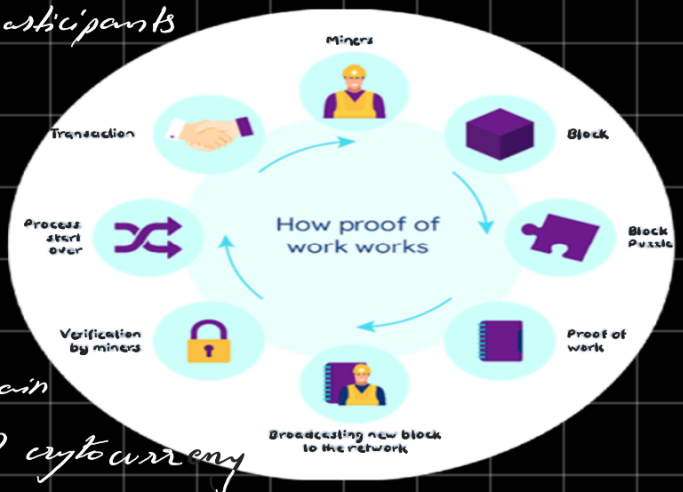
- Smart Contracts →

Blockchain technology supports programmable contracts called smart contracts. Smart contracts are self-executing agreements with predefined rules and conditions. Once the conditions are met, the contract automatically executes without the need for intermediaries, ensuring transparency and efficiency.

Consensus →

① POW (Proof of Work)

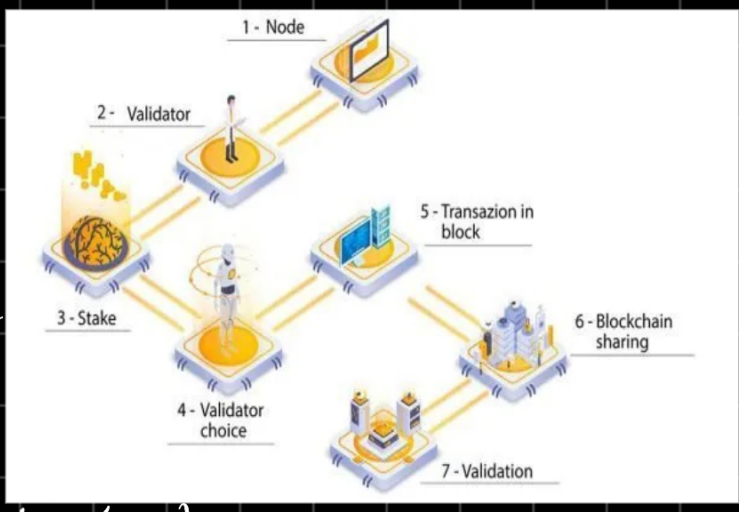
- Original Pow blockchain consensus algorithm "Satoshi Nakamoto" bitcoin blockchain it works by requiring network participants solve complex mathematical problem which takes a signifant amount of computational power.



- The first participant to solve the problem gets to add the next block of transaction of the block chain and is rewarded with a predefined amount of cryptocurrency this algorithm requires a large amount of computation power that makes it slow and expensive.

- Miner rewarded if block added in the blockchain and verified by others.
- For verification the miner broadcasts its solution of the complex hashing.
- No one else gets any thing other than the successful miner.
- Reward is pre-defined
- One can take over the blockchain by gain more than 50% of the network's computational power.

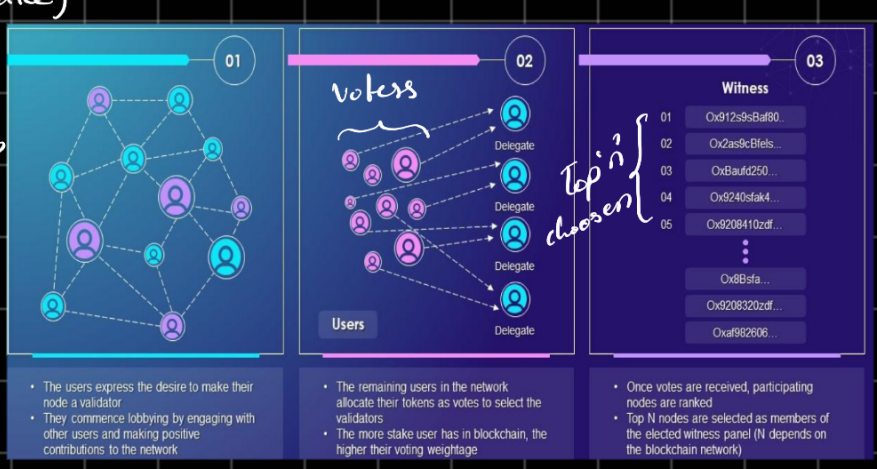
② POS (Proof of Stake)



- POS was created as an alternavite of POW.
- While POS mechanism requires miners to solve cryptographic puzzels, POS requires validators to hold and stake token's for the privilage of easing rewards.
- POS is less risky regarding the potential for an attack on the network (due to its structure).
- The next block writes on the block chain is selected at random with higher odds being assigned to the nodes, with larger state positions.

- Block creator is choosen by the user stake algorithm.
- No Reward for the block creator he takes a transaction fee.
- Uses a lottery system, chances of miner staking higher coins increases for the validation process.
- If the validator does not validates correctly then the staked coin will be taken.
- The lottery system is also coupled with parameters likes → coin age and randomize block selection.

③ DPoS (Delegated Proof of Stake)



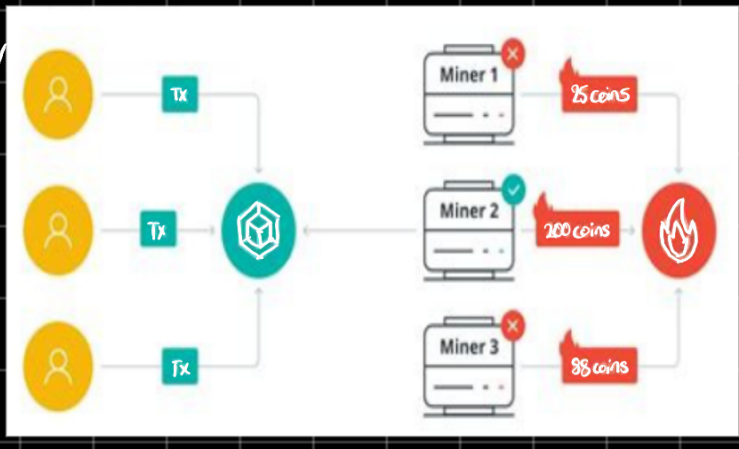
- Envolves delegated Proof of stake
- No physical transfer of coins from the walet.
- Delegates are selected using voting system. ex → Tron, Etherium

- Advantages —
  Accessable, Democrative, Scalable
- Disadvantages —
  Centralization, Risky

- By staking coins the user gets the right to vote, he can vote multiple representative.
- Top of the few delegates are selected for varification and addy of block in the block chain.
- The delegates will be given chance in a sequencial manner.
- Delegate need to return certain percentage of the rewards to the voters.
- The process of voting is an decentralized manner and after the selection of delegate the process becomes centralized, quick and power efficient as no sort of race is taking place.
- The vote casted can be changed at any point by the voter to represent another delegate as per their wish.

