**Blockchain:-**

Blockchain is a constantly growing **ledger** that keeps a **permanent** record of all the transactions that have taken place in a **secure**, **chronological**, and **immutable** way. It can be used for the secure transfer of money, property, contracts, etc. without requiring a third-party intermediary such as bank or government. Blockchain is a software protocol, but it could not be run without the Internet (like SMTP is for email).

A blockchain is a chain of blocks which contain information. Each block records all of the recent transactions, and once completed goes into the blockchain as a permanent database. Each time a block gets completed, a new block is generated.

[Link1](https://www.javatpoint.com/blockchain-tutorial)

**Ledger: -**

A ledger is a book or collection of accounts in which account [transactions](https://en.wikipedia.org/wiki/Financial_transaction) are recorded. Each account has an opening or carry-forward [balance](https://en.wikipedia.org/wiki/Balance_(accounting)), and would record each transaction as either a [debit or credit](https://en.wikipedia.org/wiki/Debits_and_credits) in separate columns, and the ending or closing balance.

**Public Ledger: -**

The public ledger is used as a record-keeping system that maintains participants’ identities in secure and (pseudo-)anonymous form, their respective cryptocurrency balances, and a record book of all the genuine transactions executed between network participants.

[Link1](https://www.investopedia.com/tech/what-cryptocurrency-public-ledger/#:~:text=The%20public%20ledger%20is%20used,transactions%20executed%20between%20network%20participants.)

**Bitcoin**: A Peer-to-Peer Electronic Cash System was written by Satoshi Nakamoto. The first key idea introduce was that purely peer-to-peer electronic cash that does need an intermediary bank to transfer payments between peers. Bitcoin is built on decades of cryptographic research such as the research in Merkle trees, hash functions, public key cryptography, and digital signatures. Moreover, ideas such as BitGold, b-money, hashcash, and cryptographic time stamping provided the foundations for bitcoin invention. All these technologies are cleverly combined in bitcoin to create the world's first decentralized currency. Bitcoin can be defined in various ways: it's a protocol, a digital currency, and a platform. It is a combination of peer-to-peer network, protocols, and software that facilitate the creation and usage of the digital currency named bitcoin. Note that Bitcoin with a capital B is used to refer to the Bitcoin protocol, whereas bitcoin with a lowercase b is used to refer to bitcoin, the currency. Nodes in this peer-to-peer network talk to each other using the Bitcoin protocol. Decentralization of currency was made possible for the first time with the invention of bitcoin. Moreover, the double spending problem was solved in an elegant and ingenious way in bitcoin. Double spending problem arises when, for example, a user sends coins to two different users at the same time and they are verified independently as valid transactions.

**Bitcoin Working Mechanism:** When you send an email to another person, you just type an email address and can communicate directly to that person. It is the same thing when you send an instant message. This type of communication between two parties is commonly known as Peer-to-Peer communication. Whenever you want to transfer money to someone over the internet, you need to use a service of third-party such as banks, a credit card, a PayPal, or some other type of money transfer services. The reason for using third-party is to ensure that you are transferring that money. In other words, you need to be able to verify that both parties have done what they need to do in real exchange.

For example, Suppose you click on a photo that you want to send it to another person, so you can simply attach that photo to an email, type the receiver email address and send it. The other person will receive the photo, and you think it would end, but it is not. Now, we have two copies of photo, one is a simple email, and another is an original file which is still on my computer. Here, we send the copy of the file of the photo, not the original file. This issue is commonly known as the double-spend problem. The double-spend problem provides a challenge to determine whether a transaction is real or not. How you can send a bitcoin to someone over the internet without needing a bank or some other institution to certify the transfer took place. The answer arises in a global network of thousands of computers called a Bitcoin Network and a special type of decentralized laser technology called blockchain.

[Link1(pg38)](https://mrcet.com/downloads/digital_notes/CSE/IV%20Year/CSE%20B.TECH%20IV%20YEAR%20II%20SEM%20BCT%20(R18A0534)%20NOTES%20Final%20PDF.pdf) [Link2](https://www.javatpoint.com/bitcoin)

## Blockchain 2.0: Smart Contracts

**The next development in blockchain technology expanded on the capabilities of the blockchain protocols. Four years after the rise of Bitcoin, Vitalik Buterin introduced the concept of Ethereum, a technology based on blockchain with some notable improvements over the previous generation.**

Ethereum was the first blockchain with the smart contract integrated into its protocol. In simple terms, smart contracts are a set of codes that are automated when certain conditions are met. These contracts enable two users or organizations to do more than just simple cryptocurrency transactions.

Ethereum, for example, is a technology with multiple components - a virtual machine with multiple layers of information, user accounts, contracts, and cost accounting (a metric known as ‘gas’).

Therefore, smart contracts allow two parties to automatically execute complex tasks while facilitating the exchange of digital currency. As a result, Blockchain 2.0 introduced several new opportunities that were not possible before due to the limited scope of the first-generation blockchains.

The entire decentralized finance (DeFi) infrastructure pretty much became practicable due to the launch of the smart contracts. It allowed programmers around the world to develop [decentralized applications](https://originstamp.com/blog/11-examples-of-decentralized-applications-dapps/) and software on the existing blockchain platforms.

Soon, the industry witnessed a surge in growth in [decentralized autonomous organizations](https://originstamp.com/blog/10-examples-of-decentralized-autonomous-organizations/) (DAOs), redeemable tokens for ICOs or governance controls, and identifiers for unique items such as NFTs (non-fungible tokens).

This new generation of blockchains still had its flaws. For example, smart contract code could include bugs and security vulnerabilities. The latter can have serious consequences as they leave the blockchain open to attacks by hackers. Moreover, due to the ever-increasing traffic on the Ethereum blockchain, the ecosystem suffered from delayed transactions and high gas fees.

Examples of Blockchain 2.0 technologies include cryptocurrency platforms (Ethereum, Lisk, and Neo), Defi applications (like MakerDAO, Uniswap), and browser extensions like MetaMask

[**Link1**](https://originstamp.com/blog/blockchain-1-vs-2-vs-3-whats-the-difference/)[**Link2**](https://www.cnbctv18.com/cryptocurrency/blockchains-a-look-at-3-generations-and-what-is-next-14241342.htm)

**Smart Contracts :-**

Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary’s involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met.

