## **How blockchains work**

A blockchain is a shared database that is managed by a global network of computers. Information held in the database is distributed and continually reconciled by the computers in the network. The computers are often referred to as *nodes, miners,* or *peers.* Whatever they’re called, the computers are creating and maintaining their blockchain by validating and transmitting entries. And *entries* are the data that is published by the users of the network.

“Blockchain Basics:

Bitcoin’s transaction history is stored in a data structure called a blockchain. The blockchain — logically — is made up of a chain of blocks. Blocks are produced roughly every ten minutes and are made up of transactions. Transactions are transfers of Bitcoin from one account to another broadcasted on the network. Transactions within the blocks are ordered by the miners (block creators) according to the optional fee a user includes as an incentive. The higher the fee, the more likely the transaction is included. Anyone can run a mining node. Every miner has a copy of the same blockchain. The act of creating a block is called mining.

”

https://medium.com/coinmonks/simply-explained-why-is-proof-of-work-required-in-bitcoin-611b143fc3e0

Often this data represents the movement of cryptocurrency from one user of the network to another user of the network.(not always)

The computers in the network are preventing you from “double-spending.” Double spending is a scenario in which a Bitcoin owner illicitly spends the same bitcoin twice. there is a risk that the holder could make a copy of the digital token and send it to a merchant or another party while retaining the original.

Decentralized:

Most blockchains are not controlled by any single entity and do not have a single point of failure. Blockchain was designed to be decentralized and distributed across a large network of computers. This decentralizing of information reduces the ability for data tampering.

All the entries are viewable by the whole network.

When data is entered into a blockchain, it can’t be removed. It’s there forever.

Why?

Now imagine a whole suite of incorruptible digital ledgers of economic transactions that can be programmed to record and track not only financial transactions but also virtually everything of value. The blockchain can track things like medical records, land titles, and even voting.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Each block -> each record

BLock has:

1. Relevant info (bitcoin -from,to)
2. Hash (Unique has code for the block)
3. Prev Hash

FIrst block is The Genesis block - > This wont have a previous hash

> Can track History

> Incase you change data, you have to change the hash, followed by the previous hash value for every block….this is nearly impossible and hence secure.

>Adding data: Consensus Rule - Needs more vote

>Is hashing enough?

Using hashing isn’t enough to prevent tampering. That’s because computers these days are super fast, and they can calculate hundreds of thousands of hashes per second. Technically, a hacker can change the hash of a specific block and then calculate and change all the hashes of the following blocks in order to hide the tampering.

So?

On top of the hashes, blockchains have additional security steps including things like proof-of-work and peer-to-peer distribution.

A *proof-of-work* (PoW) is a mechanism that slows down the creation of the blocks. In Bitcoin’s case, for example, it takes about ten minutes to calculate the required PoW and add a new block to the chain.

* A cryptographic puzzle must be solved to create the new block.
* The computer that solves the puzzle shares the solution with all the other computers in the network.
* Finally, all the computers involved in the network verify the proof-of-work. If 51 percent of the network testifies that the PoW was correct, the new block is added to the chain.

A third way blockchains secure themselves is by being distributed. Blockchains don’t use a central entity to manage the chain. Instead, they use a peer-to-peer (P2P) network. In public blockchains like Bitcoin, everyone is allowed to join. Each member of the network is called a validator or a node. When someone joins the network, they get the full copy of the blockchain. This way, the node can verify that everything is still in order.

A smart contract is a digital computer program stored inside a blockchain. It can directly control the transfer of cryptocurrencies or other digital assets based on certain conditions.

Use Cases:

Real Estate

Healthcare industry

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A cryptocurrency, broadly defined, is virtual or digital money that takes the form of tokens or “coins.” While some cryptocurrencies have ventured into the physical world with credit cards or other projects, the large majority remain entirely intangible.

Bitcoin:

Purely peer-to-peer version of electronic money.

TAsk:

* Use cases of blockchain.
* Why do we need blockchain?
* What is crypto wallet and account
* Private Key and Public key
* Symmetric and Asymmetric Encryption - CNS
* Disadvantages of crypto transaction
* Why is bitcoin’s value high?
* Consensus Algorithm
* FLow of blockchain ….flow of adding the blocks

**Applications of blockchain.**

* Secure sharing of medical data:
* NFT marketplaces: NFTs transform digital works of art and other collectibles into one-of-a-kind, verifiable assets that are easy to trade on the blockchain.
* Music royalties tracking: With blockchain, musicians are able to receive equitable royalty payments.
* Through **smart contracts** (automated programs on the blockchain), musicians could be paid for fractions of a cent each time a listener presses play on one of their songs, allowing the process of royalty payments to take place in seconds instead of multiple months.
* Cross-border payments:
* A cross border payment refers to **a transaction involving individuals, companies, banks or settlement institutions operating in at least two different countries** that don't necessarily share a border.
* Real-time IoT operating systems
* **blockchain to store IoT data adds another layer of security to prevent malicious attackers from gaining access to the network.**
* Personal identity security
* Anti-money laundering tracking system
* Supply chain and logistics monitoring
* Voting mechanism
* Advertising insights
* Original content creation
* Cryptocurrency exchange
* Real estate processing platform

**Why do we need blockchain?**

* When you want to manage and secure digital relationships or keep a decentralised, shared system of record. Anytime you want to keep a long-term, transparent record of assets (for example, to record property or land rights), blockchain could be the ideal solution. ‘Smart contracts’, in particular, are great for facilitating digital relationships and transactions. With a smart contract, automated payments can be released when parties in a transaction agree that their conditions have been met.
* Anywhere a middleman or gatekeeper function is expensive or time-consuming. For example, most accommodation providers currently have to interact with guests via a centralised aggregator platform, like Airbnb or Expedia (who, in turn, take a cut on each booking). Blockchain could change all that. For example, travel company TUI is so convinced of the power of blockchain, it’s pioneering ways to connect hoteliers and customers directly, so that they can transact via blockchain in an easy, safe and consistent way, rather than via a central booking platform.
* When you want to record secure transactions, especially between multiple partners. A traditional database may be good for recording simple transactions between two parties, but when things get more complicated, blockchain can reduce bottlenecks and simplify relationships.
* Where the data is in constant flux, but you want to keep a record of past actions. Blockchain is a better, safer way to record activity and keep data fresh, while maintaining a record of its history. The data can’t be corrupted by anyone or accidentally deleted, and you benefit from both a historical trail of data, plus an instantly up-to-date record.

**When you shouldn't use blockchain:**

They’re not the ideal solution for super-fast digital transactions that take place in milliseconds.

blockchains are, by their very nature, open chains of information. So if anything confidential, then no…

Crypto Wallet:

<https://www.simplilearn.com/tutorials/blockchain-tutorial/blockchain-wallet#why_use_a_blockchain_wallet>

<https://cointelegraph.com/bitcoin-for-beginners/bitcoin-wallets-a-beginners-guide-to-storing-btc>

A cryptocurrency wallet is a device, physical medium, program or a service which stores the public and/or private keys for cryptocurrency transactions

The keys can be used to track ownership, receive or spend cryptocurrencies.[11] A public key allows others to make payments to the address derived from it, whereas a private key enables the spending of cryptocurrency from that address.

Crypto wallets keep your private keys – the passwords that give you access to your cryptocurrencies – safe and accessible, allowing you to send and receive cryptocurrencies like Bitcoin and Ethereum.

The device containing your Bitcoin wallet stores the private key, not the coins themselves.

**3 types of forms of wallet:**

**Paper wallet -** COntains keys and QR codes

**Hardware wallets:** Thumb drive and has to be plugged in the computer to use

**Online Wallets:** App, or browser that stores

Bitcoin wallets come in two broad categories: **Hot storage or cold storage wallets.**

**Hot storage:** Hot storage cryptocurrency wallets are directly connected to the internet, such as a phone app, a desktop software program or an online provider .(coinbase)

**A cold wallet** is a small, encrypted portable device that allows you to download and carry your bitcoin and other cryptocurrencies.

**Ledger Nano X - hardware - 10k**

[**Trezor Model**](https://takethisbestdeal.com/pn92)

Anyone might easily follow your whole payment history if you use the same public key every time you receive Bitcoin. Treating keys as one-time-use tokens increases a user’s privacy significantly.

**What determines the cost of 1 Bitcoin?**

* The supply of Bitcoin and the market's demand for it
* The cost of producing a bitcoin through the mining process
* The rewards issued to Bitcoin miners for verifying transactions to the blockchain
* The number of competing cryptocurrencies

Bitcoin’s value may increase over time if demand for it grows, but its value may also go to zero at any time if individuals and firms stop accepting it in exchange for goods and services or other forms of money.

**Risks of transaction:**

**Security Risk:**

Transactions made with **bitcoin cannot be reversed or canceled**.

If a thief gains access to a Bitcoin owner's computer hard drive and steals their private encryption key, they could transfer the stolen Bitcoin to another account.

**MArket Risk:** The price of cryptocurrency fluctuates constantly.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Consensus Theorem:**

THis is how a decentralised , no authoritative figure makes decisions.To ensure records are true and honest. To decide on which peer adds the block to the blockchain.

If everyone were able to create blocks and add it , it would be flooded.

So it takes some hard computation to achieve it(roughly 10mins)

EAch miner has a set of unconfirmed set of transactions they receive, to be added to the blockchain

**Proof of Work:(Bitcoin/Ethereum)**

All the miners compete to solve a mathematical problem to find the hash. DOne by powerful computer(Try to find the input/hash that would yield a particular output)

They will try to find the nonce to get the hashed output.

Algorithms - for miners.

Then once found, he will broadcast it to other, they will verify and agree

The miner who cracks it get the reward of new bitcoins and any transactional fee available from the previously added block.

**Proof of Stake: (Cardano/EOS)**

The validator is chosen based on the amount of stake placed by them.

**Proof of Authority:**

Group of people are chosen from a pool.

**FLow of blockchain ….flow of adding the blocks…steps**

<https://blog.goodaudience.com/how-a-miner-adds-transactions-to-the-blockchain-in-seven-steps-856053271476#92ef>

Step1: A user wants to make a transaction from his wallet to someone else

Step2: The transaction is broadcasted by the wallet application and is now awaiting to be picked up by a miner on the according blockchain.

Step3: A miner will pick it up(More chances if the the user offers higher mining reward for faster processing)

Step4: Selected transactions are added to the miner’s bloc. This is added to the blockchain. To do this, needs a signature(proof of wrk)

Miners repeat this process of changing the nonce indefinitely until they randomly hit an output string that meets the signature requirements.

* **Step 5**: The miner that finds an eligible signature for its block first, broadcasts this block and its signature to all the other miners.