④ POB (Proof of Burn)

- Consences algorithm that requires user to destroy/burn a certain amount of coins in order to participate in the network.
- More the user burn the higher the chances of being a evaluater.
- Validator receive rewards in the form of transaction fee and newly mined coin.



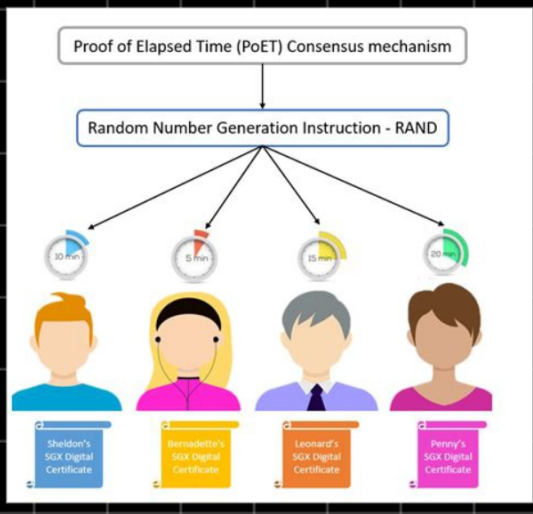
- Advantages
  - Reduces the risk of centralization.
  - Eleminates the problem of nothing at stake.
- Disadvantage
  - Causes deflationary effects on coins supply (as the burned coins are now gone from the circulation).

- The miner need to burn / destroy their coin, higher they burn the chances increases of getting selected for adding a block in the blockchain.
- They will earn reward + transaction fee.

⑤ PoC (Proof of Capacity)

- Evaluator are supposed to invest their hard disk space insted to staking coin.
- PoC authentication system imployes spare space on participants device to store solutions to cryto currency hashing problem.
- ex – Signum, Chia

⑥ PoET (Proof of Elapsed Time)



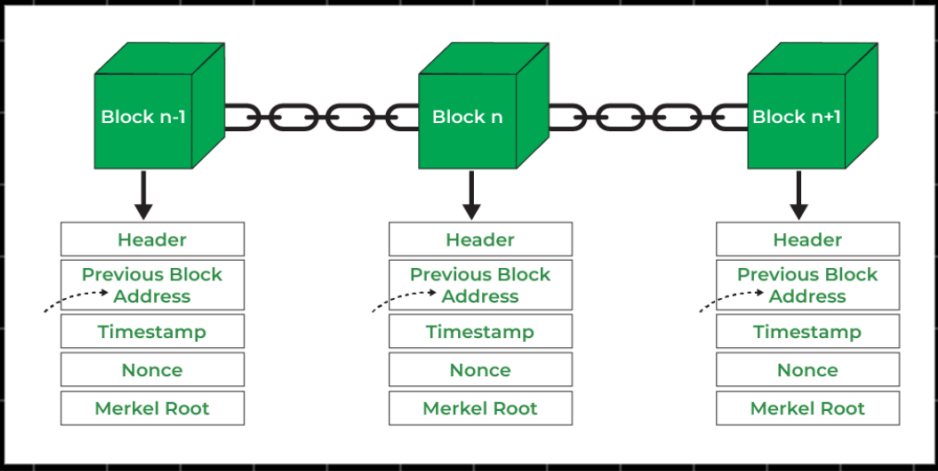
- PoET is a consensum algorith that uses less resources and energy efficient.
- PoET is permissioned blockchain.
- Every miner is imposed with waiting time period generated at random.
- The person with less wating time wakes up first and have higher chances of adding a block to the block chain.

⑦ PoA (Proof of Authority)

- Proof of Authority is an consensus algorithm where miners stake their identity.
- POA is Permissioned Blockchain, who ever wishes to join the network needs to show their true identity, i.e. {the block chain manager needs to do an KYC (Know Your Customer)}.
- If any miner tries to do any sort of activity effecting the blockchain their reputation will take a hit.
- In POA is sort of centralised as the blockchain manger can kick out any miner if found doing any sort of malatious activity.
- It has 51% resistance from attacks.

* Private Business Enterprises normaly uses it, has high throughput.

Block chain Architecture →



Header →

It is used to identify the particular blockchain. A block header is hashed periodically by miners by changing the nonce value as part of normal mining activity, also three sets of block metadata are contained in the block header.

Previous Block Address / Hash →

It is used to connect $i+1^{th}$ block to the i^{th} block using hash. In short, it is reference to the hash of the previous block in the chain.

Timestamp →

Peice of data that records the exact time and date at which block was created. This info is included in the block header and is used to verify the order of block in blockchain.

- They help prevent double spending
- To prove the existance of data at a specific point in time.
- Establish chronological order for transaction.

Nonce →

A number compared to the live target if smaller or equal to current target.

Merkel Root →

Type of data structure frame of different blocks of data. It is used to compute the root hash by computing the hash value of each transaction.

Key characteristics of Blockchain Architecture →

- **Destributed Network →**

  Block chain operates on a decentralized network of computers called nodes. These nodes communicate with each other to acchive consensus.

  Distributed network is resistant to fault tolerance and against attacks.

- **Blocks →**

  Block chain consists of series of blocks containing transaction or data. They are linked in cronological order formity a chain.
  Each block contains a hash and reference to privous block, ensuring intigrity and immutability of the block chain.

- **Cyptography →**

  The data recorded in the blocks are incrypted usig various methods, hash function, digital signature, public-key cryptography.

  Insures Integarity and data privacy.

- **Consensus Mechanism →**

  Enables nodes in the network to agree on the validity and order of transaction.

  Types of consensus algorithm → POW, POS, DPOS, POC, POA, POB, POET.

- **Smart contract →**

  Self-executig contracts with pre-defined rules and conditions.

- **Wallets →**

  Software application for the users to store and maintain cryptographic keys, which are required to access and transact on block chain.

  User friendy integrated with block chain, managing digital assets.

Types of Block Chain

Private
- Only permissioned nodes can participate.
- Centralized.