Or

A Smart Contract (or cryptocontract) is a computer program that directly and automatically controls the transfer of digital assets between the parties under certain conditions. A smart contract works in the same way as a traditional contract while also automatically enforcing the contract. Smart contracts are programs that execute exactly as they are set up(coded, programmed) by their creators. Just like a traditional contract is enforceable by law, smart contracts are enforceable by code.

* The [bitcoin](https://www.geeksforgeeks.org/what-is-bitcoin/) network was the first to use some sort of smart contract by using them to transfer value from one person to another.
* The smart contract involved employs basic conditions like checking if the amount of value to transfer is actually available in the sender account.
* Later, the [Ethereum](https://www.geeksforgeeks.org/introduction-to-ethereum-part-1/) platform emerged which was considered more powerful, precisely because the developers/programmers could make custom contracts in a Turing-complete language.
* It is to be noted that the contracts written in the case of the bitcoin network were written in a Turing-incomplete language, restricting the potential of smart contracts implementation in the bitcoin network.
* There are some common smart contract platforms like Ethereum, Solana, Polkadot, [Hyperledger fabric](https://www.geeksforgeeks.org/hyperledger-fabric-in-blockchain/), etc.

**Features of Smart Contracts**

The following are some essential characteristics of a smart contract:

1. **Distributed:**Everyone on the network is guaranteed to have a copy of all the conditions of the smart contract and they cannot be changed by one of the parties. A smart contract is replicated and distributed by all the nodes connected to the network.
2. **Deterministic:**Smart contracts can only perform functions for which they are designed only when the required conditions are met. The final outcome will not vary, no matter who executes the smart contract.
3. **Immutable:**Once deployed smart contract cannot be changed, it can only be removed as long as the functionality is implemented previously.
4. **Autonomy:** There is no third party involved. The contract is made by you and shared between the parties. No intermediaries are involved which minimizes bullying and grants full authority to the dealing parties. Also, the smart contract is maintained and executed by all the nodes on the network, thus removing all the controlling power from any one party’s hand.
5. **Customizable:**Smart contracts have the ability for modification or we can say customization before being launched to do what the user wants it to do.
6. **Transparent:**Smart contracts are always stored on a public distributed ledger called blockchain due to which the code is visible to everyone, whether or not they are participants in the smart contract.
7. **Trustless:**These are not required by third parties to verify the integrity of the process or to check whether the required conditions are met.
8. **Self-verifying:**These are self-verifying due to automated possibilities.
9. **Self-enforcing:**These are self-enforcing when the conditions and rules are met at all stages.

**How Do Smart Contracts Work?**

A smart contract is just a digital contract with the security coding of the blockchain.

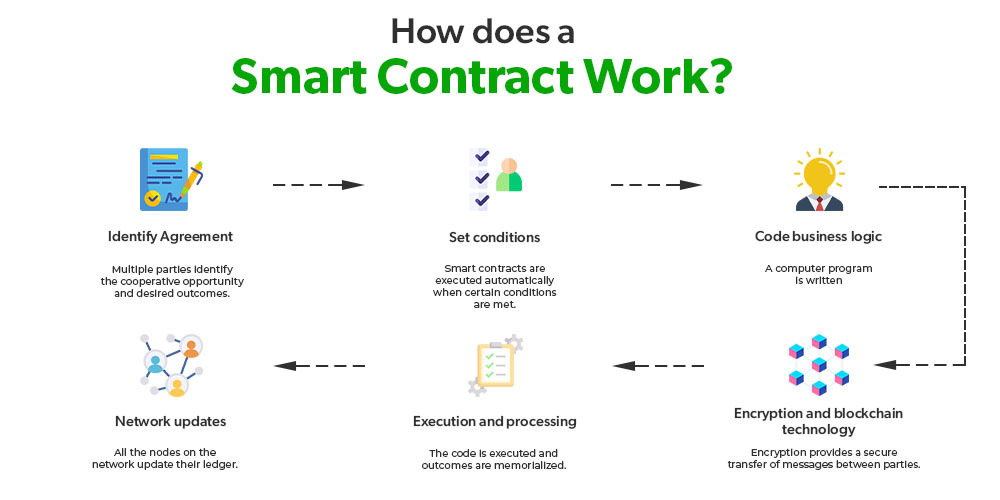
* It has details and permissions written in code that require an exact sequence of events to take place to trigger the agreement of the terms mentioned in the smart contract.
* It can also include the time constraints that can introduce deadlines in the contract.
* Every smart contract has its address in the blockchain. The contract can be interacted with by using its address presuming the contract has been broadcasted on the network.

The idea behind smart contracts is pretty simple. They are executed on a basis of simple logic, IF-THEN for example:

* **IF** you send object A, **THEN** the sum (of money, in cryptocurrency) will be transferred to you.
* **IF** you transfer a certain amount of digital assets (cryptocurrency, for example, ether, bitcoin), **THEN** the A object will be transferred to you.
* **IF** I finish the work, **THEN** the digital assets mentioned in the contract will be transferred to me.

**Note:**The WHEN constraint can be added to include the time factor in the smart contracts. It can be seen that these smart contracts help set conditions that have to be fulfilled for the terms of the contract agreement to be executed. There is no limit on how much IF or THEN you can include in your intelligent contract.

**Smart Contract Working**



* **Identify Agreement:**Multiple parties identify the cooperative opportunity and desired outcomes and agreements could include business processes, asset swaps, etc.
* **Set conditions:**Smart contracts could be initiated by parties themselves or when certain conditions are met like financial market indices, events like GPS locations, etc.
* **Code business logic:**A computer program is written that will be executed automatically when the conditional parameters are met.
* **Encryption and blockchain technology:**Encryption provides secure authentication and transfer of messages between parties relating to smart contracts.
* **Execution and processing:**In blockchain iteration, whenever consensus is reached between the parties regarding authentication and verification then the code is executed and the outcomes are memorialized for compliance and verification.
* **Network updates:**After smart contracts are executed, all the nodes on the network update their ledger to reflect the new state. Once the record is posted and verified on the blockchain network, it cannot be modified, it is in append mode only.