Step6: Other minors confirm the block

STep7: After a block has been added to the chain, every other block that is added on top of it counts as a ‘*confirmation’* for that block.

**Task2:**

* ~~Supply Chain Management - Use case of blockchain~~
* ~~Zero Proof of knowledge~~
* Algorithms - used miners.
* HD Wallet
* ~~SegWit and Legacy~~
* ~~Torrent working~~
* ~~BTC mining - working~~
* Wallet - processing
* Disadvantages of transactions- Double spending,
* Disadvantages of blockchain - eg:high power
* ~~P2P - Torrent.~~
* ~~Contents inside a block.~~

**Supply Chain Management:**

Supply chain management (SCM) is the centralized management of the flow of goods and services and includes all processes that transform raw materials into final products.

Supply chain management has five key elements—planning, sourcing raw materials, manufacturing, delivery, and returns.

Advantages:

* better ability to predict and meet customer demand;
* fewer process inefficiencies and less product waste;

Smart Contracts: We can create a global ledger, where all the information about the products is put. Each component would have an entry as a block in the blockchain. In this way, you can trace where your products (each part of it )came from.

Manage and Monitor, Avoid Risks, Trust the stakeholders, Track the flow, track the quality at real time, Also pay autonomously using smart contracts. ---> Increased Transparency.

**Zero Proof of Knowledge:**

Zero-knowledge proof is an encryption scheme whereby one party (the prover) can prove the truth of specific information to another party (the verifier) without disclosing any additional information.

It minimizes the proof that someone is lying to you.

**Types:** Interactive and Non-interactive

INteractive - takes some effort for interrogation back and forth for proving

Non Interactive - Proof that can be verified by themselves.

(Eg: ZCash uses this to hide all the transactions unline Bitcoin)

ZK-SNARKS(Succinct Non-interactive Arguments of Knowledge)

Disadvantage - Loss of data( If they die, only the proof exists, not the data)

**HD Wallet:**

A hierarchical deterministic (HD) wallet is a [digital wallet](https://www.investopedia.com/terms/d/digital-wallet.asp) commonly used to store the digital keys for holders of cryptocurrencies such as Bitcoin and Ethereum.

*A hierarchical-deterministic (HD) wallet generates a new key pair from a master key pair for each crypto transaction to enhance privacy and security.*

Deterministic wallets were created to offer a solution, one in which all keys can be traced back to an original random seed(a mnemonic word), With a deterministic wallet, the original seed is enough to recover all private and public keys, therefore requiring only a single backup at the time of creation.

HD Wallet contain keys in a tree structure, in which parent keys can produce children keys.

By having multiple addresses, one would still not be able to know your exact balance.

For example, let’s say you run a blog and allow for Bitcoin donations. You’d share one of your addresses on your blog’s page so everyone could send you some BTC if they’d like. However, since the rest of your balance is on other addresses, they’d still have no idea how much BTC you own.

Another reason is increased security. As mentioned earlier, the one who has access to an address’s Private Keys would have access to the funds on an address. Since you’d have diversified your funds over multiple addresses, they’d need to get multiple private keys to get access to all your crypto assets.

Wallet Working:

When you create your Bitcoin wallet, a [seed](https://river.com/learn/terms/s/seed-bitcoin/) is created. Seeds are displayed in the form of a series of words, known as a [mnemonic phrase](https://river.com/learn/terms/m/mnemonic/). This seed is used to derive every individual Bitcoin key that you’ll use to send and receive bitcoin.

This design is known as a [Hierarchical Deterministic (HD)](https://river.com/learn/terms/h/hd-wallet/) framework, which is an industry standard for Bitcoin key generation and management.

Most wallets will automatically create new public keys each time you want to receive bitcoin. This alleviates the problem of public key or **address reuse**. If you use the same public key every time you receive bitcoin, it would become trivial for anyone to track your entire payment history. Treating keys as single use tokens greatly improves a user’s privacy.

Coinbase - Yet to explore

## Addresses vs. Public Keys

An address is derived directly from a public key through a hashing algorithm. Behind every address is a public key.

Addresses are shorter than public keys, and usually begin with a ‘1(legacy)’, ‘3(segwit)’, or ‘bc1’, while public keys begin with ‘02’, ‘03’, or ‘04’.

**Peer to Peer:**

The primary goal of peer-to-peer networks is **to share resources and help computers and devices work collaboratively**, provide specific services, or execute specific tasks. As mentioned earlier, P2P is used to share all kinds of computing resources such as processing power, network bandwidth, or disk storage space

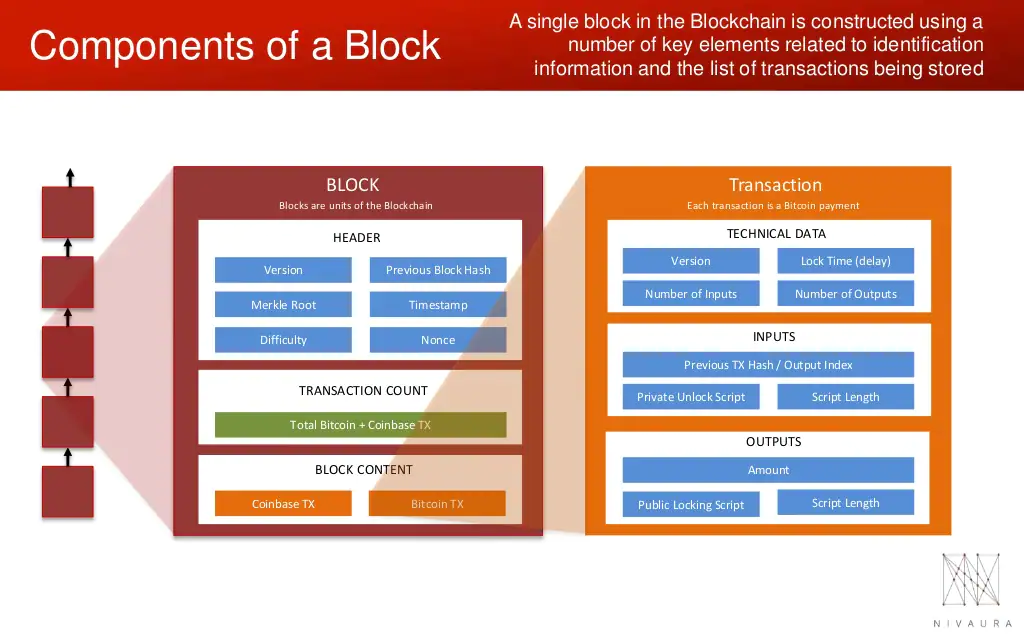
When you create an ad-hoc network between two computers, you create a peer-to-peer network between them.

Eg: Bit Torrent, Napster, Kazaa, (mp3), Limewire.

**Contents of a block:**

[**https://dev.to/damcosset/blockchain-what-is-in-a-block-48jo**](https://dev.to/damcosset/blockchain-what-is-in-a-block-48jo)

* Block contains more than 500 transactions on average. T
* The average size of a block seems to be 1MB and can go up to 8MB.
* One way to identify a block is using its hash. The block hash is a unique identifier.
* The other way to identify a specific block is the block height(not unique). The is the position of the block in the blockchain.



**Header:**

**3 metadata:**

1. The previous block hash& version -> Constant info
2. Timestamp (updated every 10 mins)

the nonce (Guessing number to meet certain length of leading zeros of new hash block)

the difficulty level (updated every 2016 block)

1. merkle tree root (updated every 10 mins)

(

* The Version: This is a version number to track the software and/or protocol upgrades.
* Timestamp: Seconds from Unix Epoch. When the block was created.
* Target: Proof of Work algorithm target for this block
* Nonce: Counter used for the Proof of Work algorithm

)

**Nonce:**

Nonce: Number only used once

The Nonce is a random whole number, which is a 32-bit (4 byte) field, which is adjusted by the miners, so that it becomes a valid number to be used for hashing the value of block.

**Merkle Root:**

****

The transactions in a block are contained in a structure called a merkle tree or binary hash tree.

A merkle tree is constructed by recursively hashing pairs of nodes ( in this case, transactions ), until there is only one hash, called the root or merkle root.

The cryptographic hash algorithm used is SHA256.

Double Hashing: const hA = sha256(sha256(tA))

**Process of Mining:**

1. **Creation of candidate block**(It is called candidate bcoz it has no proof of work yet)

* gathering the transactions in the transaction pool. It removes the transactions already present in the previous block, if there are any.

1. **Coinbase transaction**

-The coinbase transaction is just a transaction that the miner includes in the block he is mining. The miner indicates that he is paying himself. He calculates the transaction fees ..

*Reward for mining block + transactions fees*

When the block is validated, the miner receives the reward in its wallet.

1. **Construct the block header**

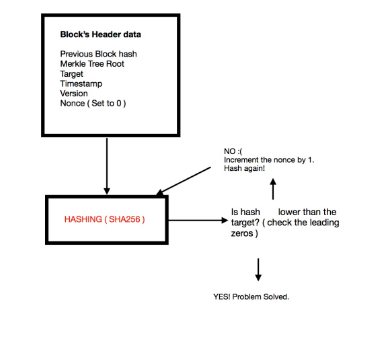
Version- Const value.

Timestamp - This is the number of seconds elapsed since the January 1, 1970.(UNIX EPOCH)

Target - the target is generated by the network and defines what makes a block's hash valid or not. The hash value should be lesser than the target.

Nonce - This is what the miner should guess to get the hash right.

1. Mining



Target is adjusted: The difficulty increases or decreases depending on the time it took to mine those blocks.

1. Once the hash is discovered, it is transmitted to all peers for validation. If the block is valid, the other miners will update their own copy of the blockchain with the new block. NOw this block will be used to mine the next block.

**Cons of mining:**

1. Shifting rewards - Even if the difficulty increases, the reward is halves since the bitcoin production will cap at 21 million.
2. Electricity Bill increases - not only for mining but also to keep the system from overheating
3. Efficiency - To solve the problem, the work done by your system
4. Time - a lot of time mining
5. Mining pool - altho it is speedy, the cost is shared between miners

**SegWit** is the **process by which the block size limit on a blockchain is increased by removing signature data from bitcoin transactions**.

Segregate means to separate, and witnesses are the transaction signatures. Hence, segregated witness, in short, means to separate transaction signatures.

Segwit removes scalability issue(no.of transactions per blocK issue), malleability issue (an attack that lets a person change a Bitcoin transaction’s unique ID before the transaction is confirmed on the Bitcoin network.)

Non Segwit block are called Legacy blocks. It is only 1mb that contains all info (inputs, outputs, sign and scripts).