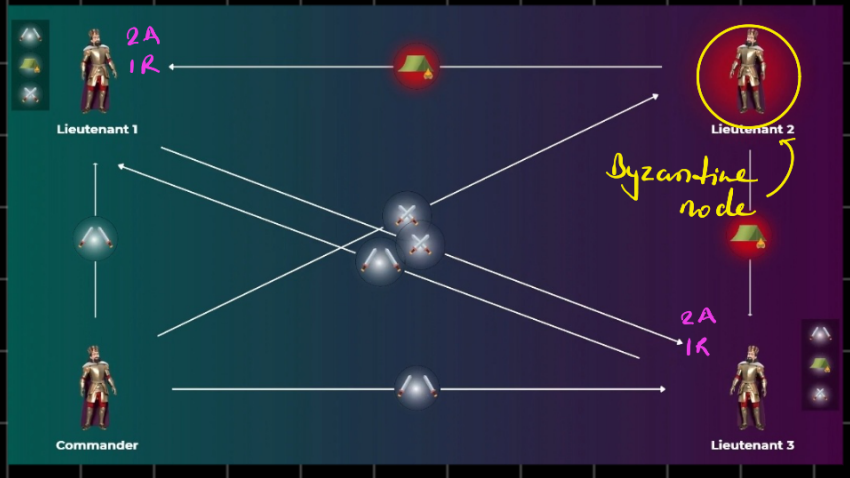
Public
- Any one can read, write and audit the activity on a public block chain.
- Any one can join and take part in activity.
- Decentralized.

Consortium
- Partially centralized, managed by a group than a single permissioned node.
- Only elected nodes by government and a group of organization can participate.

Permissioned
- Centralized
- Only authorised nodes can do transaction, read, write audit the entire activity.

Permissionless
- Decentralized
- No node is more powerful than other.
- Decision are made as per the majority voting.

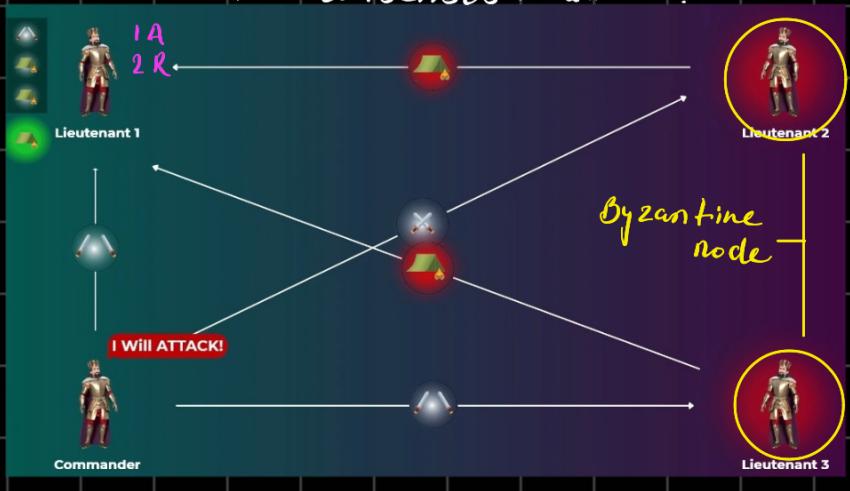
Byzantine Fault Tolerance (BFT)



* They retreat or attack at the same time.

- No one knows who the traitor is.
- They all come to an concensus as per the max vote.

- Here 3 people have voted for and 1 against.
- Reached on consensus to attack.



- Here like (1) has gotten 1 for and 2 against.
- Retreats at consensus.

If traitor number > 1/3, block chain is influenced.

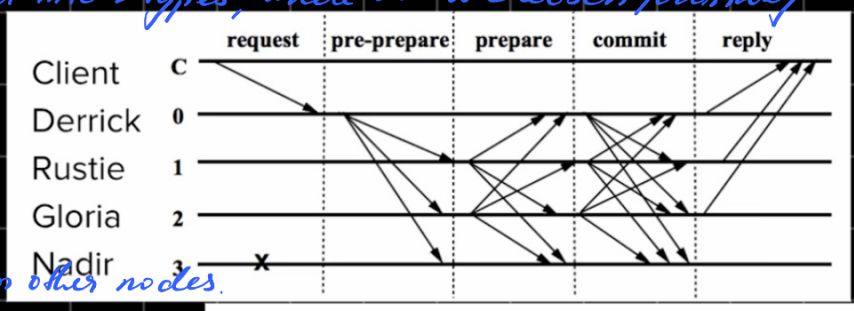
Practical Byzantine Fault Tolerance (pBFT) →

Applied in modern distributed computer system.

The node in the network which tries to mislead others is termed as Byzantine fault node.

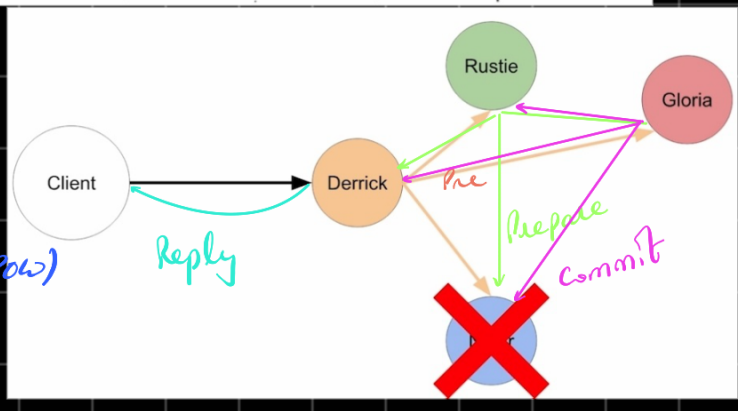
The number of malicious nodes in the network should not be more than 1/3.

In PBFT the nodes are divided into 2 types, where one is choosen primary and other as secondary node.



Role of Primary node →

- Addition of Block
- Collect all the votes from other nodes.
- And reach at an concensus.



Benefits of pBFT →

1. Low energy consumption (comparison to POW)

Limitation of pBFT →

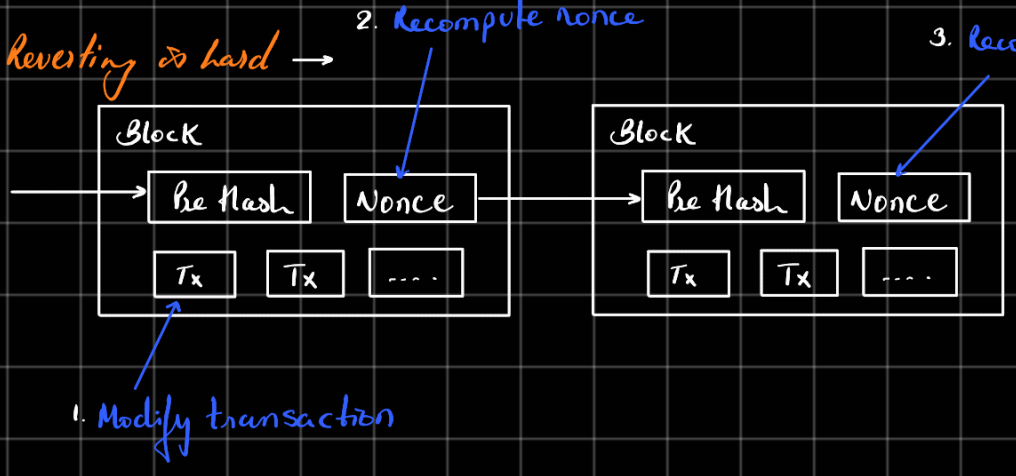
1. Communication overhead

   as every node in the network communicates with each other.

