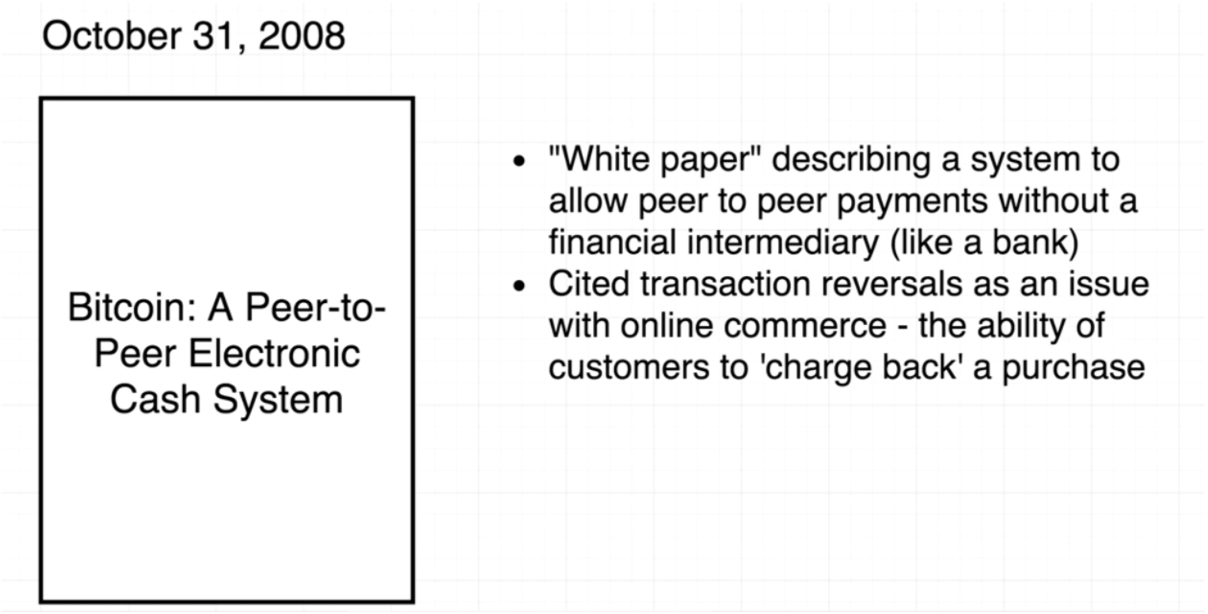
🤔

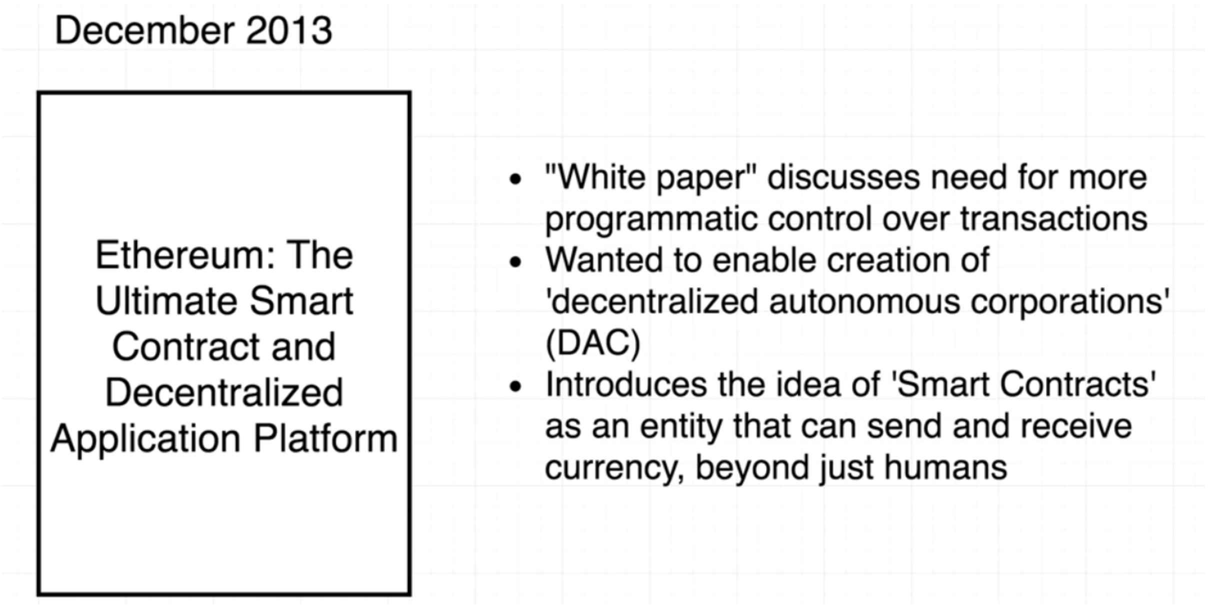
What is Ethereum?