Segwit blocks are a total of 4 mb with an extended block, by placing the signature in the extended block(witness data) and the input and output(sender and receiver info) will be placed in the base transaction block. Thus more transactions can fit in the 1mb.

Today >50% transaction is Segwit

Segwit is cheaper.

Steps: Create a segwit wallet, Transfer funds from legacy to segwit

**Double Spending:**

A double-spend is when the same unit of a digital currency is fraudulently spent more than one time.

Prevent:

P2P network - everyone knows the transactions

Nonce

**Algorithms used by miners:**

1. **SHA-256 -**  The cryptographic hash function that outputs a 256 bits long value. Bitcoin uses double SHA-256, meaning that it applies the hash functions twice.

In fact, it is nearly impossible to reveal the initial data from a hash value itself. Moreover, a brute force attack is extremely unlikely to succeed thanks to the astronomical number of potential combinations. In addition, it’s also severely unlikely that two data values (known as collision) have the same hash.

1. **Ethash** is currently a widely used algorithm. Among the projects that make use of this algorithm stand out Ethereum, [Ethereum Classic](https://academy.bit2me.com/en/what-is-ethereum-classic-etc-cryptocurrency/), Musicoin, Expanse, WhaleCoin.

Its main characteristics were its resistance to mining using ASIC devices, and for this it uses a structure that uses a **DAG (Directed Acyclic Graph)**.

1. [**Scrypt**](https://academy.bit2me.com/en/what-is-scrypt-hash-function/)it is key derivation function Password-based used in many cryptocurrencies to protect your systems from massive attacks. Upon entering the system, authorized users must perform this function only once. They don't need to spend too much time for that. However, in the event of a brute force attack, the algorithm begins to generate many pseudo-random numbers. This causes attackers to spend a large amount of RAM to continue their actions. As a result, any attack on a system using Scrypt is expensive and time consuming. These features make Scrypt perfect for resisting ASIC mining.
2. X11 - X11 consists of a group of 11 different hashing is considered as one of the safest mining algorithms that exist. [X11](https://academy.bit2me.com/en/what-is-x11-mining-algorithm/) It is intended to be friendly to CPU and GPU mining but has strong resistance to ASIC mining. minimizes electric power costs, achieving incredible performance for CPU and GPU mining. It is also an effective protection for miners using ASI.

### Equihash - for ZCash

1. [CryptoNight](https://academy.bit2me.com/en/what-is-the-algorithm-of-cryptonight-mining/) - It is a mining algorithm to provide a high level of security and anonymity.

**Day 3:**

* Crypto Exchanges - Coinbase. (How to buy a bitcoin)
* Proof of Work in----> Bitcoin
* ~~Custodial and noncustodial wallets~~
* ~~MemPool~~
* Multiple miners confirm one transaction at the same time.

https://criptomo.com/two-simultaneous-blocks/

* ~~Transaction: Nonces~~
* ~~Unspent Transaction (UTXO)~~
* ~~Forking.~~

**Crypto Exchanges:**

Cryptocurrency exchanges are platforms that facilitate the trading of cryptocurrencies for other assets, including digital and fiat currencies. In effect, cryptocurrency exchanges act as an intermediary between a buyer and a seller and make money through commissions and transaction fees.

Centralized cryptocurrency exchanges act as a third-party between a buyer and a seller.--->Coinbase

Decentralized cryptocurrency exchanges (DEX) allow users to execute peer-to-peer transactions without the need for a third party or an intermediary.

Decentralized exchanges do not allow for the trading of fiat currencies for digital ones

Decentralized exchanges do not require customers to fill out know-your-customer (KYC) forms, offering privacy and anonymity to users.

**How to buy Bitcoins:**

Once you’ve created an account, you may notice that the brokerage provides its own wallet. Many third-party exchanges have wallets — called “custodial wallets” — built into the platforms themselves

This is sometimes called “spot” buying, and consists of purchasing bitcoin by linking a bank account, debit/credit card or other payment method like PayPal to the exchange account. If you’re looking to purchase a large amount of bitcoin, consider a [hardware wallet](https://river.com/learn/terms/h/hardware-wallet/) or other cold storage option, which provide greater security than hot wallets and internet-connected storage devices.

## How to Send Bitcoin

### 

**Proof of Work :**

Data -> A

Find x

Find Hash of x = h(x)

Such that h(x)+A < B

**MemPool**

[**https://coinmarketcap.com/alexandria/glossary/mempool**](https://coinmarketcap.com/alexandria/glossary/mempool)

A mempool is essentially a cryptocurrency node's mechanism for storing information on unconfirmed transactions, acting as a waiting room for transactions that have not yet been included in a block.

When a node receives a new valid block, it ends up removing all of the transactions which are contained in the block from its mempool, as well as the transactions that have conflicting inputs. This can result in a sharp drop within the mempool size.

In fact, to submit a transaction to the miners, nodes have to relay it to each other until it has propagated across the entirety of the network.

**UTXO**

The term UTXO refers to the amount of digital [currency](https://www.investopedia.com/terms/c/currency.asp) someone has left remaining after executing a [cryptocurrency](https://www.investopedia.com/terms/c/cryptocurrency.asp) transaction such as [bitcoin](https://www.investopedia.com/terms/b/bitcoin.asp).

When a transaction is completed and there are outputs that aren't spent, they are deposited back into a database as inputs that can be used at a later date for a new transaction.

Received amounts (UTXOs) are used individually during a transaction and new outputs are created – one for the receiver and, if applicable, one for the amount that is left over (change output). The amount sent to the recipient becomes a new UTXO in the recipient’s address while the change output becomes a new UTXO in the sender’s address that may be used in a future transaction.

**Example:**

Alice owns a UTXO worth 1 BTC and wishes to pay Bob 0.4 BTC, she must spend the entire 1 BTC as an input. In order to send Bob exactly 0.4 BTC, Alice creates two outputs: the first to Bob, in the amount of 0.4 BTC, and the second back to herself, in the amount 0.59 BTC, assuming that she paid a 0.01 BTC transaction fee.

**Custodial and Non-Custodial Wallets:**

[**https://appinventiv.com/blog/custodial-vs-non-custodial-wallets/**](https://appinventiv.com/blog/custodial-vs-non-custodial-wallets/)

A non-custodial wallet is a decentralized type of wallet, where the customer owns its private keys.

A custodial wallet is a type of digital wallet which keeps a customer’s private keys and provides backup and security for your assets. Meaning, the third party has full control over your funds while you only have to give permission to send or receive payments.

(How non custodial works)

**Bitfinex, Bithumb, Cryptsy - all of them were hacked onced**

**Nonce: (In Etherium)**

**The nonce is the number of transactions sent from a given address.**

**Each time you send a transaction, the nonce value increases by 1. There are rules about what transactions are considered valid transactions, and the nonce is used to enforce some of these rules. Specifically:**

* **Transactions must be in order. You cannot have a transaction with a nonce of 1 mined before one with a nonce of 0.**
* **No skipping! You cannot have a transaction with a nonce of 2 mined if you have not already sent transactions with a nonce of 1 and 0.**

Why does it matter?

* **This value prevents double-spending, as the nonce will always specify the order of transactions.**

**Day 4:**

* **Bitcoin scripting**
* **multi-sig wallet**
* **Bitcoin units**
* **ECDSA - Elliptic curve digital currency**
* **Lightning network**
* **Proof of work - Architecture**
* **Generate public and private key using NodeJS**
* **Work flow in custodial and non-custodial wallet.**
* **How much money that orphaned miners will get?**

**BItcoin Scripting:**

Bitcoin Script is the language Bitcoin uses to do everything it can do, from sending funds from a wallet to allowing the creation of multi-user accounts.

It is this programming language that determines whether or not an operation can be performed.

This language is not [Full Turing](https://es.wikipedia.org/wiki/Turing_completo) because its functionality is limited and cannot loop(malicious parts of the program can be free to create complicated operations to consume the rate of [hash](https://academy.bit2me.com/en/what-is-hash/) and slow down the Bitcoin system through infinite loops)

## **Two types of script: scriptSig and scriptPubKey**

**The scriptSig** is the **unlock script**, which requires a public key and a digital signature (to verify the authenticity of such actions.)

The **scriptPubKey**, is the **blocking script**, which contains a public key hash, also called a **Bitcoin address**. The **scheduling of Bitcoin transactions** as such is stored in this part of the script.

**scriptPubKey** is a locking script placed on the output of a [Bitcoin](https://www.mycryptopedia.com/bitcoin-explained/) transaction that requires certain conditions to be met in order for a recipient to spend his/her bitcoins;

Conversely, **scriptSig** is the **unlocking script** that satisfies the conditions placed on the output by the scriptPubKey, and is what allows it to be spent; outside of code, scriptSig is also known as Signature scripts.

Both scriptPubKey and scriptSig are written in [Script](https://en.bitcoin.it/wiki/Script), the programming language used for constructing Bitcoin transactions.

OP\_CODES that in the operation are the following:

* OP\_DUP: Duplicate the item on the top stack.
* OP\_HASH160: The input is encoded twice: first with SHA-256 and then with RIPEMD-160.
* OP\_EQUALVERIFY: Verify that the data entered is correct and valid.
* OP\_CHECKSIG: The outputs, inputs, and script of the entire transaction are summarized in a hash. The signature used must be a valid signature for this hash and must be next to the public key.

Working of these scripts:

1. First, the original public key of the owner (which is in the scriptSig) of the funds is duplicated.
2. The duplicate public key then goes through a hashing process. In this process, a hash is applied first [SHA-256](https://academy.bit2me.com/en/sha256-bitcoin-algorithm/) and a RIPEMD-160 hash is then applied to the result.
3. The result of the hashes is compared with the hash of the public key that is in the scriptPubKey to make sure it is EQUALVERIFY (that is, it is the same key and is verified as valid).
4. If it matches, the script continues to run and CHECKSIG is performed to verify the signature with the public key.

**MultiSig Wallet:**

A multi-signature wallet ("multisig" for short) is [**a cryptocurrency wallet**](https://www.makeuseof.com/what-is-a-cryptocurrency-wallet/) that requires two or more private keys to sign and send a transaction. This type of digital signature makes it possible for two or more users to sign documents as a group. Co-owners and signatories to a shared multisig wallet are known as "copayers."

Commonly used types of MultiSig wallets:

* **n-of-n**: Transactions require more than one key to be authorized. All keys need to be used to create the signature. (2-of-2, 3-of-3, etc.).
* **n-of-m**: Transactions require some of the keys, but not necessarily all of them, to be authorized (1-of-2, 2-of-3, 3-of-5, etc.).