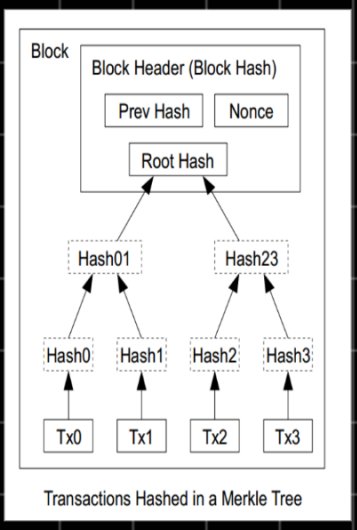
2. Scaling

   pBFT does not scale well because of communication overhead.

Reverting is hard →

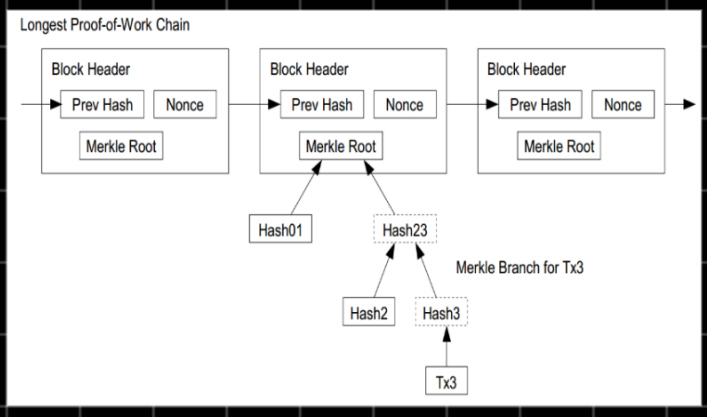


Merkel Tree



Transactions Hashed in a Merkle Tree

- only keep the root hash
  - delete the interior hash values to save disk.



- Any user can verify a transaction easily by asking a node.
- First get the longest POW chain
- Only need Hash 01 and Hash 2 to verify.

Zerocoin →

Zerocoin works by using a cryptographic technique called zero knowledge proofs. Zero-knowledge proof allows a party to prove to another party that they know a certain peice of information without actually revealing the information itself. In this case information is the value and the serial number of bitcoin.

Peer to Peer →

- Group of devices at same hierarchy
- Any system can become a server and a client (Dual function of node)
- Equal power and rights.
- Can be used to perform same tasks
- Distributed network
- No central authority
- Simultaneous sharing and recving makes the network efficient and faster

Types of P2P network →

- Unstructured P2P (Un organized)
- Structured P2P (Hash function)
- Hybrid P2P

Criteria	Structured P2P	Unstructured P2P
Node organization	Nodes are organized into a specific structure or topology, such as a ring or a tree	Nodes are connected randomly or in a P2P fashion
Routing and searching	Efficient routing and searching of data within the network	Routing and searching can be more difficult
Complexity	More complex due to the organization and structure of nodes	Simpler to implement
Pros	Enables efficient routing and searching of data within the network	Simpler and more flexible to set up and implement
Cons	Can be more difficult to implement and manage	Routing and searching can be more difficult due to lack of structure

Bitcoin

- Crypto Currency
- POW
- Transparent and Secure
- decentralised and destributed digital ledger
- Satoshi Nakamoto
- Global Currency
- Limited amount
- Investment
- Financial Inclusion
- Privacy
- Ownership controll over your wealth.

- Low scalability
- Volatility
- size 1MB
- upto 7 transaction 1 min
- Privacy leakage
- High energy consumption
- Security of 51%.
- Lack of adequate skill set.

Bitcoin : a pear to pear electronic cash system

9 pages
31 oct 2008
Proposal for trustless system of electronic transaction
Satoshi Nakamoto
Fundamental change in execution of global payment
Transformation in terms of data management
Flow of paper
abstract, introduction (origin of idea, transaction, digit signatures, public key, time stap server, POW, network, disk space privacy,

conclusion.)

structure of bitcoin transaction →

Input	Output
Previous transaction Id	Value
Previous transaction index	script public key (script PubKey)
script Sig	

1. Input →

An input references a previous transaction output and provides funds for current transaction.

2. Output →

Output specifies the recepient of the Bitcoin and the funds to be received. Also has the script locking that specifies the condition under which the recipient can spend.

3. Transaction ID →

Unique Id used for tracking and verifying transaction on the block.

4. Digital Signature →

To prove the ownership and authorization of the funds being spent.

5. Transaction fee →

To incentivize miners to include their transactions in the next block.

6. Change Output →

When transaction spend < total amount of previous transaction then its send back to the sender.

1. Input

The coins that user owns and wants to transfer to another person.

2. Previous transaction id

This is the id of the transaction in which the bitcoin was created and assigned to the owner.

3. Previous transaction index

Every bitcoin transaction can have multiple output and every output is identified with a unique index

4. Scrip Signature

It encodes the public key and the signature of current owner.

5. Outputs

Are newly generated bitcoins locked to the hash of public key of payee.

6. Value

The number of bitcoin key transfered to the payee.

10^8 satoshi = 1 Bitcoin

7. Scrif Pub key

This is sequence of instructions that takes script sig as input and returns true if a legitimate owner tries to unlock the bitcoin, other wise it returns false.

Bitcoin scripting Language and their Uses

Simple stack based programming language used to specify condition under which the Bitcoin can be spent.

1. Pay to Public Key Hash →
   - Most common type
   - The recipiant can spend the funds by providing digital signature

2. Pay to Script Hash →
   - Conversion of complex scripts into individual scripts
   - To redeem need to provide script that matches Hash specified in the transaction output.

3. Multisignature Wallet →
   - Multiple public Key in a transaction
   - Multiple signatures are required

4. Time-Locked Transaction →
   - The reciver can only spend after a period of time.

5. Escrow →
   - Once only the third pasty approves for the transaction then its valid.

6. Atomic Swap →
   - Exchange of Bitcoin for other cryptocurrency
   - Trustless exchange

- **Bitcoin Scripting →**

Bitcoin's scripting language is called stack based scripting language because it uses the stack data structure, that allows 2 operation → 1. push 2. pop

Principle LIFO,

The scripting language executes the script by processing each item from left to right. Numers are pushed on to the stack and operators push or pop one or more parameter from the stack, act on them and might put the results back into the stacks.

eg :- OP_ADD

OP_EQUAL

Ethereum →

- Ethereum blockchain it is a decentralized blockchain platform that is establishes a peer to peer network that securely executes and verify application codes called smart contract.
- Smart contracts allows the participant to transact with each other without a trusted central authority. Transaction records are immutable, verifiable and securely destributed across the network giving participants full ownership and visibility into transaction data.
- The transactions are sent from and received by user created ethereum accounts. A sender must sign transactions and spend ether as a cost of processing transaction on the network.