October 31, 2008

Bitcoin: A Peer-to-Peer Electronic Cash System

- "White paper" describing a system to allow peer to peer payments without a financial intermediary (like a bank)
- Cited transaction reversals as an issue with online commerce - the ability of customers to 'charge back' a purchase

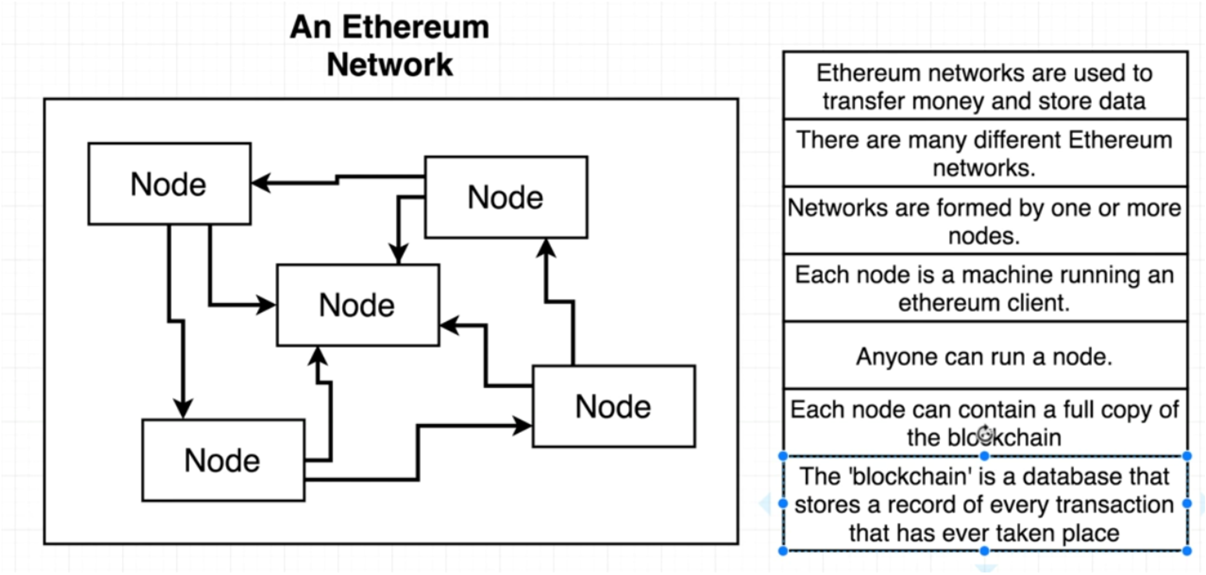


December 2013

Ethereum: The Ultimate Smart Contract and Decentralized Application Platform

- "White paper" discusses need for more programmatic control over transactions
- Wanted to enable creation of 'decentralized autonomous corporations' (DAC)
- Introduces the idea of 'Smart Contracts' as an entity that can send and receive currency, beyond just humans

Ethereum -



Ethereum networks are used to transfer money and store data

There are many different Ethereum networks.

Networks are formed by one or more nodes.