**Applications of Smart Contracts**

1. **Real Estate:** Reduce money paid to the middleman and distribute between the parties actually involved. For example, a smart contract to transfer ownership of an apartment once a certain amount of resources have been transferred to the seller’s account(or wallet).
2. **Vehicle ownership:**A smart contract can be deployed in a blockchain that keeps track of vehicle maintenance and ownership. The smart contract can, for example, enforce vehicle maintenance service every six months; failure of which will lead to suspension of driving license.
3. **Music Industry:**The music industry could record the ownership of music in a blockchain. A smart contract can be embedded in the blockchain and royalties can be credited to the owner’s account when the song is used for commercial purposes. It can also work in resolving ownership disputes.
4. **Government elections:** Once the votes are logged in the blockchain, it would be very hard to decrypt the voter address and modify the vote leading to more confidence against the ill practices.
5. **Management:**The blockchain application in management can streamline and automate many decisions that are taken late or deferred. Every decision is transparent and available to any party who has the authority(an application on the private blockchain). For example, a smart contract can be deployed to trigger the supply of raw materials when 10 tonnes of plastic bags are produced.
6. **Healthcare:**Automating healthcare payment processes using smart contracts can prevent fraud. Every treatment is registered on the ledger and in the end, the smart contract can calculate the sum of all the transactions. The patient can’t be discharged from the hospital until the bill has been paid and can be coded in the smart contract.

**Example Use cases:**

1. Smart contracts provide utility to other contracts. For example, consider a smart contract that transfers funds to party A after 10 days. After 10 days, the above-mentioned smart contract will execute another smart contract which checks if the required funds are available at the source account(let’s say party B).
2. They facilitate the implementation of ‘multi-signature’ accounts, in which the assets are transferred only when a certain percentage of people agree to do so
3. Smart contracts can map legal obligations into an automated process.
4. If smart contracts are implemented correctly, can provide a greater degree of contractual security.

**Advantages of Smart Contracts**

1. **Recordkeeping:** All contract transactions are stored in chronological order in the blockchain and can be accessed along with the complete audit trail. However, the parties involved can be secured cryptographically for full privacy.
2. **Autonomy:**There are direct dealings between parties. Smart contracts remove the need for intermediaries and allow for transparent, direct relationships with customers.
3. **Reduce fraud:**Fraudulent activity detection and reduction. Smart contracts are stored in the blockchain. Forcefully modifying the blockchain is very difficult as it’s computation-intensive. Also, a violation of the smart contract can be detected by the nodes in the network and such a violation attempt is marked invalid and not stored in the blockchain.
4. **Fault-tolerance:**Since no single person or entity is in control of the digital assets, one-party domination and situation of one part backing out do not happen as the platform is decentralized and so even if one node detaches itself from the network, the contract remains intact.
5. **Enhanced trust:**Business agreements are automatically executed and enforced. Plus, these agreements are immutable and therefore unbreakable and undeniable.
6. **Cost-efficiency:**The application of smart contracts eliminates the need for intermediaries(brokers, lawyers, notaries, witnesses, etc.) leading to reduced costs. Also eliminates paperwork leading to paper saving and money-saving.

**Challenges of Smart Contracts**

1. **No regulations:** A lack of international regulations focusing on blockchain technology(and related technology like smart contracts, mining, and use cases like cryptocurrency) makes these technologies difficult to oversee.
2. **Difficult to implement:**Smart contracts are also complicated to implement because it’s still a relatively new concept and research is still going on to understand the smart contract and its implications fully.
3. **Immutable:**They are practically immutable. Whenever there is a change that has to be incorporated into the contract, a new contract has to be made and implemented in the blockchain.
4. **Alignment:**Smart contracts can speed the execution of the process that span multiple parties irrespective of the fact whether the smart contracts are in alignment with all the parties’ intention and understanding.

[Link1](https://www.ibm.com/in-en/topics/smart-contracts#:~:text=Smart%20contracts%20are%20simply%20programs,intermediary's%20involvement%20or%20time%20loss.) [Link2](https://www.geeksforgeeks.org/smart-contracts-in-blockchain/) [Link3](https://www.simplilearn.com/tutorials/blockchain-tutorial/what-is-smart-contract)

**Block in a Blockchain :-**

Blocks are data structures within the blockchain database, where transaction data in a cryptocurrency blockchain are permanently recorded. A block records some or all of the most recent transactions not yet validated by the network. Once the data are validated, the block is closed. Then, a new block is created for new transactions to be entered into and validated.

A block is thus a permanent store of records that, once written, cannot be altered or removed.If a blockchain is thought of as a ledger book, a block is like one page from the book.

[**Link1**](https://www.investopedia.com/terms/b/block-bitcoin-block.asp)  [Link2](https://medium.datadriveninvestor.com/what-is-a-block-in-the-blockchain-c7a420270373)  [Link3](https://help.coinbase.com/en/coinbase/getting-started/crypto-education/glossary/block)