- 2022 - POW
- Now POS

Note → 15/sep/22 ethereum blockchain moved from POW to POS consenses mechanism.

This upgrade improved sustainability of ethereum by lowering the energy consumption and improving scalability, security and sustainability.

Smart contracts —

So these are self executing aggrements, with the terms of aggrement directly written in the code. They automatically inforce conditions and actions specified within the contract.

So theirs are certain key aspects →

- Turing-Completness
- Decentralised execution
- Gas and EVM (Etherium Vertual Machine)

  Gas is a limit of measurement representing computational efforts required to execute a smart contract. Each operation in a contract consumes certain type of Gas. Users pay for gas using ether. EVM is runtime environment where smart contracts are executed.
- ERC standards (Etherium Request for perfarmance)

  This defines common rules and interfaces for different tokens and contract.

  eg: ERC - 20

  ERC - 721

- Smart contract interactions
- DEFI (Decentralised finance)

  DeFI protocal leverage smart contract provides different services like lending and borrowing the assets.

Turing completeness →

Etherium turing is able to use its code base to perform vertually any task as long as it has correct instructions, enough time and processing power.

Verification challanges in smart contract →

1. Correctness of code :-

Verifying the correctness of smart contract code is crucial to avoid vernabilities and unintended behaviour. For this the code has to thoroughly tested, audited and reviewed.

2. Security Vernabilities :-

Re entrancy attacks, integer over/under flow, unauthorised access are some of the succeptable security vernabilities.

3. External data verification :-

Smart contracts relie upon external sources to obtain information. To ensure accuracy and reliability, the data has to take from a trusted source.

4. Formal verification :-

It is a methamatical technique used to check the correctness and the security of a smart contract. However it is complex, resource intensive and require special skills and tools.

5. Interoperability and Integration :-

When integrating multiple smart contracts, compatibility and interoperability could be a challange. Verifying the correct integration of different components becomes crucial to avoid unsafe behaviours.

6. Upgradibility and Governance :-

Smart contracts may need to be modified over time, so there must be some governance mechanism to upgrade the decisions that can be challenging.

7. Using smart contracts to enforce legal contracts :-

Smart contracts have the tensile to revolutionized the enforcement of legal contracts by automating and streamlining the process.

Note :- Not all smart contracts can be automated.

a) self executing agreements.
b) Immutable and transparent record keeping.
c) Automated payment and Escrow (Escrow services can be implement ing smart contracts, here the funds are held in a secure manner until specific conditions are met.)
d) Condition execution and dispute resolution.
e) Multiparty contracts (multiple agreement undes are contract involving multiple stake holders).
f) Digitization of legal contracts

Comparing Bitcoin scripting with Ethereum Smart Contract →

Functionality and Turing Completeness →

- Bitcoin scripting language is limited in functionality and is not turing complete. Provides predefined operations that can lead lack of features.

- Thereum smart contract language is turing complete and it allows creation of complex and recursive functions as well

Use cases →

- Bitcoin scripting language is designed for the creation of simple transaction scripts and time log

- Ethereum smart contract are used for creation of decentralized applic

"DApps" (Decentralised Application) and development of decentralised finance protocols "Defi"

Programming Paradigm →

- Bitcoin scripting language is stack based thus following the principle LIFO

- Ethereum solidity language used Object oriented programming paradigm.

Gas Module →

- Bitcoin does not have any concept of gas buildin. Transaction fees in bitcoin is based upon size of the transaction in the bitcoin parallel bit.

- Ethereum the users have to pay gas fees to execute smart contracts.

Development Echo system →

- Ethereum has more extensive and mature development echo system as compared to bitcoin.

Interoperability →

- Ethereum's smart contract are more interoperatable as compared to bitcoin scripting language. These smart contracts can interact with other's smart contract with in the ethereum echo system enabling the creation of a complex decentralised system.

- Bitcoin scripting focuses on bitcoin transaction and does not have builtin support for extensive interoperability.

Hyperlegure →

Collection of opensource projects created to support the development of blockchain based distributed legure's it aims to create framework, tools, library required to build block chain and related applications.

Linux foundation in 2016 created the Hyper ledgures framework later Ztel, IBM, Samsung, Microsoft, U28A, American Express and some blockchain startup like Block Force contributed.
Hyperledgure acts as a hub, for different distributed ledgure framework and libarys. It works by providing a infrastructure and standard required for developing block chain system and application.
Developers used by per ledgure green house to develop business block chain projects.

Layer → Concensus
Smart contract
Communication
API
Identification Management

Security consedrations and best practeces for "fabric networks"

1. Identy and access management →
   - Used to ensure that only authorised participants can access the network.
   - Cryptography identities are used to authenticate (x.509 certificate).
   - Implement role based access controll, to controll the permissions. (RBAC)

"Roall Based Access Controll"

2. Secure Communication →

· Employ TLS (transport layer security) encryption for secure communication between the nodes, peers and clients.

· Mutual TLS is also established for personal communication.

3. Chain code security → Chain code "Smart contract"

· The chain code has to throughly audited and tested for vulnabilities.

· Code reviews and analysis tools must be used to fix potential security threats.

(Endrossmen) → Aggrement

4. Endgrossmen Policy →

· Define and implement a robust endrersmen policies to ensure that the transaction are validated by the required set of peers.

5. Private data and confidentiality →

· Neutralize fabrics private data collection feature to store sensative data of network.

6. Secured deployment →

· Implement network segmentation, harding of nodes and firewalls etc. to deploy a secure network.

7. Monitory and Auditing →

· Implement robust loging and monitory mechanism to track network activities and detect suspecious behaviour.

8. Regular testing and security Accessment →

· Perform regular penitration testing and security accessments to identify the potential vulnabilities.

Development tools & frameworks →

1) Ethereum

2) Hyperlegure fabric

3) Cosda

- It is an open source blockchain platform designed for businesses. It is a permissioned blockchain
- It focuses on privacy and allows the development of CosDx (Cosda decentralized Application).
- It usualy use Kotlin and Java programming.

4) Truffle

- Ethereum based blockchain framework.
- It provides a collection of tools for smart contract compilation, deployment and testing.

5) Web3.Js

- It is a Java script library that provides an interface to interact with ethereum based application.
- It allows the developers to interact with smart contract, send transaction and peri using Java script.

6) Solidity

- Progrmg language designed for writing smart contract for ethereum based applications.
- It is statically tied and supports inheritance, libraries and complex user defined types.