**2-of-2 MultiSig wallet**

Wallets protected with the two-factor authentication feature use the 2-of-2 multisignature algorithm. The idea is to keep private keys on two separate devices. For example, one private key is stored on a computer, the other on a mobile device. Transactions cannot be authorized without a signature from both devices. The 2-of-2 MultiSig wallet increases security, but at the risk of losing access to your funds if one of the devices is compromised.

**2-of-3 MultiSig wallet**

This type of MultiSig wallet requires 2 out of 3 existing private keys to authorize transactions. They are often used by exchanges to enhance the security of their hot wallets. An exchange that supports 2-of-3 MultiSig addresses holds one private key online and keeps the second one offline on an isolated device (sometimes called a “paper” backup). The third private key is stored by a separate security company. Since two separate entities are in charge of private keys, if one of them gets hacked, the wallet remains safe. The offline backup additionally secures the hot wallet in case the security partner goes out of business.

* **1-of-2 Multisig wallet**

MultiSig wallets can also be used to share funds among multiple users. If you want to share the use of funds in a single wallet with a trusted person, you can set up a wallet that allows any of your two keys to create the signature. But both keys are not necessary, meaning that both of you can operate with the funds independently.

**Unit of Bitcoin:**

The **satoshi** is currently the smallest unit of the bitcoin currency recorded on the [block chain](https://en.bitcoin.it/wiki/Block_chain).

one hundredth of a bitcoin -> 1 Satoshi.

**1 satoshi = 0.00000001 BTC.**

**1 RS(INR) -> 26 Satoshis.**

**Lightning Network:**

Lightning is a decentralized network using smart contract functionality in the blockchain to enable instant payments across a network of participants, also allows off-chain transactions ( transactions between parties not on the blockchain network).

This protocol enables the creation of a peer-to-peer payment channel between two parties, like between a customer and a coffee shop. Once established, the channel allows them to send an unlimited amount of transactions that are nearly instant as well as inexpensive. It acts as its own little ledger for users to pay for even smaller goods and services such as coffee without affecting the Bitcoin network.

To create a payment channel, the payer must lock a certain amount of Bitcoin into the network. Once the Bitcoin is locked in, the recipient can invoice amounts of it as they see fit. If the customer wants to keep the channel open, they can choose to add Bitcoin consistently.

The two parties can transfer funds between themselves indefinitely without telling the main blockchain. Because all transactions within a blockchain do not need to be approved by all nodes, this strategy substantially speeds up transaction times.

when the two parties decide to finish transacting, they can close the channel. All of the channel’s information is then consolidated into one transaction, which is sent to the Bitcoin mainnet for recording.

The Lightning Network creates a smart contract between two parties.

**EXAMPLE:**

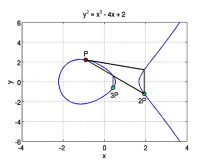
With the Lightning Network, Mike can open up a payment channel with the coffee shop. Each coffee purchase is recorded within that channel, and the shop still gets paid. The transaction is cheap or possibly even free, as well as instant. Then, when the Bitcoin that started the channel is spent, Mike can choose to close the channel or refill it. When a channel is closed, all of its transactions will then be recorded to the main Bitcoin blockchain.

**ECDSA:**

ECDSA stands for Elliptic Curve Digital Signature Algorithm (Elliptic Curve Digital Signature Algorithm). This system is used to create a digital signature that allows verification by third parties without compromising security.

ECDSA also uses what is known as elliptic curves over finite fields. Think of an elliptic curve group as a finite group of points on a curve where some operation is easy to perform in one direction but difficult to perform in the other direction.

In this way the signature can created for transactions using the private key and the public key can be generated from the private key not vice versa



On this curve,

a point is chosen at random ---> Origin

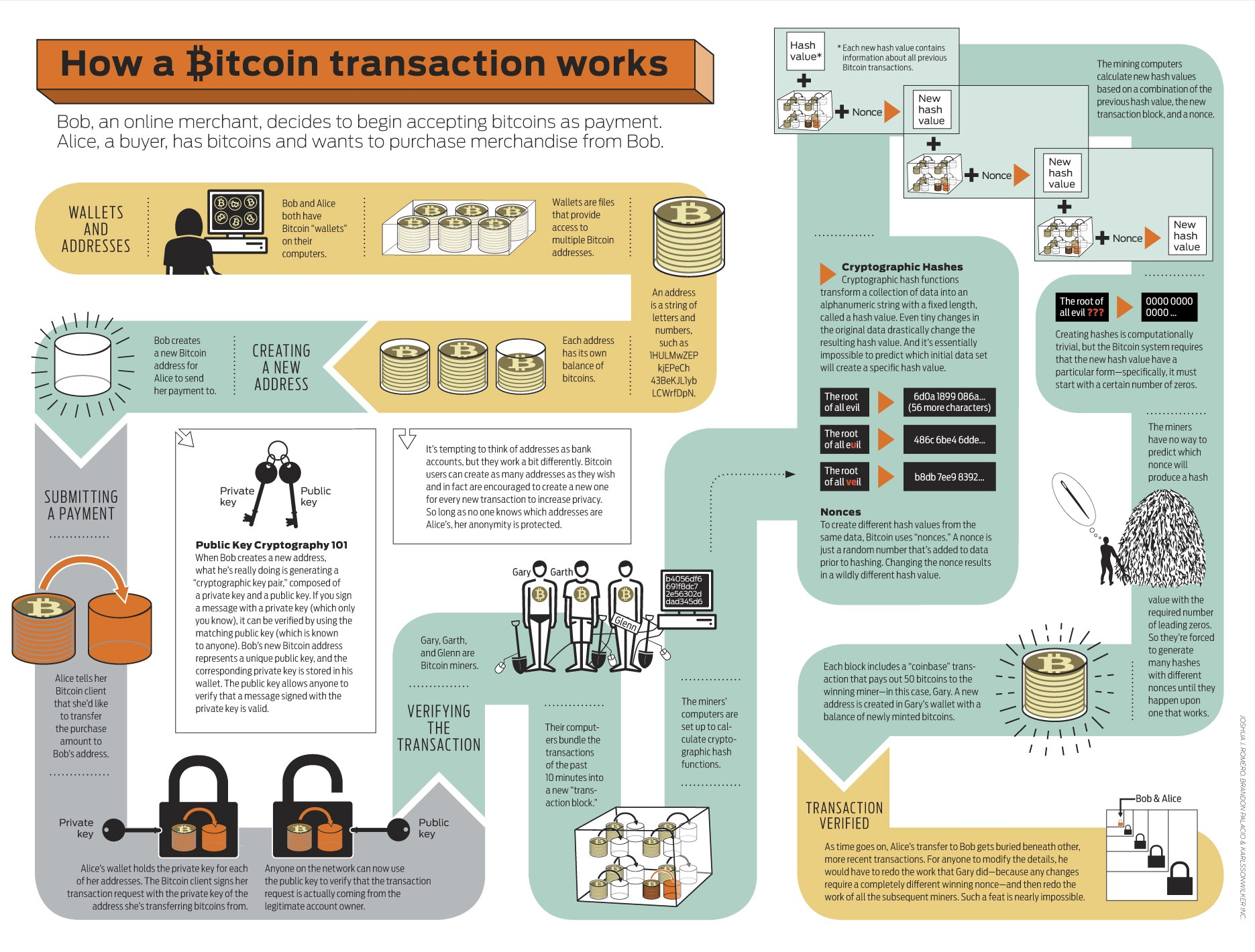
Next, a random number is generated ---> Private Key

Then, using the private key and the origin point, you perform another equation and we get a second point on the curve ---> public key.

It is precisely the use of this new equation together with the point of origin and the public key, which allows establishing the relationship between the public and private keys.

This process is considered safe, because at the moment it can only be done in one way. In other words, it is only possible to establish the mathematical relationship from a private to a public key, but not the opposite.

**FLow:**

****

Alice Wants to send Bob - 3 btc (She has 5 btc)

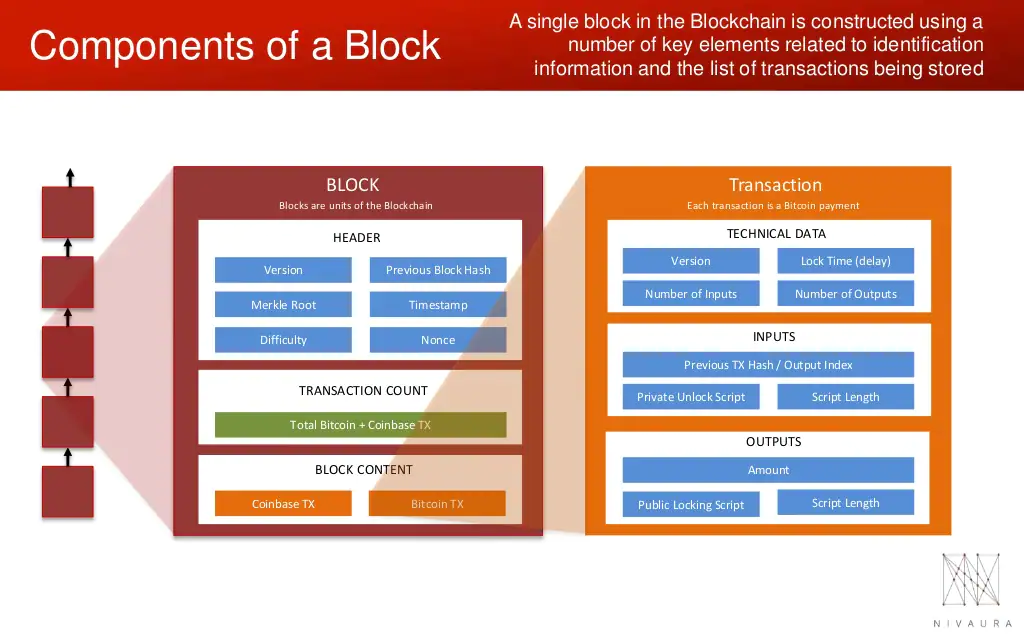
Transaction is made from sender to receiver with signature