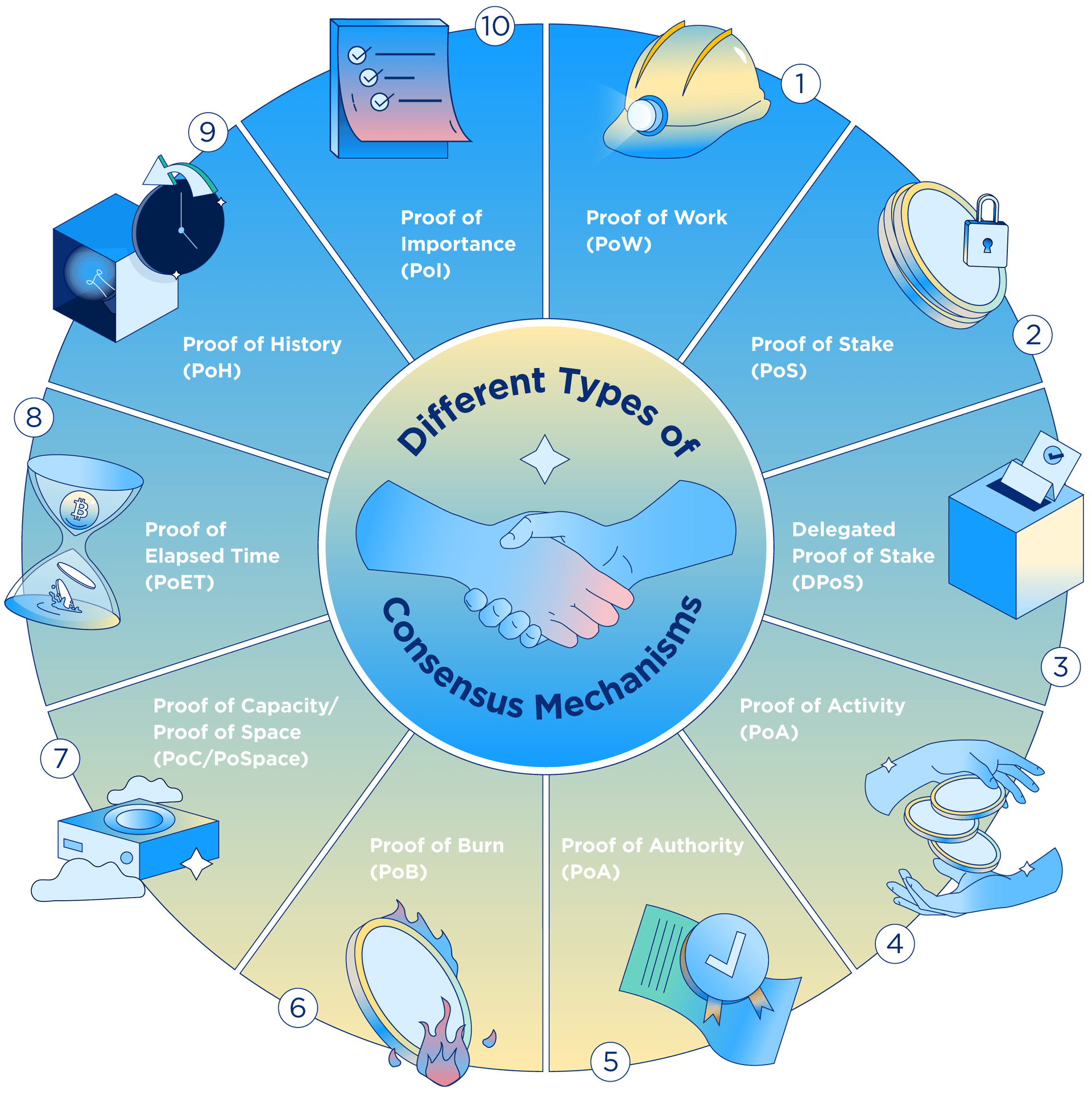
**Distributed Consensus:-**

A procedure to reach a common agreement in a distributed or decentralized multi-agent platform. It is important for the message passing system.

Consensus is the process by which a group of peers – or nodes – on a network determine which blockchain transactions are valid and which are not. Consensus mechanisms are the methodologies used to achieve this agreement. It’s these sets of rules that help to protect networks from malicious behaviour and hacking attacks.

There are many different types of consensus mechanisms, depending on the blockchain and its application. While they differ in their energy usage, security, and scalability, they all share one purpose: to ensure that records are true and honest. Here’s an overview of some of the best known types of consensus mechanisms used by distributed systems to reach consensus.

## Types of Consensus Mechanisms



### Proof of Work (PoW)

Used by Bitcoin, Ethereum, and many other public blockchains, proof of work (PoW) was the very first consensus mechanism created. It is generally regarded to be the most reliable and secure of all the consensus mechanisms, though [concerns over scalability](https://cointelegraph.com/news/proof-of-work-vs-proof-of-stake-for-scaling-blockchains) are rife. While the term ‘proof of work’ was first coined in the early 1990s, it was Bitcoin founder Satoshi Nakamoto that first applied the technology in the context of digital currencies.

In PoW, miners essentially compete against one another to solve extremely complex computational puzzles using high-powered computers. The first to come up with the 64-digit hexadecimal number (‘hash’) earns the right to form the new block and confirm the transactions. The successful miner is also rewarded ​​with a predetermined amount of crypto, known as a ‘block reward’.

As it requires large amounts of computational resources and energy in order to generate new blocks, the operating costs behind PoW are notoriously high. This acts as a barrier of entry for new miners, leading to concerns about centralisation and scalability limitations.

And it’s not just the costs that are high. The most common criticism of PoW is the impact the electrical consumption has on the environment. This has led many to seek more sustainable, energy-efficient consensus protocols, such as proof of stake (PoS).

### Proof of Stake (PoS)

As the name suggests, this popular method of consensus revolves around a process known as staking. In a proof of stake (PoS) system, miners are required to pledge a ‘stake’ of digital currency for a chance to be randomly chosen as a validator. The process is not unlike a lottery whereby the more coins you stake, the better your odds.

Unlike in PoW where miners are incentivised by block rewards (newly generated coins), those who contribute to the PoS system simply earn a transaction fee.

PoS is seen as a more sustainable and environmentally-friendly alternative to PoW, and one that’s more secure against 51% attack. However, as the system favours entities with a higher number of tokens, PoS has drawn criticism for its potential to lead to centralisation. Prominent PoS platforms include Cardano (ADA), Solana (SOL), and Tezos (XTC).

### Delegated Proof of Stake (DPoS)

A modification of the PoS consensus mechanism, delegated proof of stake (DPoS) relies upon a reputation-based voting system to achieve consensus. Users of the network ‘vote’ to select ‘witnesses’ (also known as ‘block producers’) to secure the network on their behalf. Only the top tier of witnesses (those with the most votes) earn the right to validate blockchain transactions.

To vote, users add their tokens to a staking pool. Votes are then weighted according to the size of each voter’s stake – so the more skin in the game, the more voting power. Elected witnesses who successfully verify transactions in a block receive a reward, which is usually shared with those who voted for them.