7) Remix

- Web oriented online development environment. That provides a web based interface for smart contracts.

3) Ganache

- Private block chain for ethereum development
- It allows the developers to create and manage a local ethereum block chain network for testing purpose.

Languages for Block chain development —

i) Solidity

Solidity programming language is one of the most powerful language designed for development designed for developing smart contract's for ethereum that is the second largest market of crypto currency by capitilization.

- It is an object oriented programming language liscenced under GNU (General Public licence V.3.0).
- It was designed by Gabin Wood and developed by Cristain and Alex along with ethereum core contributers.
- Programs written in solidity run on "EVM" is other compatible "VM."

Java script →

It is particular used for front end dvelopment for decentralised application It is commonly used in conjunction with libraries like web 3.js to interact with block chain network, smart contract and decentralised protocols.

Go or Golan →

- Developed by Google, known for its simplicity efficiency and strong support for concurrent programming.

Java →

- Used in block chain frameworks like corda to develop DaPs and smart contr
- Corda primarily uses Kotlin, which is interoperatable with Java and can use Java for primary development as well.

C++ →

- It as used in block chain projects like Bitcoin for core protocol development It is favoured for low level development and performance critical parts of blockchain system.

Rust →

- It is a system programming language that emphasise on safety performance and concurrency.
- It has strong memory safety garenty that given it the ability to build efficient and secure blockchain system.
- Some of the most popular blockchain project that uses Rust are →
  - Polkadot
  - Solana
  - Oasis network etc.

Kotlin →

- Modern programing language developed by "Jet brains".
- It was initially targetting Java Vertual machine, but gained popularity, now diffent blockchain that uses Kotlin.

Secure communication and Encryption in Hyperledgure fabric →

1) TLS (Transport layer Security)
   - fabric uses TLS to secure the communication channeal between the network and participants.
   - It ensures confidentiality and integrity of data by encrypting it during the transmission using digital signatures.

2) mTLS (mutual TLS)
   - It is an extension of TLS that adds an extra layer of security by requiring both client and server to present their digital signatures

during TLS handshake.

3) Channel encryption →

- Fabric supports the concept of channels, thus allows different groups of network participants to have private communications governed by set of policies.

4) Private data collections →

- This features allows certain data to be shared selectively with authorised network participants.
- Access control protocols govern data collection.

5) Key management →

- Fabric networks require robust key management to ensure the security of the encryption key.
- Key is used for TLS, mTLS, channel encryption and Private data collection should be securely generated, store, acculated following best practacies.

Basic Cryptography →

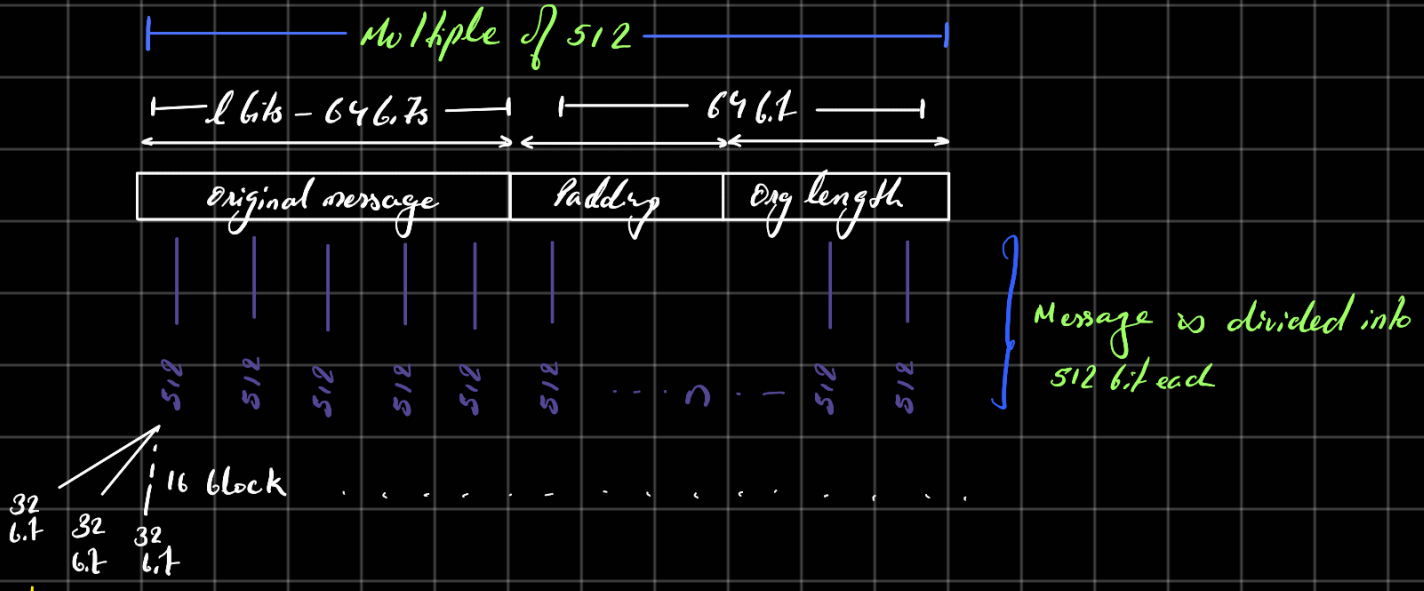
Hash function in blockchain transaction →

Cyptographic hash functions generate a fixed length characters string from data records of any length (can be a word or sentence or a complete line). It is used for security perposes and contribes to be backbone of crypto security.

Hash of a transaction makes it easy to identify transactions on the blockchain

SHA-256 →

- Plaintext of 2^{64} bit limit
- Hash value output 256 bit
- Irreversable



- $H(A) \neq H(B)$
- $H_1(A) = 0123$, $H_2(A) = 0123$: $H_1(A) = H_2(A)$

Perpose of Hash functions →

Hash functions were orginated to make content uniform and associate unique identefiers with them.

- Hash function is mathematical function that converts the input of variable lenght size into fixed sized compressed value.
- Hash function with n-bits → n bit hash function.
- Protects the integrity of the message.

Apart from this it is used for →

1. Calculating check sum of an object
2. Abiguity in the content

Eg :- some popular hash funch's are SHA-256, MD-4, MD-5, SHA-1, SHA-512.

↳ For a hash function to be cryptographically secure →

1) Collision Resistant
2) Hiding
3) Puzzel friendly

Puzzle friendly hash →

A Hash function is puzzel friendly if its infeasable to find solution in 2^n time. Meaning their exists no solution other than apply of Bruteforce method

Fixed length mapping →

- The output size of an hash function is fixed and independent of input size.
- Output size of SH-256 function is 256 bits for any arbitarly sized input.