The notification goes to to the blockchain network

Verification process authenticates the transaction

It goes to the mempool

It is in the form of UTXO, it is added to the block



**BLOCK CONTENT**

**Whats in a transaction (inside a block):**

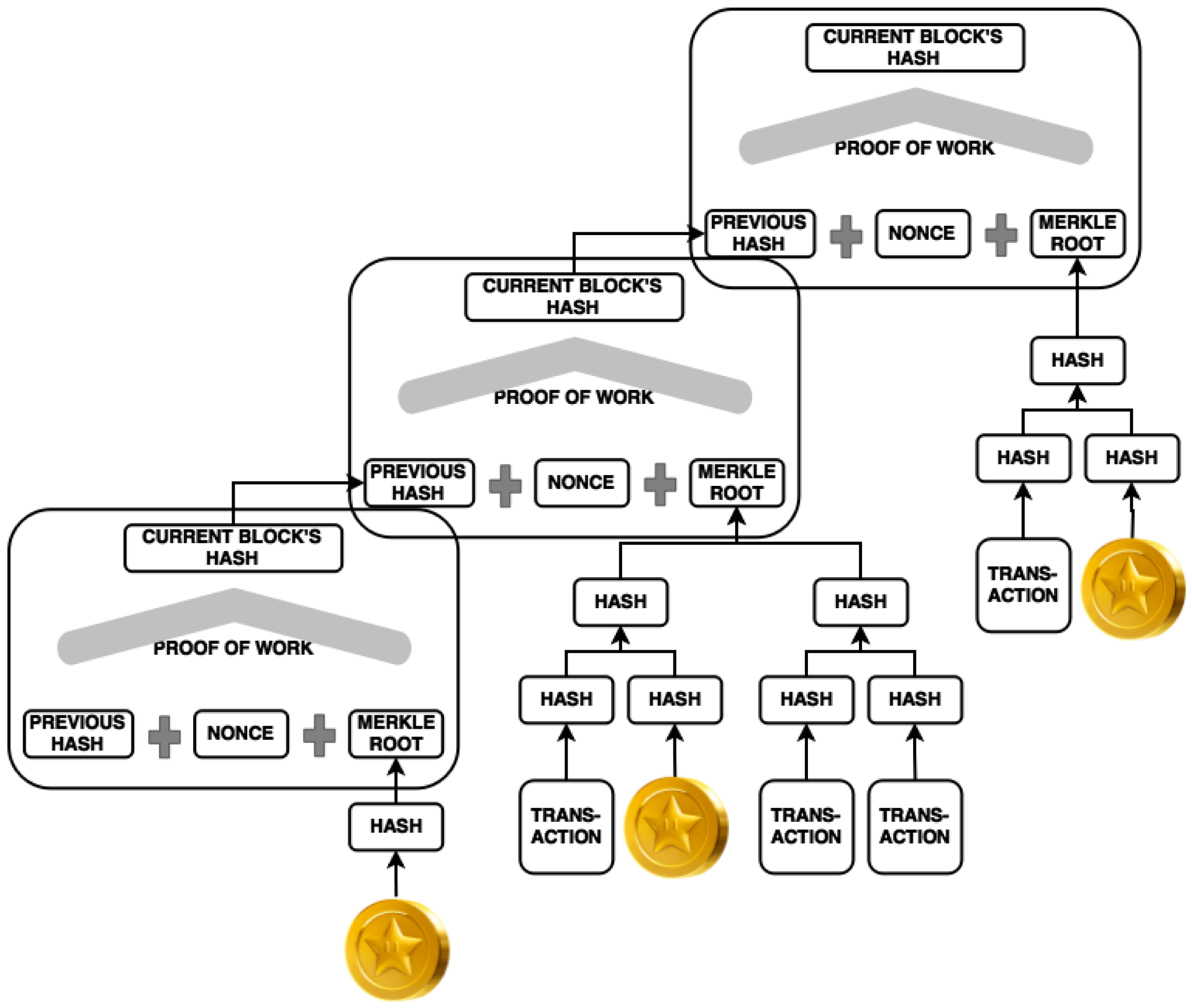
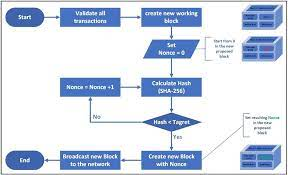
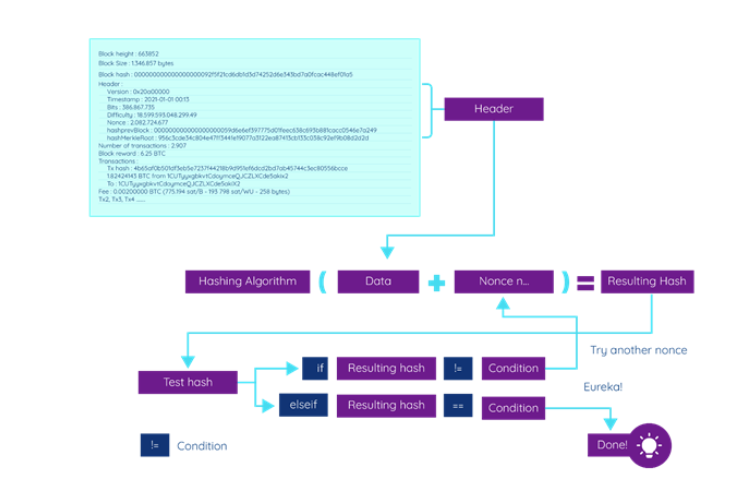
No. of inputs Inputs and Outputs -> WR to UTXO

Inputs and Output Info is added (Unlock script, Locking script, script length)

Along with the BTC transaction , we will have a **Coinbase transaction**.

**TRANSACTION COUNT:**

BTC + Transaction



PREV HASH + NONCE + MERKLE ROOT(generated with the set if transactions that the miner collected )

**Day 4:**

* Etherium - Why ?
* Consensus alg in Ethereum
* Etherium 2
* ERC 20 in Ethereum
* Create a transaction between A and B in TestNet.
* TapRoot.
* Turing complete and turing incomplete (For Bitcoin Script)

<https://medium.com/@claudio_69833/how-to-create-bitcoin-transactions-with-javascript-b3b43f53ca0c>

**Etherium:**

Ethereum is a blockchain platform with its own cryptocurrency, called **Ether** (ETH) or Ethereum, and its own programming language, called **Solidity.**

Unlike the Bitcoin blockchain, the Ethereum blockchain was not created to support a cryptocurrency. **The Ether cryptocurrency was created to provide an in-house currency for applications built on the Ethereum blockchain.** It allows software developers to create games and business applications, called dApps, and market them to users. Users pay fees to use dApps on the Ethereum platform. These fees are called "gas" because they vary depending on the amount of computational power used.

Unlike Bitcoin, there is no limit to the number of ETHs that can be created.

**Ethereum Virtual Machine:** Ethereum provides the underlying technology—the architecture and the software—that understands smart contracts and allows you to interact with it.

**Why Ethereum?**

Ethereum was created to enable developers to build and publish [smart contracts](https://www.investopedia.com/terms/s/smart-contracts.asp).

A smart contract is a simple computer program that facilitates the exchange of any valuable asset between two parties. It could be money, shares, property, or any other digital asset that you want to exchange. Anyone on the Ethereum network can create these contracts. The contract consists primarily of the terms and conditions mutually agreed on between the parties (peers)

Also, if you want to deploy a contract on Ethereum, you will need gas, and you would have to pay for that gas in ether. So gas is the execution fee paid by a user for running a transaction in Ethereum.

Bitcoin uses SHA-256, and **Ethereum uses Ethash.**

The average time taken on Bitcoin for mining a block is 10 minutes, whereas on Ethereum it is **12 to 15 seconds**. As of today, the mining reward for Bitcoin is 12.5 bitcoins; **for Ethereum it’s three ethers plus the transaction fee**—the cumulative transaction fees of all the transactions of a block.

1 Ether - 3 lakh

**APPLICATIONS**

* Voting Systems  
    
  As we’ve seen with DAO, voting systems are adopting Ethereum. The results of polls are publicly available, ensuring a transparent and fair democratic process by eliminating voting malpractices.
* Agreements  
    
  With Ethereum smart contracts, agreements can be maintained and executed without any alteration. So in an industry that has fragmented participants, is subject to disputes, and requires digital contracts to be present, Ethereum can be used as a technology for developing smart contracts and for digitally recording the agreements and the transactions based on them.

## **PROOF-OF-WORK**

Proof-of-work is the mechanism that allows the decentralized Ethereum network to come to consensus, or agree on things like account balances and the order of transactions. This prevents users from "double spending" their coins and ensures that the Ethereum chain is tremendously difficult to attack or manipulate.

**HOW DOES ETHEREUM'S PROOF-OF-WORK WORK?**

Ethereum transactions are processed into blocks. Each block has a:

block difficulty – for example: 3,324,092,183,262,715

mixHash – for example: 0x44bca881b07a6a09f83b130798072441705d9a665c5ac8bdf2f39a3cdf3bee29

nonce – for example: 0xd3ee432b4fb3d26b

**The work in proof-of-work**

The proof-of-work protocol, Ethash, requires miners to go through an intense race of trial and error to find the nonce for a block. Only blocks with a valid nonce can be added to the chain.

When racing to create a block, a miner will repeatedly put a **dataset**, that you can only get from downloading and running the full chain (as a miner does), through a mathematical function. The dataset gets used to generate a mixHash below a target nonce, as dictated by the block difficulty. The best way to do this is through trial and error.

The difficulty determines the target for the hash. The lower the target, the smaller the set of valid hashes.

Miners who successfully create a block get **rewarded with two freshly minted ETH and all the transaction fees within the block.**

A miner may also get 1.75 ETH for an uncle block.

**Etherium 2.0 - Vitalik Buterin**

Ethereum 2.0, also known as Eth2 or “Serenity,” is an upgrade to the Ethereum blockchain. The upgrade aims to enhance the speed, efficiency, and scalability of the Ethereum network so that it can process more transactions and ease bottlenecks.

While Ethereum 1.0 uses a consensus mechanism known as proof-of-work (PoW), Ethereum 2.0 will use a proof-of-stake (PoS) mechanism.

With Ethereum 1.0, the network can only support around 30 transactions per second; this causes delays and congestion. Ethereum 2.0 promises up to [100,000 transactions per second](https://decrypt.co/34204/ethereum-2-0-will-walk-and-roll-for-two-years-before-it-can-run).

**Proof of STAKE**

PoS is a consensus mechanism that relies on validators and staked ETH for the continuation of blocks on the blockchain, and is necessary for sharding. Validators are people who elect to continue the blockchain by Hey depositing (or “staking”) 32 ETH into the deposit contract. On a continuous basis, validators are randomly selected from the pool of all validators to be given the opportunity to create the next block. Should a validator successfully validate a block, they will receive an ETH reward. If a validator attempts to compromise the truthful continuation of the blockchain, their deposit will be ‘slashed’ – meaning they will lose some or all of their 32 staked ETH.

**SHARDING:**

sharding will reduce network congestion and increase transactions per second by creating new chains, known as “shards”.