Witnesses in the top tier are always at risk of being replaced by those deemed more trustworthy and who therefore get more votes. They can even be voted out if they fail to fulfil their responsibilities or try to validate fraudulent transactions. This helps to incentivise witnesses to remain honest at all times, ensuring the integrity of the blockchain.

Though less prevalent than PoS, DPoS is regarded by many as being more efficient, democratic, and financially inclusive than its predecessor. It is used by Lisk (LSK), EOS.IO (EOS), Steem (STEEM), BitShares (BTS), and Ark (ARK).

### Proof of Activity (PoA)

Proof of activity (PoA)  is a hybrid of the PoW and PoS consensus mechanisms. It is used by the Decred (DCR) and Espers (ESP) blockchain projects.

In PoA systems, the mining process begins like PoW, with miners competing to solve an elaborate mathematical problem using immense computing power. Once the block is mined, however, the system switches to resemble PoS, with the successfully generated block header being broadcast to the PoA network. A group of validators are then randomly selected to sign off on the hash, validating the new block. Like with PoS, the more crypto the validator holds, the higher their chances of being selected. Once every chosen validator has signed the block, it is added to the blockchain network and ready to record transactions. The block rewards are then shared among the miner and validators.

Though the PoA system was designed with the intention of combining the very best features of PoW and PoS, while avoiding their shortcomings, it has drawn criticism for its energy-intensive mining phase and inherent partiality towards validators holding a greater number of coins.

### Proof of Authority (PoA)

Not to be confused with proof of activity (also ‘PoA’), proof of authority (PoA) works by selecting its validators based on reputation. A modified version of PoS, it was proposed by Ethereum co-founder and former CTO Gavin Wood in 2017.

In PoA, validators don’t stake coins. Instead, they must put their reputations on the line for the right to validate blocks. This is very different from the majority of blockchain protocols which usually do not require you to reveal your identity to take part.

As this mechanism requires almost no computing power, it is far less resource-intensive than some of its predecessors, in particular PoW. It is also one of the less costly options, making it a heavily favoured solution for private networks, such as JP Morgan (JPMCoin). Other PoA-based projects include VeChain (VET) and Ethereum Kovan testnet.

Though highly scalable, where it compromises is in the decentralisation area, as only a select few can participate in the network. Additionally, the requirement for the validators to be identifiable also increases the risk of corruption and third-party manipulation.

### Proof of Burn (PoB)

Another more sustainable alternative to Bitcoin’s PoW algorithm is proof of burn (PoB). In PoB, miners gain the power to mine a block by ‘burning’ (destroying) a predetermined amount of tokens in a verifiable manner – namely, sending them to an ‘eater address’ where they cannot be recovered or spent. The more coins burned, the greater the chances of being randomly selected.

Unlike in PoS where miners are able to retrieve or sell their locked coins should they ever leave the network, burned coins are irretrievably lost. This method of requiring miners to sacrifice short-term wealth in order to gain the lifetime privilege to create new blocks helps to encourage long-term commitment from miners. The act of burning coins also leads to coin scarcity, limiting inflation and driving up demand.

Cryptocurrencies that use the proof of burn protocol include Slimcoin (SLM), Counterparty (XCP), and Factom (FCT).

### Proof of Capacity / Proof of Space (PoC / PoSpace)

Unlike the majority of its predecessors which grant mining rights based on computational power or coins staked, proof of capacity (PoC) – also known as proof of space (PoSpace) – bases its mining algorithm on the amount of space available in a miner’s hard drive.

In PoC, miners generate a list of all the possible hashes beforehand in a process called ‘plotting’. These plots are then stored on a hard drive. The more storage capacity a miner has, the more possible solutions. The more solutions, the higher the chances of possessing the correct combination of hashes and winning the reward.

As it doesn’t require expensive or specialised equipment, PoC opens up opportunities for the average person to participate in the network. As such, it is a less energy-intensive and more decentralised alternative to some of the more prevalent consensus mechanisms covered in this guide. However, as of yet, not many developers have chosen to adopt the system, and there are concerns about its susceptibility to malware attacks. The mechanism is currently used by Signum (SIGNA) – formerly Burstcoin (BURST), Storj (STORJ), and Chia (XCH).

### Proof of Elapsed Time (PoET)

Usually used on permissioned blockchain networks (those that require participants to identify themselves), proof of elapsed time (PoET) leverages trusted computing to enforce random waiting times for block construction. It was developed by Intel in early 2016 and is based on a special set of CPU instructions called Intel software guard extensions (SGX).

A time-lottery-based consensus algorithm, PoET works by randomly assigning different wait times to every node in the network. During the waiting period, each of these nodes goes to ‘sleep’ for that specified duration. The first to wake up (that is, the one with the shortest waiting time) is awarded the mining rights. This randomisation guarantees that every participant is equally as likely to be the winner, ensuring fairness within the network.

The PoET consensus mechanism is highly efficient, less resource-intensive, and scalable. It has been implemented in Hyperledger’s Sawtooth.

### Proof of History (PoH)

As the name suggests, proof of history (PoH) provides proof of historical events. Developed by Solana, PoH  allows for ‘timestamps’ to be built into the blockchain itself, verifying the passage of time between transactions without having to rely on other nodes.

This timestamping method is enabled by what’s known as a SHA-256, sequential-hashing verifiable delay function (VDF). It works by taking the output of a transaction and using it as input for the next hash, which enables everyone to clearly see which event took place in a particular sequence. As the VDFs can only be solved by a single CPU score, PoH severely reduces the processing weight of the blockchain, making it faster and more energy-efficient than many of his contemporaries.

As PoH is only employed by Solana, it has yet to be tested on a large scale.

### Proof of Importance (PoI)

First introduced by NEM (XEM), proof of importance (PoI) selects its miners based on certain criteria in a process called ‘harvesting’. Common factors include the number and size of transactions in the last 30 days, amount of vested currency, and network activity. It’s based on these factors that an importance score is attributed to nodes. The higher the score, the higher the probability of being chosen to harvest a block and receive the accompanying transaction fee.

Though similar to PoS, PoI’s use of additional metrics does away with the former’s tendency to inherently reward the rich by taking into account participants’ overall support of the network. As such, simply staking high in POI does not necessarily guarantee a chance of winning the block.