Uses of Hash functions →

1. Digital Signature

To sign a message cyptographic hash is used and then receivers public key is used to encrypt.

2. Data entigrity

3. Storing Passcoards in a database

Collision resistance →

In cryptography collision resistance is a property of cryptographic hash functions. A hash function is collision resistant if, it is hard to find two inputs that hash to the same output.

Digital signatures are used to prevent messages from being tempered during a transmission. The digital signature of the reciver acts as a secret key that can be used to open the message.

How Hash functions helps with signatures ?

- When the message is recieved and the signature, they do two things →
  1. They apply magic machine (collision resistant hash function) to the message to get the fingerprint.
  2. Use of public key to check, the authenticity of the message.
  3. The magic machine is collision resistant no one can create a diffrent message matching a hash.

Note →

A collision resistant hash function create unike hash for messages and digital signature use this hash to ensure, messages integrity and authenticity.

Public Key cryptography →

- Way of keeping messages safe while comunicating over the network.
- Uses Public and Private Key.

Public Key →

- For encryption of the messages.

Private Key →

- for decryption of the messages.

Verifiable Random Functions (VRFs) →

- Way of generation of random function in a special verifiable way.
- It is like mayor process of generating random number that every one in the network can agree upon.

Randomness →

- Random number generation is a process of picking out a numbers from a pool without knowing.
- VRFs have a property that allows pasticipants to check the numbers is genuinly random or not.

Verifiable →

- This property of VRFs ensures that the numbers generated are not intervented by any algorithm or external sources.

VRFS are used for different applications →

- Generating of secure cyptographic key.
- Creating unpredictable challanges in online games or ensuring fareness in lottery draws.

Verifiable Random function is a cryptographic function takes input of public/private key pair and a seed computy gives as a pseudorandom output along with authentication proof which can be verified by any one.

- Verifiable → Any one can verify the randomness of VRF generated.

VRF secret key can compute the Hash and anyone with the public key can check the correctness of the hash.

- Random → The output of VRF is unpredictable to anyone who doesn't know the seed / private key and follows no pattern.

- Function → VRF rely on a mathematical algorithm to produce both the random number and a proof that verifies its authenticity.

Zero Knowledge System / Zero Knowlege Proof (ZKP) →

- It is an encryption scheme proposed by MIT researchers, "Silvio Micali", "shafi Gold Wasser" and "charls Rekof" in 1980's.
- In this method one party (prover) can prove a specific statement is true to another party (verifier) without disclosing any additional information.
- Zero Knowledge encryption makes sure that no one except the service provider or blockchain application development agency can access to your secure data.

- Benifits of KZP →
  1. Simple
  2. Secure
  3. Transactions (Easy Transaction)

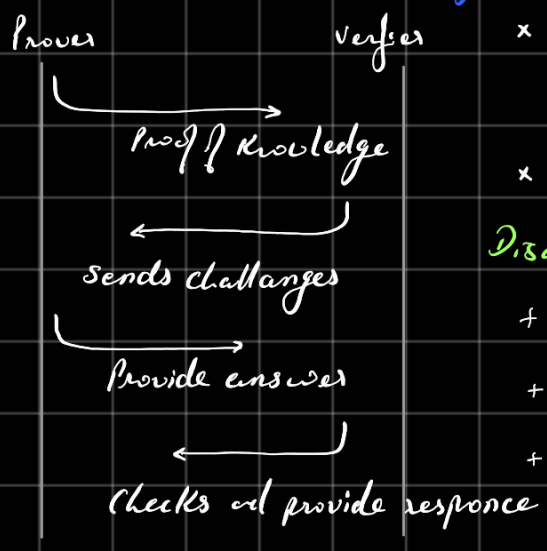
- Disadvantage →
  1. Lengthy (More number of computations are envolved)
  2. Limited
  3. The ZKP protocol demands the secreat to be a numerical value in other cases translation is required.

- Properties of ZKP →
  1) Completeness
     - 2 parties envolved verifers ad approver
  2) Zero Knowledge

- Types of ZKP →

1) Interactive Zero Knowledge Proof

Exchange of message takes place between the "prover" and "verifier".



× Flexible can be used to prove Knowledge of different types of information. {Password, Authority over an asset "crypto curve"}

× Random challanges

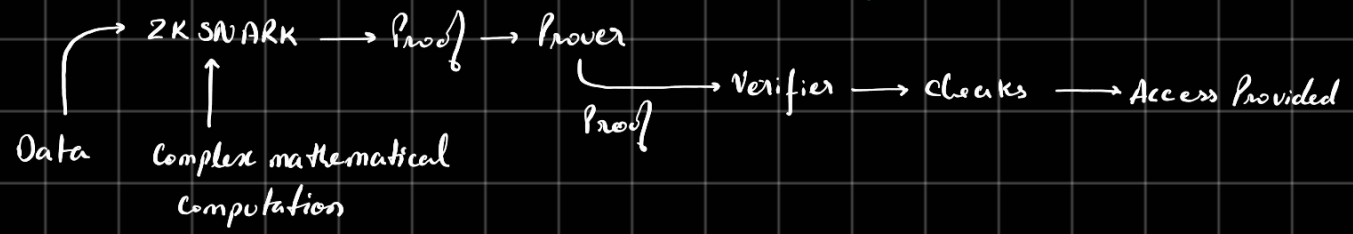
Disadvantage →

+ High level trust between "Prover" and "Verifier" is required.

+ Computation resource

+ Time consuming

2) Non Interactive Zero Knowledge Proof



Proof → Highly compressed representation of information. (small in size too)

× Fast, Efficient and Scalable

× Few Computation Resources required.

Benficial in Destributed system → "Block chain".

Drawbacks →

+ Less flexible, can be only be used to prove Knowledge of certain types of information.

+ Technical expertise is required.

- Application of ZKP in blockchain technology →

1) Messaging
2) Authentication
3) Storage Protection
4) Creating Private Block chain

5) File system control

6) Vote verification.

Z-Cash →

- Uses Zero Knowledge Proof
- Created in 2016
- Pesoudonoms, data stored on public ledgure, but personal data is not directly linked the data stored on the ledgure.
- Provides shielded and unshilded blockchain capability.

ZKSNARKs → (Zero Knowledge Succient Non-Interactive Argument of Knowledge)

Data → ZK SNARK → Proof → Highly compressed representation of information. (Small in size too)

Complex mathematical computation → ZK SNARK

- Fast, Efficient and Scalable
- Few Computation Resources required.
- Benficial in Destributed system → "Block chain".
- Non-Interactive
- Argument of Knowledge →
  - The proof generated by ZKSNARKs convinces the verifier that the prover posses certain knowledge without revealing it.
  - In Content of Z-Cash, the prover can demonstrate the ownership of funds and validity of transaction without disclosing about the sender or reciver or the transaction between them.