### **Everyone can run a node**

### Sharding is a good way to scale if you want to keep things decentralized as the alternative is to scale by increasing the size of the existing database. This would make Ethereum less accessible for network validators because they'd need powerful and expensive computers. **With shard chains, validators only need to store/run data for the shard they're validating, not the entire network** (like what happens today). This speeds things up and drastically reduces hardware requirements.

**DAY 7**

* TapRoot.
* Tokens
* ERC 20 in Ethereum
* Why we need ERC20 Tokens?
* Solidity is turing complete or not
* Gas price and gas fees in ethereum how its generated
* Complete the bitcoin transfer process

**TAPROOT:**

The enhancement, called Taproot, is the most significant change to the bitcoin protocol since the SegWit (Segregated Witness) block capacity change

The Taproot upgrade consists of three separate upgrade proposals. However, at its core, the upgrade introduces a **new digital signature scheme** called "**Schnorr"** that will help bitcoin transactions become **more efficient and more private**.

Schnorr - The Schnorr signatures can make more complex transactions on the bitcoin protocol, **such as those from wallets that require multiple signatures,** look like just any other transaction. This makes transactions more private and more secure.

The Taproot upgrade **batches multiple signatures and transactions together.**

Before Taproot's implementation, verification of transactions on Bitcoin's network was slow because each digital signature was validated against a public key. This method multiplied the amount of time required for complex multisig transactions that require multiple inputs and signatures.

The biggest impact would be the bitcoin network's ability **to process more smart contracts, similar to what Ethereum does.**

**(Smart contracts are self-executing transactions whose results depend on pre-programmed inputs.)**

**Crypto Token**:

A crypto token is a virtual currency token or a denomination of a cryptocurrency. It represents a **tradable asset or utility that resides on its own blockchain** and allows the holder to use it for investment or economic purposes.

Coins need to be exchanged with each other through cryptocurrency exchanges because they are built on different, non-standardized code protocols. Conversely, tokens on Ethereum (e.g., ERC-20) can be exchanged through internal applications amongst each other with minimal friction because they are built on standardized code protocols.

**ERC20 (Ethereum request for comment)**

ERC20" refers to **a scripting standard** used within the [Ethereum](https://www.investopedia.com/terms/e/ethereum.asp) blockchain. This technical standard dictates a number of rules and actions that an Ethereum token or smart contract must follow and steps to be able to implement it.

**It is perhaps easiest to think of ERC20 as a set of basic guidelines and functions that any new token created and for creating and issuing smart contracts on the Ethereum blockchain in the Ethereum network must follow.**

* Plenty of well-known digital currencies use the ERC-20 standard, including Maker (MKR), Basic Attention Token (BAT), Augur (REP), and OMG Network (OMG).
* Many of them have been used to create non-fungible tokens (NFTs) for the purpose of an [initial coin offering](https://www.investopedia.com/terms/i/initial-coin-offering-ico.asp) (ICO).

**ICO -** A company looking to raise money to create a new coin, app, or service launches an ICO as a way to raise funds.

Interested investors can buy into the offering and receive a new cryptocurrency token issued by the company. This token may have some utility in using the product or service the company is offering, or it may just represent a stake in the company or project.

*Every token on the Ethereum platform is an ERC-20 token.*

ERC20 contains several functions that a **compliant token** must be able to implement.

* *TotalSupply:* provides information about the total token supply
* *BalanceOf:* provides account balance of the owner's account
* *Transfer:* executes transfers of a specified number of tokens to a specified address
* *TransferFrom:* executes transfers of a specified number of tokens from a specified address
* *Approve:* allow a spender to withdraw a set number of tokens from a specified account
* *Allowance:* returns a set number of tokens from a spender to the owner

<https://cointelegraph.com/explained/erc-20-tokens-explained>

QUESTION?

It essentially means you can write programs (contracts) that can (for the most part) solve any reasonable computational problem.

Using Solidity and Serpent **you have the ability to perform looping and branching statements as well as local state storage.** This functionality is important to have in order to implement most non-trivial computer programs.

**Because the Ethereum VM is turing complete, I can technically implement logic in say Python and translate to solidity if I want.**

**Turing completeness is important for Ethereum smart contracts because you have the ability to implement sophisticated logic.**

This is why some folks say that Solidity is not the right choice for the smart contract. We need something more secure, auditable. This is where Vyper comes in. A basic example of this would be to store health data on Ethereum blockchain. Solidity may not be the best language for that. Vyper surely is.

**GAS:**

[**https://ethgas.io/**](https://ethgas.io/)

*Gas is the unit of measurement used to measure cost of running an operation on the Ethereum blockchain.*

*The gas is used to allocate resources of the Ethereum virtual machine (EVM) so that DApps such as* [*smart contracts*](https://www.investopedia.com/terms/s/smart-contracts.asp) *can self-execute.*

* **Gas prices** are denoted in small fractions of ether called **gwei*.***

To pay Miners, tiny denominations of ETH–nicknamed Gwei–are attached to each Unit. This is the Gas Price.

Gas fees are payments made by users to compensate for the computing energy required to process and validate transactions on the Ethereum [blockchain](https://www.investopedia.com/terms/b/blockchain.asp).

"Gas limit" refers to the maximum amount of gas (or energy) that you're willing to spend on a particular transaction. A higher gas limit means that you must do more work to execute a transaction using ETH or a smart contract.

**The Gas Limit** is your guess at the total amount of work you’re requesting.

How is it determined?

The exact price of the gas is determined **by supply and demand between the network's miners**, who can decline to process a transaction if the gas price does not meet their threshold, and users of the network who seek processing power.

**Day 8:**

* Use case of tokens.
* NFT vs FT
* OpenZeplin - module for creating token
* Token creation ERC20

setting symbol, name, decimals, and total supply value for our token.

* What is solidity, solidity basics
* What happens when the gas limit exceeds?

the transaction will fail and you would have effectively have burnt the Ether / given the miner the Ether for nothing. The contract will run until there is a gas (in Ethereum) to make it happen. When there is no gas, then the contract stops. That’s how solidity supports looks yet avoids indefinite loops.

**PRACTICAL TASK**

* MultiSig wallet + transaction
* **ERC20 token.**

<https://www.quicknode.com/guides/solidity/how-to-create-and-deploy-an-erc20-token>

**Using OpenZeppelin:**

//SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20.sol";

contract MyToken is ERC20{

constructor(uint256 initialSupply) ERC20 ("NAV Coin", "NAV"){

\_mint(msg.sender, initialSupply);

}

function decimals() public view virtual override returns(uint8){

return 8;

}

}

**Tokenization:**

Tokenization can help you transform any real-world asset into a digital asset, thereby enabling easier exchange and transfer of assets. Therefore, tokenization basically implies transferring real-world assets to blockchain or transforming the items of value into tokens that reflect the values.

* Tokenization of Real Estate Assets.
* Tokenization of Artwork.
* Tokenization of Physical Commodities.
* Tokenization in Cargo.

**NFT**

Non Fungible - not exchanged with anything else - Non distinguishable

**It is**

UNIQUE

Limited Number - Only 1 copy mostly

Indivisible

Ownership

Easily transferable

Secure

ERC-721 is used to create tokens

(eg: Cryptokitties)

Use case of fungible tokens

**Smart Contracts:**

A smart contract is a self-executing contract with the terms of the agreement between buyer and seller being directly written into lines of code.

**Solidity:**

pragma solidity >=0.4.0 <0.6.0;

contract SimpleStorage {

uint storedData;

function set(uint x) public {

storedData = x;

}

function get() public view returns (uint) {

return storedData;

}

}

**OpenZeppelin:**

Smart COntract library - OPen Source

Contains collection of implementations

You can find the implementation for major token standards like erc20, erc721,etc.

---------------------------------------------------------------------------

DAY 9:

PRACTICAL:

multisig

hardhat

Ethereum transaction

TOPICS:

* ~~unspent transaction in Eth~~
* ~~ethers.js~~
* ~~eip1559 standard~~
* web3
* ~~Overview of Ethereum transaction - etherscan - transaction details~~

**UNspent transaction:**

The Account/Balance Model, on the other hand, keeps track of the balance of each account as a global state. The balance of an account is checked to make sure it is larger than or equal to the spending transaction amount.

1. Alice gains 5 ethers through mining. It is recorded in the system that Alice has 5 ethers.

2. Alice wants to give Bob 1 ether, so the system will first deduct 1 ether from Alice’s account, so Alice now has 4 ethers.

3. The system then increases Bob’s account by 1 ether. The system knows that Bob has 2 ethers to begin with, therefore Bob’s balance is increased to 3 ethers.

In addition to simplicity, the Account/Balance Model is more efficient, as each transaction **only needs to validate that the sending account has enough balance to pay for the transaction.**

One drawback for the Account/Balance Model is the **exposure to double spending** attacks. **An incrementing nonce** can be implemented to counteract this type of attack. **In Ethereum, every account has a public viewable nonce and every time a transaction is made, the nonce is increased by one.** This can prevent the same transaction being submitted more than once. (Note, this nonce is different from the Ethereum proof of work nonce, which is a random value.)

**EIP1559**

**It allows users more control over fees spent on these transactions.**

EIP-1559 is a core proposal to improve the efficiency of the transactions through the Ethereum network. Once implemented, the proposal will modify the ETH transaction process to a **two-fee mechanism**. The first is a **base fee** that will be auto-adjusted by the ETH protocol according to the network congestion and burned upon completion of the transaction. **A priority fee amount**, used to compensate miners.

The automatic adjustment of the base fee by the protocol enables the Wallets to auto-set the Gas fees for users.

**Ether.js**

Ethers.js is a JavaScript library allowing developers to easily interact with the Ethereum blockchain and its ecosystem.

**Cheatsheet:**

<https://dev.to/hideckies/ethers-js-cheat-sheet-1h5j>

ListAccounts

getBalance

COnnect with metamask

**Block Mined - Details:**

Block height

Timestamp

Transactions

Mined by

Block Reward

Uncles Reward

Difficulty

Total difficulty

Size

Gas used

Gas limit

Nonce

**Transaction Details**

Status

Block Number

Timestamp (date and time the transaction is mined)

Transaction action: Could be token transferred to avail some DApp

From

To

Value

Transaction fee

GAs price

**Web 3**

.

THe next evolution of the internet which is a decentralized web.

Gives ownerships to the users using blockchain technology.

Odyssey - Youtube like.