[Link1](https://crypto.com/university/consensus-mechanisms-in-blockchain) [Link2](https://www.geeksforgeeks.org/distributed-consensus-in-distributed-systems/) [Link3](https://coinmarketcap.com/alexandria/glossary/distributed-consensus) [Link4](https://www.techtarget.com/whatis/definition/consensus-algorithm)

**Cryptocurrency to Blockchain 2.0 :-**

[Link1](https://originstamp.com/blog/blockchain-1-vs-2-vs-3-whats-the-difference/)  [Link2](https://blockchain.oodles.io/blog/a-prefatory-guide-to-blockchain-2-0/)

**Permissioned Model of Blockchain :-**

A permissioned blockchain is a distributed ledger that is not publicly accessible. It can only be accessed by users with permissions. The users can only perform specific actions granted to them by the ledger administrators and are required to identify themselves through certificates or other digital means.

You might consider the addition of permissioned users as an extra blockchain security system. Administrators maintain an access control layer to allow certain actions to be performed only by certain identifiable participants. Records are kept within the blockchain of who is involved in the transactions. This makes permissioned blockchains different from public blockchains.

## Understanding Permissioned Blockchains

A [blockchain](https://www.investopedia.com/terms/b/blockchain.asp) can be built and accessed in multiple ways. Some blockchains need special permissions to read, access, and write information. Others only require that you have the ability to connect and can conduct work for the network. The intrinsic configuration of each blockchain controls the participants' transactions and defines their roles in which each participant can access and contribute to the blockchain.

It may also include maintaining the identity of each blockchain participant on the network. Such blockchains are called permissioned blockchains.

## Difference Between Permissionless and Permissioned Blockchains

Permissioned blockchains are similar to permissionless blockchains because they use the same technologies. However, permissioned blockchains do not allow users to access the blockchain without identification.

Much of the inner workings of the blockchains are the same. The key differences between them are:

* Enterprise vs. Public use
* Decentralization
* Development
* Transparency

### **Enterprise vs. Public Use**

[Bitcoin](https://www.investopedia.com/terms/b/bitcoin.asp), the most popular [cryptocurrency](https://www.investopedia.com/terms/c/cryptocurrency.asp) permissionless blockchain, allows anyone to participate in the network in the capacity of a [full node](https://www.investopedia.com/news/running-full-bitcoin-node-investors/) or a contributing miner. Anyone can take a read-only role or make legitimate changes to the blockchain, like adding a new block or maintaining a full copy of the entire blockchain.

As blockchain uses grow and mature, more businesses and governments have realized the cost-saving benefits the technology introduces into an organization. As a result, permissioned blockchains have become popular among industry-level firms and enterprises, for which security, identity, and role definition are essential.

For instance, a manufacturer producing a product may use a permissioned blockchain that ties into its [supply chain management](https://www.investopedia.com/terms/s/scm.asp). The transactions on this blockchain would likely involve logistics partners, financing banks, and other vendors involved in the supply and financing processes. Each entity would have its own level of transparency and permissions that it could use to streamline operations, track inventory, or monitor spending and invoicing.

### **Decentralization**

Permissionless blockchains have a broad decentralization in that they allow for more users and can extend across a much larger network. On the other hand, permissioned blockchains have limited decentralization as they are generally used for enterprise and business purposes, requiring various amounts of centralization.

### **Development**

Generally, permissionless blockchains are open source, which means that a community develops them; they can be changed and used by anyone. Permissioned blockchains are generally proprietary and controlled by the developers or the business using them.

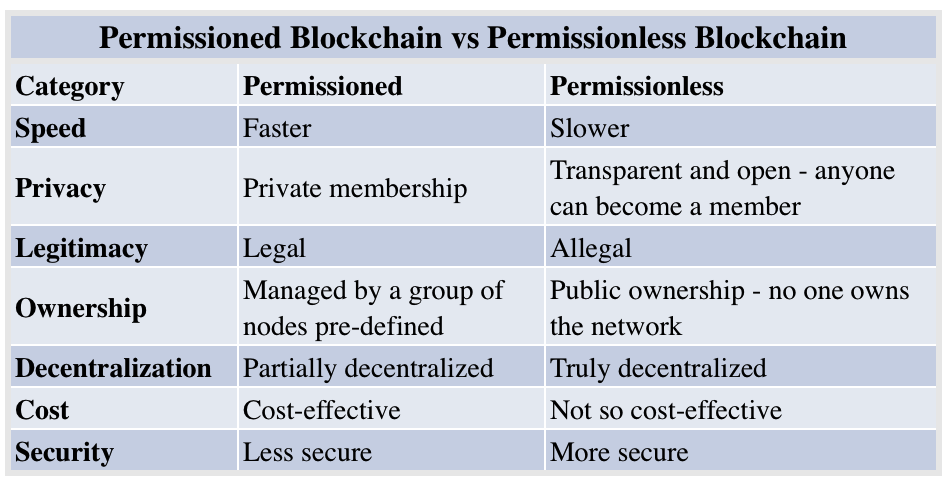
A developer building a permissioned blockchain may opt to make a few select records, like product name and quantity involved in a transaction, available for everyone to read. However, only select participants are allowed to view the transaction price. Other implementations may include limiting participants to act as nodes on the network, enhancing the network's security.