Attacks →

1) Sybil

Creation of Multiple fake accounts / nodes in order to control the network.

There are of two types →

1) Direct Sybil attack →

The incorrect nodes makes direct contact with other nodes on the system influencing the node.

2) Indirect attact →

The good node communicates with the node in direct contact with the Sybil node.

- In this an individual or a group creates multiple fackle identities (Sybil Nodes) in order to controll major part of the network.
- It is a form of online security violation.
- The origin of the learn Sybil can be traced back to 1973 by "Flora Schreiber".
- There are 2 types of sybil attacks →

1) Direct Sybil Attack →

- In this attack the malecious nodes, interact directly with a network nodes.
- This type of attack is usually effective because the genevn nodes are enable to identify the node which are not legitimate.

2) Indirect Sybil Attack →

- It is a kind of attack that happens when the fake note come in contact with the nodes that are connected to the authentic node.
- In contrast direct attacks, hackers use proxi node to launch an indirect attacks.
- The intermediate nodes, which are positioned between the sybil node and the genvin node represents the point of failure.

2) Selfish mining

- A malacious attack targetting the blockchain based bitcoin system.
- It is deciteful cryptocurrency in which one / multiple miner solves a hash opens a new block and withhold it from the public blockchain.
- This action creates a fork (copy a code and creating different software) which is then mined to create publick blockchain.

3) 51% (majority attack) it is a situation where a single entity or a group colluding entities control more than 50% of total hash rate of a blockchain network it is an attack on 51% of nodes on the network, theoretically giving the controlling party the power to alter the blockchain.

Advent Algorand Algorithm →

- A concences mechanism designed for Algorant blockchain, which is decentralized and permissionless in nature.
- It was proposedly "Silvio Micali" who is a turing awarded winner computer Sintest along with its team.

- The primary goal of this algorithm is to provide a secure scalable and decentralized platform for fast and no cost transaction, while insuring security and decentralization.

* Algorand algorithm uses Byzantine Agreement that leverages POS.

- Key Features of Algorand Algorithm →

1) PPOS (Pure Proof of Stake) The node with most number of tokens "Validator"

Algorand uses pure proof of stake mechanism which is based upon less computation power consumption thus more efficient than "POS".

2) Participation and Propagation

Byzantine Agreement Process - -.

3) Fast transactions

Less energy consumption

High transaction throughput

Low lantency

4) Fork free

This mean the blockchain will not experiance multiple fork or multiple complety chains.

This property ensures high security and simplefies decision making process.

5) Decentralization and Security

Random selection of participants ensure high degree of decentralization and prevents from concentration of power.

Aditionaly Algorand is designed to more resilint to attaks due to PPOS mechanism.

6) Token Economy * Platform for DApps development.

Algorands native cryptocurency is known as Algo.

It plays a vital role in securing the network through staking, participating in concences mechanism and providing insentives to evaluators.

Sharding

- Sharding is a technique used in consences mechanism to improve scalability.
- It involves partisioning of the entire network into smaller yet into managable called Shard.
- Each Shard can process indipendently and in parallel which allows the block chain process high number of "transaction per second" (TPS) and improves the overall network performance.
- Examples →
  1) Etherium 2.0 Beacon chain
  2) Ziliqua
  3) Harmony Byzantine fault Tolerance
  4) Elrond SPOS, dynamic shrads
  5) Polkadot Multichain block

Application Specific Blockchain →

- Specific Blockchain refers to the type of Blockchains which are designed for a specific use case, Unlike public Blockchain networks like Etherium which are general purpose, application specific blockchains are customised to address the requirements of perticular set of users.

- Key features of Application specific Block chains →
  1) Customization / Tailorization
  2) Performance {High thrughput and low latency}
  3) Privacy and Access Control
  4) Concensus Mechanism {Depends upon the need of the Application}
  5) Integration with public blockchains

     required for interaction with public blockchain to leverage security and decentralisation, while benifits from customization properties of private block chain.

Application specific blockchain have proven to be useful in various sectors like supply chain management, Health care, finance etc.

Block chain Standardization →

- Blockchain Standardization refers to the process of establishing common protocols, guidelines and specifications for the development, Implementation and inter operability of blockchain.

- Key aspects →
  1) Interoperability {with other blockchains}
  2) Concensus Mechanism
  3) Smart Contracts {Large used for writing smart contract}
  4) Data formats and structures {Read and Retrival of data format}
  5) Privacy and Security {Encryption Standars, Identity Management, Data Access Control Mechanism}
  6) Regulatory Compliance {Legal Compliance (IT, data regulatory law)}
  7) Governance and Upgradation
  8) Industry specific Standards {Application Specific}
  9) Organisation such as ISO (Internation Onagarisation for Standardization and Institute of Electrical and Electronics Engineers (IEEE) are certain bodies involved in the Standardization of blockchain technology.

Baas (Blockchain as a Service) →

- BaaS is a cloud based service model that allows individuals organisation and enterprises to use blockchain technology without initial setup and infrastructure.
- The service providers offer pre configured blockchain and related services, that makes businesses easy to adopt the technology.

- Key aspects of BaaS →
  1) Cloud Based Solution
  2) Blockchain network provisioning { Ready to use Blockchain (pre config) }
  3) Security and maintainance { Data Encryption Method, Load management, Regular updates }
  4) API Integration { Provides Frontend }

Emerging trends in blockchain →

1) Evolution of DeFi: (Decentralized Finance)

DeFi continued to evolve rapidly with the invent of new protocols, decentralized exchanges (DEXs) and other financial products.

2) Layer 2 Scaling Solution

With the growth of the technology layer 2 scaling solutions like rollup and state channels have provided scalable options.

3) NFT (Non fungible Tokens)

Have application in diverse areas like gaming vertual realstate, music sports etc.

4) CBDCs Advancements

Central Bank Digital Currency's

With the invent of Blockchain several countries and central banks have accelerated their exploration and development of CBDCs that helps in efficient transactions.

5) Interoprability solutions

For seamless data and assets transfer it is important that different blockchain networks provide interoperatable solutions.

Projects like polkadot, Cosmos etc. incorporate interoprability

feature.

6) Sustainable and green blockchain solution

Environmental concerns related to energy consumption of tradition al pow mechanism drives the focus towards the need of sustainable blockchain solutions like POS.

7) Enterprise blockchain adoption and supply chain solutions

Enterprises focused upon blockchain technology for transparent and efficient solution's.

8) Decentralized Identity and self sovereign Identity (SSI)

The concept of decentralized Identity gained popularity with projects working on providing individual control over their digital identities and personal data through SSI solution.

9) Regulatory developments

Governments and regulators world wide worked to established clearer regulation and frameworks for block chain crypto currency.

10) Integration of Blockchain with other technologies

Blockchain has several application in combination with multiple technology, like AI, IoT to offer enhanced security and more optimized blockchain solutions.