NO censorship

P2p - not centralised server

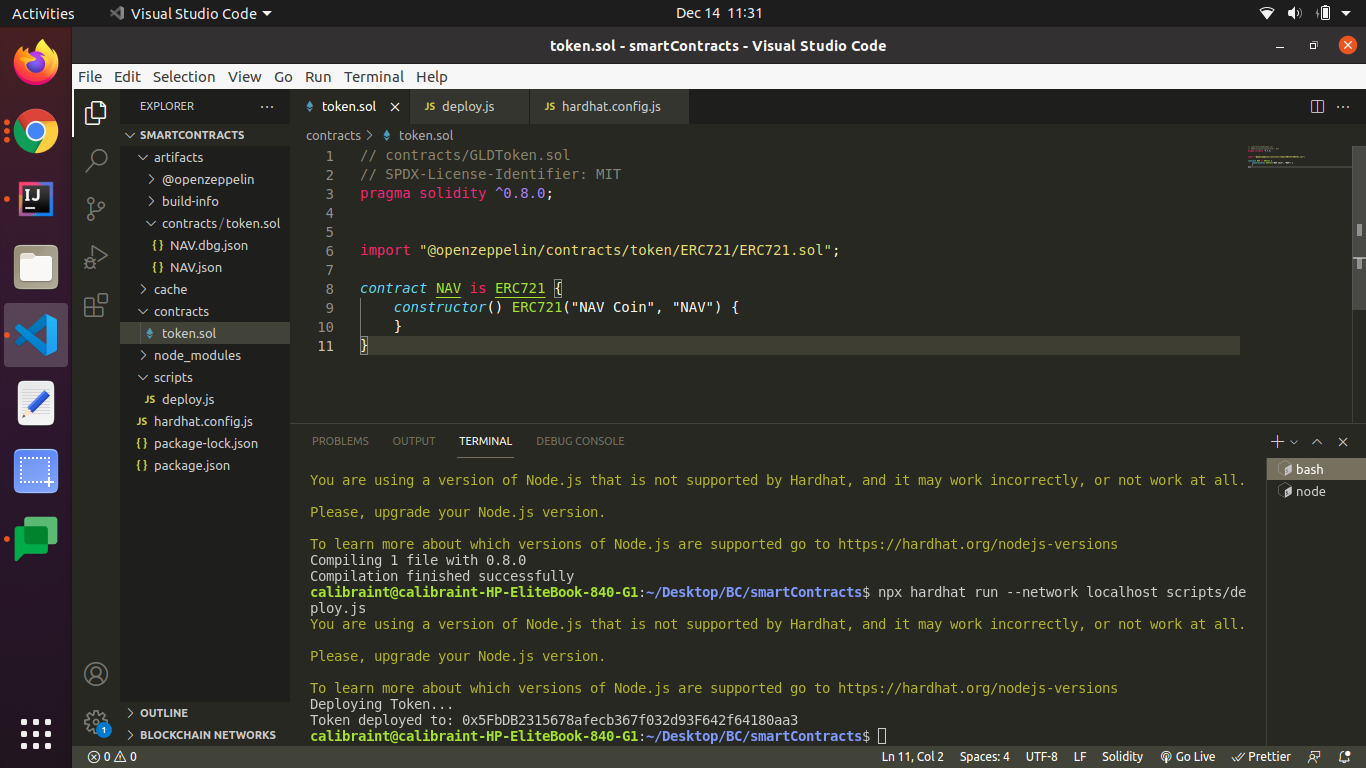
Can maintain anonymity

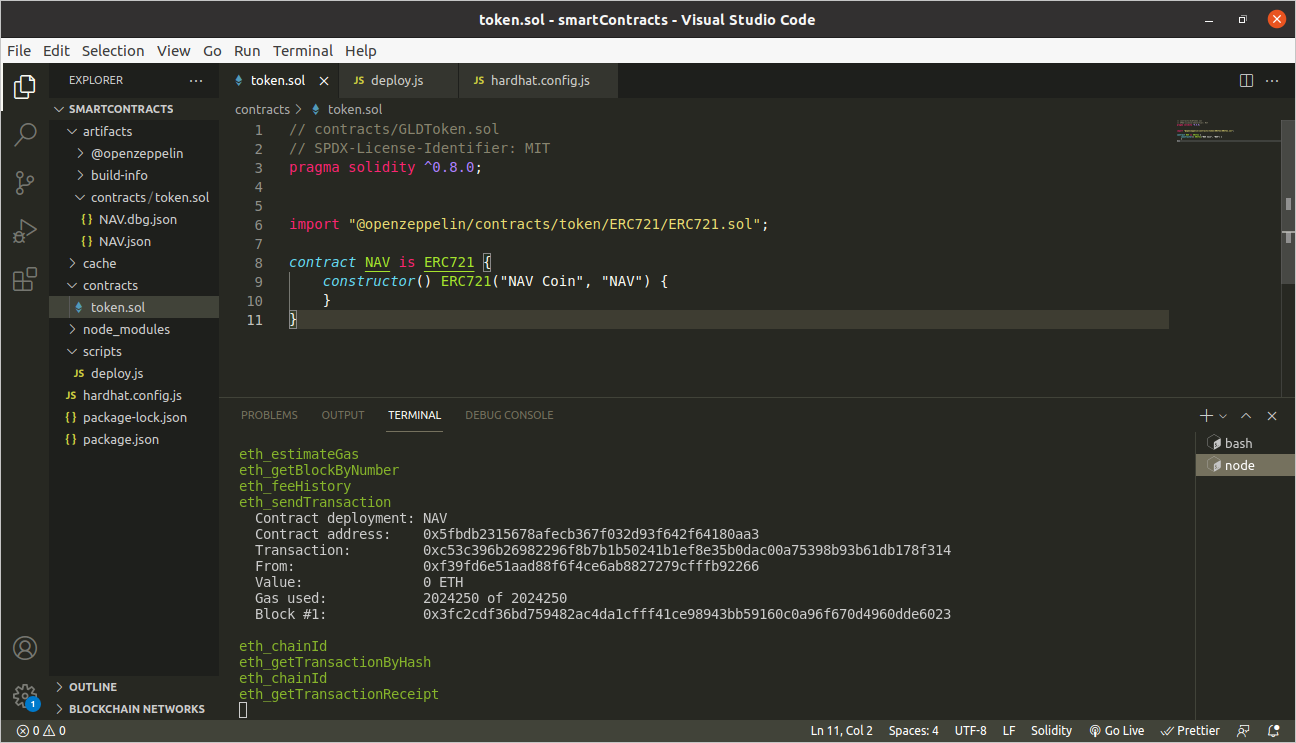
**Web3.js** can be used both in frontends and backends to read data from the blockchain or make transactions and even deploy smart contracts.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Creating using OpenZeppelin and Deployed using Hardhat**

[**https://docs.openzeppelin.com/learn/developing-smart-contracts**](https://docs.openzeppelin.com/learn/developing-smart-contracts)





DAY 10:

* DEX
* Gas price, gas fees, gas limit
* DeFI
* Layer 2 solution
* Layer 1 Vs Layer2
* Uniswap
* ABI - created while deploying smart contract.

Ethereum transaction:

<https://rinkeby.etherscan.io/tx/0xf85b6cf25ffbac787f28886e697a8772a456f11d2e5440ab9620880459a90874>

Deploy token using hardhat & Ropsten:

**Decentralized Exchange:**

A decentralized exchange (or DEX) is a peer-to-peer marketplace where transactions occur directly **between crypto traders.**

DEXs don’t allow for exchanges between fiat and crypto — instead, they exclusively **trade cryptocurrency tokens for other cryptocurrency token**

**Decentralized exchanges rely on smart contracts to allow traders to execute orders without an intermediary.**

[**https://cointelegraph.com/defi-101/what-are-decentralized-exchanges-and-how-do-dexs-work**](https://cointelegraph.com/defi-101/what-are-decentralized-exchanges-and-how-do-dexs-work)

**There are two main types of decentralized exchanges:**

**DeFi:**

**An ecosystem that is decentralized and permissionless**

DeFi is a collective term for financial products and services that are accessible to anyone who can use Ethereum – anyone with an internet connection. With DeFi, the markets are always open and there are **no centralized authorities who can block payments or deny you access to anything**. Services that were previously slow and at risk of human error are automatic and safer now that they're handled by code that anyone can inspect and scrutinize.

| **DeFi** | **Traditional finance** |
| --- | --- |
| **You hold your money.** | **Your money is held by companies.** |
| **You control where your money goes and how it's spent.** | **You have to trust companies not to mismanage your money, like lend to risky borrowers.** |
| **Transfers of funds happen in minutes.** | **Payments can take days due to manual processes.** |
| **Transaction activity is pseudonymous.** | **Financial activity is tightly coupled with your identity.** |
| **DeFi is open to anyone.** | **You must apply to use financial services.** |
| **The markets are always open.** | **Markets close because employees need breaks.** |
| **It's built on transparency – anyone can look at a product's data and inspect how the system works.** | **Financial institutions are closed books: you can't ask to see their loan history, a record of their managed assets, and so on.** |

* **Open: You don’t need to apply for anything or “open” an account. You just get access by creating a wallet.**
* **Pseudonymous: You don’t need to provide your name, email address, or any personal information.**
* **Transparent: Everyone involved can see the full set of transactions (private corporations rarely grant that kind of transparency)**

**DeFi Projects:**

**Lending and Borrowing,**

Maker Dao - Lock a collateral (eth ) and generate stable coins like Dai (follows / pegged to the price of us dollar)

Compound:

**DeX:**

Liquidity based and Order Book

**Derivatives:**

Contracts that derive their value from the performance of an underlying asset.

**Margin Trading:**

Borrowed funds to increase position in a certain assets.

**INsurance :**

Protects against smart contract failure and deposits.

## **Layer 1 Solutions:**

**Consensus protocol improvements:** POw to Pos

**Sharding** - breaking the state of the entire blockchain network into distinct datasets called "shards" a more manageable task than requiring all nodes to maintain the entire network. T

## **Layer-2 Scaling Solutions**

Layer-2 refers to a network or technology that operates on top of an underlying blockchain protocol to improve its scalability and efficiency.

Lightning Network is a Layer-2 solution built to improve transaction speeds in this fashion on the Bitcoin network.

**Nested blockchains:**

Block chain on top of another.

Parent delegates work o child chain and then returns.

Basically distribution of work

**The distribution of work** under this model reduces the processing burden on the main chain to exponentially **improve scalability.**

**State channels:**

A state channel facilitates **two-way communication** between a blockchain and [off-chain](https://www.gemini.com/cryptopedia/glossary#off-chain) transactional channels and improves overall transaction capacity and speed.A state channel does not require validation by nodes of the Layer-1 network. Instead, it is a network-adjacent resource that is sealed off by using a multi-signature or smart contract mechanism.

’Eg: Lightning network

**Sidechains:**

used for large batch transactions.

Main chain validates transaction maintained…takes care of security.

Sidechain just publicly record the transaction to the ledger.