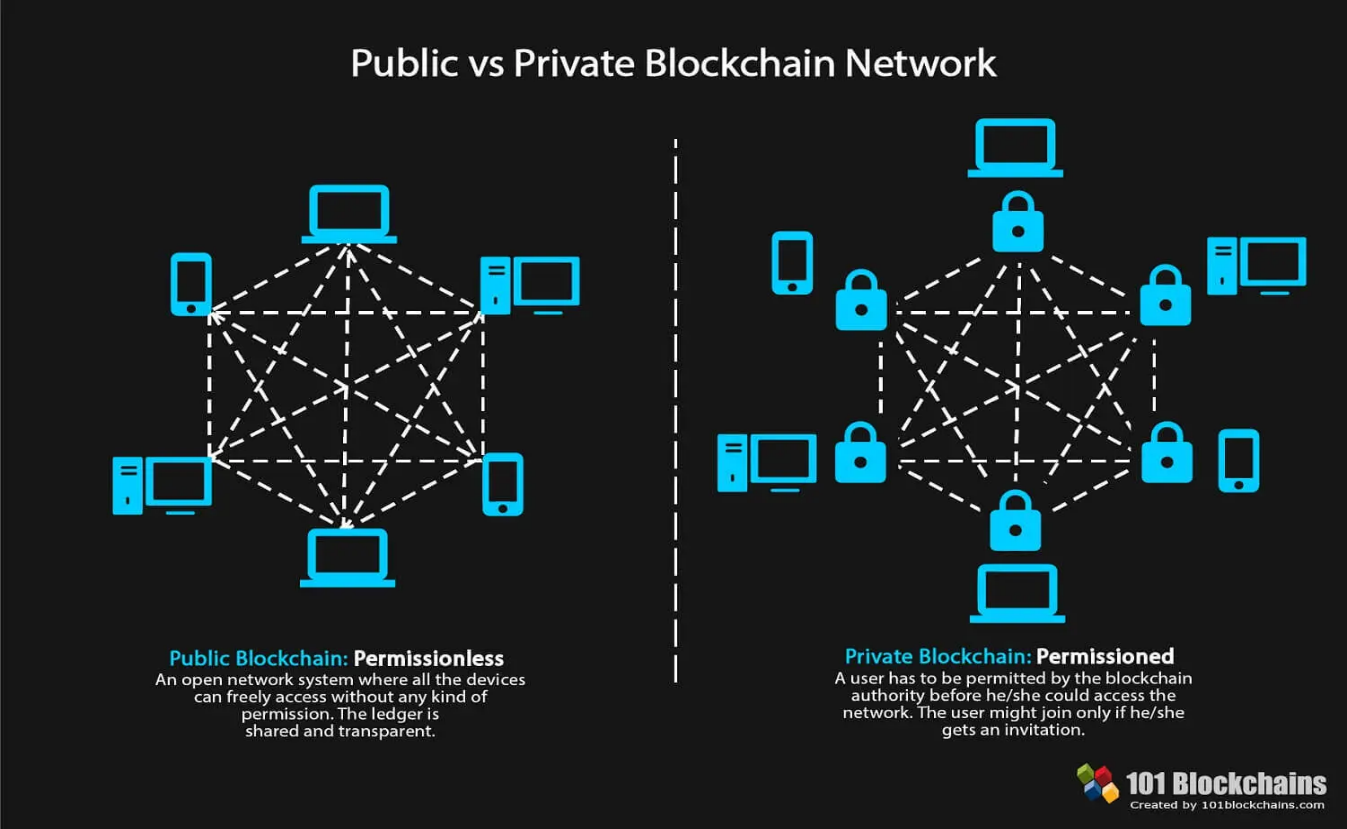
All such permissioning and profile maintenance is handled by this access-control layer. These differ from the un-permissioned or public blockchain networks, which don’t have a control layer.

### **Transparency**

Permissionless blockchains are much less transparent since they provide a certain amount of anonymity for the users. Wallet addresses cannot generally be tracked back to the blockchain users, and transactions are encrypted using various cryptography methods.

Permissioned blockchains require more transparency at certain levels since they are used for conducting business. Nodes, or the users and their connections, are known and their transactions are visible. Among many other benefits, this can defend a company against double invoicing, spending, paying, or any other number of errors that can be made using enterprise management programs.



****

[**Link1**](https://www.investopedia.com/terms/p/permissioned-blockchains.asp) [**Link2**](https://101blockchains.com/permissioned-blockchain/) [**Link3**](https://www.blockchain-council.org/blockchain/permissioned-and-permissionless-blockchains-a-comprehensive-guide/)

**Blockchain Hash Function :-**

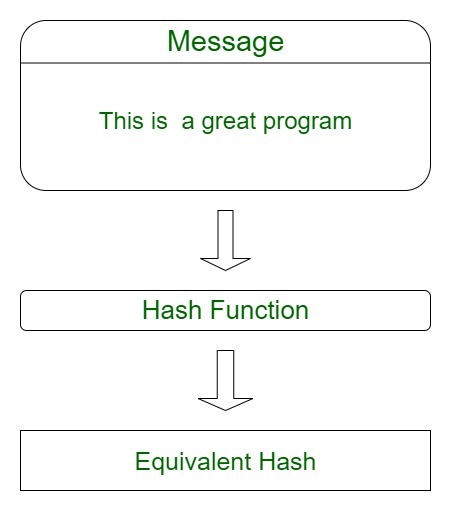
A hash function is a mathematical function that takes an input string of any length and converts it to a fixed-length output string. The fixed-length output is known as the hash value. To be cryptographically secure and useful, a hash function should have the following properties:

* **Collision resistant:**Give two messages m1 and m2, it is difficult to find a hash value such that hash(k, m1) = hash(k, m2) where k is the key value.
* **Preimage resistance:**Given a hash value h, it is difficult to find a message m such that h = hash(k, m).
* **Second preimage resistance:**Given a message m1, it is difficult to find another message m2 such that hash(k, m1) = hash(k, m2).
* **Large output space:**The only way to find a hash collision is via a brute force search, which requires checking as many inputs as the hash function has possible outputs.
* **Deterministic:**A hash function must be deterministic, which means that for any given input a hash function must always give the same result.
* **Avalanche Effect:**This means for a small change in the input, the output will change significantly.
* **Puzzle Friendliness:**This means even if one gets to know the first 200 bytes, one cannot guess or determine the next 56 bytes.
* **Fixed-length Mapping:**For any input of fixed length, the hash function will always generate the output of the same length.

### How do Hash Functions work?

The hash function takes the input of variable lengths and returns outputs of fixed lengths. In cryptographic hash functions, the transactions are taken as inputs and the hash algorithm gives an output of a fixed size.

The below diagram shows how hashes work.



### Uses of Hash Functions in Blockchain

The blockchain has a number of different uses for hash functions. Some of the most common uses of the hash function in blockchain are:

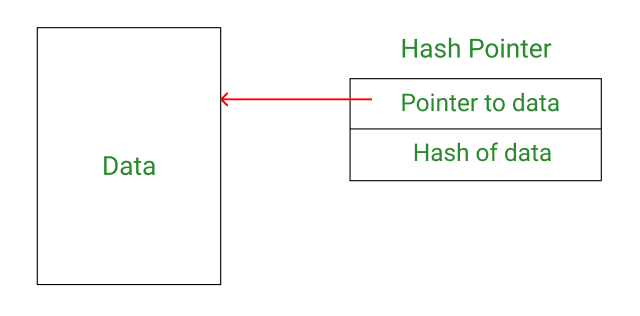
* **Merkle Tree:**This uses hash functions to make sure that it is infeasible to find two Merkle trees with the same root hash. This helps to protect the integrity of the block header by storing the root hash within the block header and thus protecting the integrity of the transactions.
* **Proof of Work Consensus:**This algorithm defines a valid block as the one whose block header has a hash value less than the threshold value.
* **Digital signatures:**Hash functions are the vital part of digital signatures that ensures data integrity and are used for authentication for blockchain transactions.
* **The chain of blocks:**Each block header in a block in the blockchain contains the hash of the previous block header. This ensures that it is not possible to change even a single block in a blockchain without being detected. As modifying one block requires generating new versions of every following block, thus increasing the difficulty.

Thus, it can be concluded hash functions are a vital part of the blockchain technology used to protect the integrity and immutability of the data stored on the blockchain.

[Link1](https://www.geeksforgeeks.org/blockchain-hash-function/)  [Link2](https://www.investopedia.com/terms/h/hash.asp#:~:text=In%20particular%2C%20cryptographic%20hash%20functions,hash%20function%20from%20its%20output.)  [Link3](https://resources.infosecinstitute.com/topic/hash-functions-in-blockchain/)

**Hash Pointer :-**

A regular [pointer](https://www.geeksforgeeks.org/pointers-in-c-and-c-set-1-introduction-arithmetic-and-array/) stores the memory address of data. With this pointer, the data can be accessed easily. On the other hand, a hash pointer is a pointer to where data is stored and with the pointer, the cryptographic hash of the data is also stored. So a hash pointer points to the data and also allows us to verify the data. A hash pointer can be used to build all kinds of data structures such as **blockchain and Merkle tree.**



# **Blockchain Merkle Tree**

Merkle tree is a fundamental part of blockchain technology. It is a mathematical **data structure** composed of hashes of different blocks of data, and which serves as a summary of all the transactions in a block. It also allows for efficient and secure verification of content in a large body of data. It also helps to verify the consistency and content of the data. Both Bitcoin and Ethereum use Merkle Trees structure. Merkle Tree is also known as **Hash Tree**.

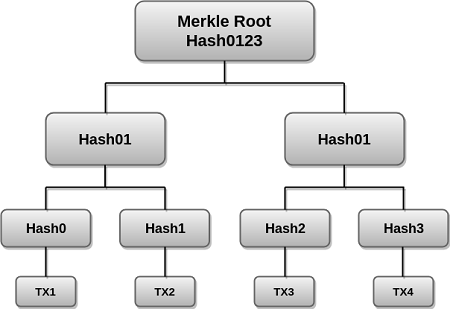
The concept of Merkle Tree is named after **Ralph Merkle**, who patented the idea in **1979**. Fundamentally, it is a data structure tree in which every **leaf node** labelled with the hash of a data block, and the **non-leaf node** labelled with the cryptographic hash of the labels of its child nodes. The leaf nodes are the lowest node in the tree.

## How do Merkle trees work?

A Merkle tree stores all the transactions in a block by producing a digital fingerprint of the entire set of transactions. It allows the user to verify whether a transaction can be included in a block or not.

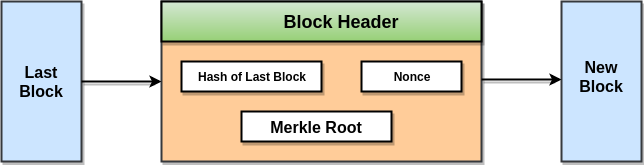
Merkle trees are created by repeatedly calculating hashing pairs of nodes until there is only one hash left. This hash is called the Merkle Root, or the Root Hash. The Merkle Trees are constructed in a bottom-up approach.

Every leaf node is a hash of transactional data, and the non-leaf node is a hash of its previous hashes. Merkle trees are in a binary tree, so it requires an even number of leaf nodes. If there is an odd number of transactions, the last hash will be duplicated once to create an even number of leaf nodes.



The above example is the most common and simple form of a Merkle tree, i.e., **Binary Merkle Tree**. There are four transactions in a block: **TX1**, **TX2**, **TX3**, and **TX4**. Here you can see, there is a top hash which is the hash of the entire tree, known as the **Root Hash**, or the **Merkle Root**. Each of these is repeatedly hashed, and stored in each leaf node, resulting in Hash 0, 1, 2, and 3. Consecutive pairs of leaf nodes are then summarized in a parent node by hashing **Hash0** and **Hash1**, resulting in **Hash01**, and separately hashing **Hash2** and **Hash3**, resulting in **Hash23**. The two hashes (**Hash01** and **Hash23**) are then hashed again to produce the Root Hash or the Merkle Root.

Merkle Root is stored in the **block header**. The block header is the part of the bitcoin block which gets hash in the process of mining. It contains the hash of the last block, a Nonce, and the Root Hash of all the transactions in the current block in a Merkle Tree. So having the Merkle root in block header makes the transaction **tamper-proof**. As this Root Hash includes the hashes of all the transactions within the block, these transactions may result in saving the disk space.



The Merkle Tree maintains the **integrity** of the data. If any single detail of transactions or order of the transaction's changes, then these changes reflected in the hash of that transaction. This change would cascade up the Merkle Tree to the Merkle Root, changing the value of the Merkle root and thus invalidating the block. So everyone can see that Merkle tree allows for a quick and simple test of whether a specific transaction is included in the set or not.

[Link1](https://www.javatpoint.com/blockchain-merkle-tree)  [Link2](https://www.geeksforgeeks.org/blockchain-merkle-trees/#:~:text=On%20the%20other%20hand%2C%20a,as%20blockchain%20and%20Merkle%20tree.)