**DAY 11:**

* Order book
* Automated market match
* Abi
* Uniswap
* Use case of defi
* Uni token
* Practical task: Create ERC20 Token and stop the minting supply after 10000

**OrderBook:**

An order book is an electronic documentation of an asset’s buy and sell activity on a trading platform

An order book comprises different key information regarding an asset. First, it has dedicated sections **for buyers and sellers.** Next, there is a **bid and ask section**. Here, "**asks" represent sell requests** while "**bids" indicate buy orders.**

**AMM:**

An automated market maker (AMM) is the underlying protocol that powers all decentralized exchanges (DEXs).

**UNISWAP:**

Uniswap is a leading decentralized crypto exchange (DEX) that runs on the Ethereum blockchain.

Allows trading of erc20 tokens

DEX uses liquidity pools to fulfill buy and sell orders rather than orderbook.

People who deposit funds in the pools are called Liquidity providers.

As a part of the deposit, the LP will get a part of the DEX’s trading fees in a process called liquidity mining.

**How is price determined?**

NO orderbook.

They dont trade with one another, instead using the liquidity pool.

INstead of last traded price, a math formula is used to compute the price called AMM (automated market maker)

Uniswap uses Constant Product MArket maker model tp determine the price.(x\*y=k)



V1 - Trading of eth to erc20

V2 - trading of erc20 to erc20

V3 - more capital to who deposit to pool

**How to use?**

Have a metamask

Connect uniswap to metamask

But **Price Slippage could occur -** Price when you place an order will differ from the the order is executed. You can cap it though, based on how much your willing to afford for the slippage.

**ABI:**

**Application Binary Interface.**

**ABI is an interface to interact with EVM bytecode**. ABI defines the methods and structures used to interact with the binary contract.

The ABI indicates **the caller of the function to encode the needed information like function signatures and variable declarations** in a format that the EVM can understand to call that function in bytecode; this is called ABI encoding

**Why is it used?**

In Ethereum, the ABI is used to **encode contract calls for the EVM and to read data out of transactions**. The purpose of an ABI is to **define the functions in the contract that can be invoked and describe how each function will accept arguments and return its result.**

ABI encoding is mostly automated, taken care of by compilers like REMIX or wallets interacting with the blockchain.

[

{

"inputs": [],

"payable": false,

"stateMutability": "nonpayable",

"type": "constructor"

},

{

"constant": true,

"inputs": [],

"name": "getResult",

"outputs": [

{

"internalType": "uint256",

"name": "",

"type": "uint256"

}

],

"payable": false,

"stateMutability": "view",

"type": "function"

}

]

**UNITOKEN:**

Sept 2020 airdropped.

Initially, It is a governance token, for uniswap users to take development decisions .

**Use Cases of Defi**

**Peer to peer borrowing and lending**

Take loans

the borrower need not find a lender, instead of the smart contract itself acts as lender, and interest is calculated based on supply and demand.

## **Asset Management**

With DeFi, you are the only custodian of your own crypto assets. You can securely interact with decentralized applications for buying, selling, and transferring your cryptos by using crypto wallets such as MetaMask.

**GAmes**

Most of the new video games have in-app purchases and loot box features in them. These features enable users to use real-life currency to buy new skins for their characters and tools.

**Prediction**

**Augur** is one of the leaders in the DeFi ecosystem that specialize in the **prediction market.** This platform allows the users to place bets on events like — sports, world events, economics, election results, and more.

**Tokenization**

* Arbitrage
* Leverage
* Futures Vs margin Trading
* Uniswap competitors

**Arbitrage:**

Arbitrage is the simultaneous purchase and sale of the same asset in different markets in order to profit from tiny differences in the asset's listed price. It exploits short-lived variations in the price of identical or similar financial instruments in different markets or in different forms.

As a straightforward example of arbitrage, consider the following. The stock of Company X is trading at $20 on the New York Stock Exchange (NYSE) while, at the same moment, it is trading for $20.05 on the London Stock Exchange (LSE).

A trader can buy the stock on the NYSE and immediately sell the same shares on the LSE, earning a profit of 5 cents per share.

**Cryptocurrency arbitrage is a strategy in which investors buy a cryptocurrency on one exchange and then quickly sell it on another exchange for a higher price.**

**Leverage:**

Leverage(Margin trading) - borrowing money to invest in a coin

Leverage Trading in Crypto denotes a tool that allows [investors](https://www.finance-monthly.com/2021/06/how-seriously-should-investors-take-the-bank-of-england-governors-cryptocurrency-warning/) to make spot transactions (purchase and sale) with the help of borrowed capital from brokers

BInance

**Difference:**

Margin trading involves borrowing assets from a lender to trade more than you normally could. ...

Futures involve an agreement to transact an asset on a specific date at a specific price and allows traders to bet on what they think the market will do in the future

Margin trading allows clients to take much smaller positions through leverage. ... The one important difference you need to remember is that when you opt for margin funding, you pay interest on the amount funded. On the contrary, when you opt for futures trading, there is no interest payable by you.

**Competitors:**

**Most of them employ a similar AMM concept**

**1inch:**

Automatically finds the most efficient route for token swaps

Supports limit orders

**Sushiswap:**

**Pancake Swp**

* ~~Wallet Connect~~
* ~~Metamask resource wallet connect~~
* ~~Web3 vs Metamask~~
* Solidity
* Virtual functions
* Require functions

Task: LOCKING

Create a token

lock it

ONly after 10-20 blocks have been added

check the block no

unlock the wallet

spend

**Wallet Connect:**

WalletConnect is an open protocol made to facilitate a secure connection between mobile cryptocurrency wallets and desktop applications, such as dapps. Transactions are made through an encrypted connection by scanning a QR code, and are confirmed on the mobile device. As the private key never leaves the user’s device, their funds are never at risk and the possibility of a hijack is very low.

WalletConnect is an open protocol to communicate securely between Dapps and Wallets. The protocol establishes a remote pairing between two apps and/or devices using a Relay server to relay payloads. These payloads are symmetrically encrypted through a shared key between the two peers. The pairing is initiated by one peer displaying a QR Code or deep link with a standard WalletConnect URI and is established when the counter-party approves this pairing request.

**WEB3 VS Metamask:**

* Metamask is a browser extension which facilitates sending transactions. It utilizes Infura to send the transactions to the blockchain
* web3 is a library to facilicate handling Ethereum transactions. You can send direct transactions with it as long as you have access to a node. It includes functionality to maintain accounts.

[Metamask](https://ylv.io/10-web3-metamask-use-cases-ever-blockchain-developer-needs/Metamask.io) is the de facto standard for dApps on the web. **It injects a Web3 instance** into a window object making it available for JavaScript code.

When a user loads a webpage, MetaMask automatically injects an Ethereum provider and a Web3 instance for the web page to use. This allows dapps to access the blockchain, propose transactions, and read their users’ account addresses.

If Web3 is not present, we try to connect to a localhost provider, like ganache.

[web3.eth.sendTransaction()](https://github.com/ethereum/wiki/wiki/JavaScript-API#web3ethsendtransaction) can be used to send messages and data but here we're just sending ether. It is possible to add a specific amount of Ether to cover the cost of making the transaction (gas price) and by excluding this value we allow MetaMask to make this decision for us.

* ~~PolyGon~~
* ~~DAI~~
* DAO with Unitoken
* ~~Storage, memory, call data, virtual~~
* ~~Optimism and Arbitram~~

**Polygon:**

[**https://www.youtube.com/watch?v=GWUwFDFOipo**](https://www.youtube.com/watch?v=GWUwFDFOipo)

Polygon is a protocol and framework for building and connecting Ethereum-compatible blockchain networks, **offering an ecosystem with lower transaction costs and faster speeds than Ethereum.**

**MATIC ->Polygon network**

**Coin is still called matic coin.**

POlygon is a layer 2 scaling platform to help apps to solve problems of handle low throughput, not user friendly, limited options for developers (All etherium projects run on the same network)

-> Increase the usage of defi tools and apps by connecting blockchains together.

Most of the Dapps that are deployed on Polygon are Defi and Games.

Block time - 2.4s

Transaction pers second: UPto 10,000

Polygon is a sidechain. It has its own consensus sep from etherium.

It is connected to eth in 2 ways:

1. The hashes of the state of polygon is stored in eth chain l1 every 5 minutes to validate transactions.
2. Anyone can challenge the validity if transaction on polygon. Validators stake poly token on eth, if anything goes wrong, thair drake will be slashed.

ETH->Poly (fast)

POly -> ETH (7days)

**DAI:**

* It is a stable coin.
* Backed by crypto assets
* Pegged to us dollars.
* Value is fixed.
* MakerDAO maintains the stability of DAI (value and circulation)
* MakerDao is built on ethereum.

**USES:**

1. You can borrow DAI by locking in collaterals. (in eth, bat) you have to miantain 150% of the total dai u want to borrow.
2. Save
3. Hedge Funds
4. Send money

**OPtimism:**

* Layer 2 scaling on ethereum.
* Does Not have a native token.
* FOr now, They have a permission mainnet deployment so only whitelisted projects can sign up. To make sure when they open the public mainnet is open everything is fine.
* Uses OPtimistic Rollups -

**ARbitram:**

* NO native token.
* FOr now, They have a permission mainnet deployment so only whitelisted projects can sign up.
* Uses OPtimistic Rollup
* Allows sidechains and channels
* Supports EVM, supports all tools.
* You needn't use a different version of solidity like optimism.

**DAO :**

A **decentralized autonomous organization** (DAO) is an entity with no central leadership. Decisions get made from the bottom-up, governed by a community organized around a specific set of rules enforced on a blockchain. ... Decisions are made via proposals the group votes on during a specified period.

The people who own tokens associated with the DAO are responsible for voting on changes and proposing new ideas to keep the DAO up and running and improving.

**Storage, Memory**

## Storage

The storage location is **permanent data, which means that this data can be accessed into all functions within the contract.** To make it more simple, you can think of it as the hard disk data of your computer where all the data gets stored permanently. Similarly, the storage variable is stored in the state of a smart contract and is persistent between function calls. Keep in mind that storage data location is expensive compared to other data locations.

## Memory

The memory location is temporary data and cheaper than the storage location. It can only be accessible within the function. Usually, **Memory data is used to save temporary variables for calculation during function execution.** Once the function gets executed, its contents are discarded. You can think of it as a RAM of each individual function.

## **Calldata**

Calldata is non-modifiable and non-persistent data location where **all the passing values to the function are stored.** Also, **Calldata is the default location of parameters** (not return parameters) of external functions.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Sidechains VS Rollups
* AMM
* EVM based blockchain
* Solidity -