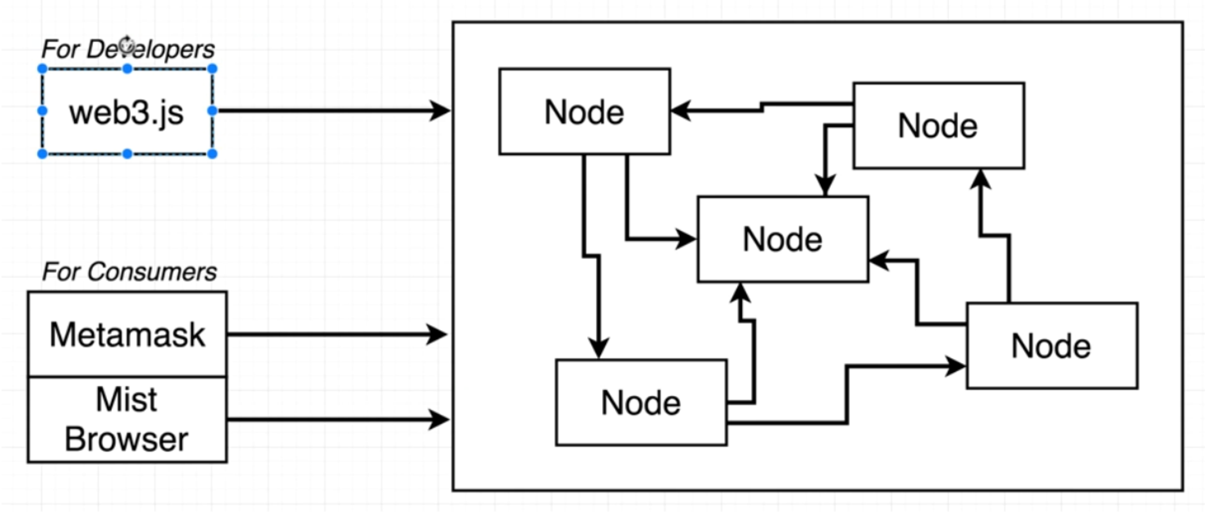
Each node is a machine running an ethereum client.

Anyone can run a node.

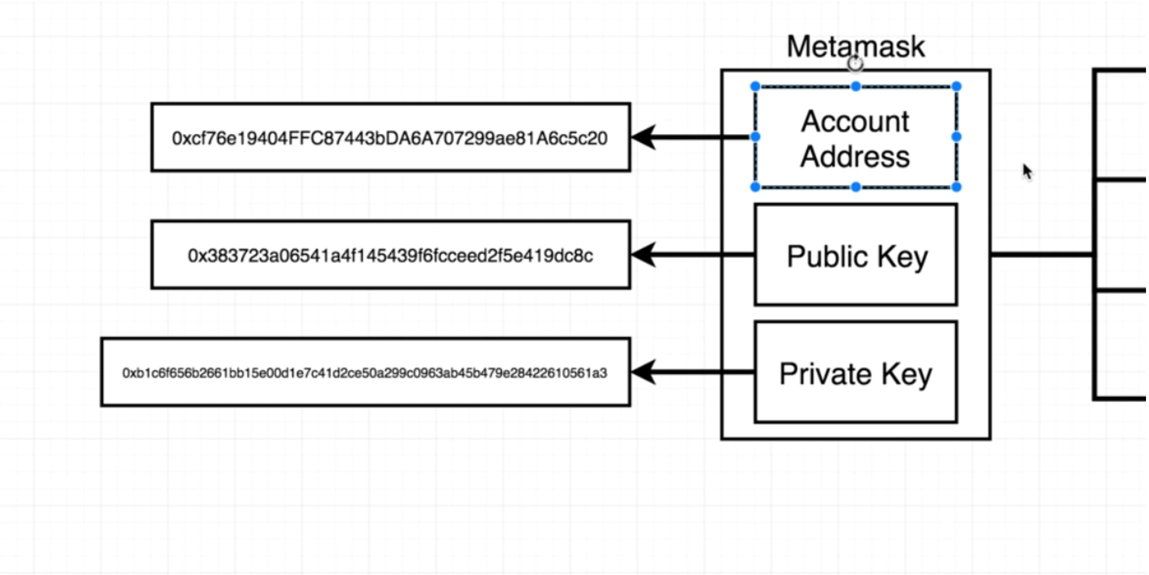
Each node can contain a full copy of the blockchain

The 'blockchain' is a database that stores a record of every transaction that has ever taken place

- Smart Contract -
  - The core of Ethereum is the Smart Contracts.
  - It's a piece of code that lives in the Ethereum blockchain.
  - This smart contract can be made to do certain things by sending a message to it from some other smart contract.
- Connecting to Ethereum Network -
  - Developers use libraries/packages like web3.js to connect to and build on the Ethereum network.
  - Consumers can connect to it using clients like Metamask.

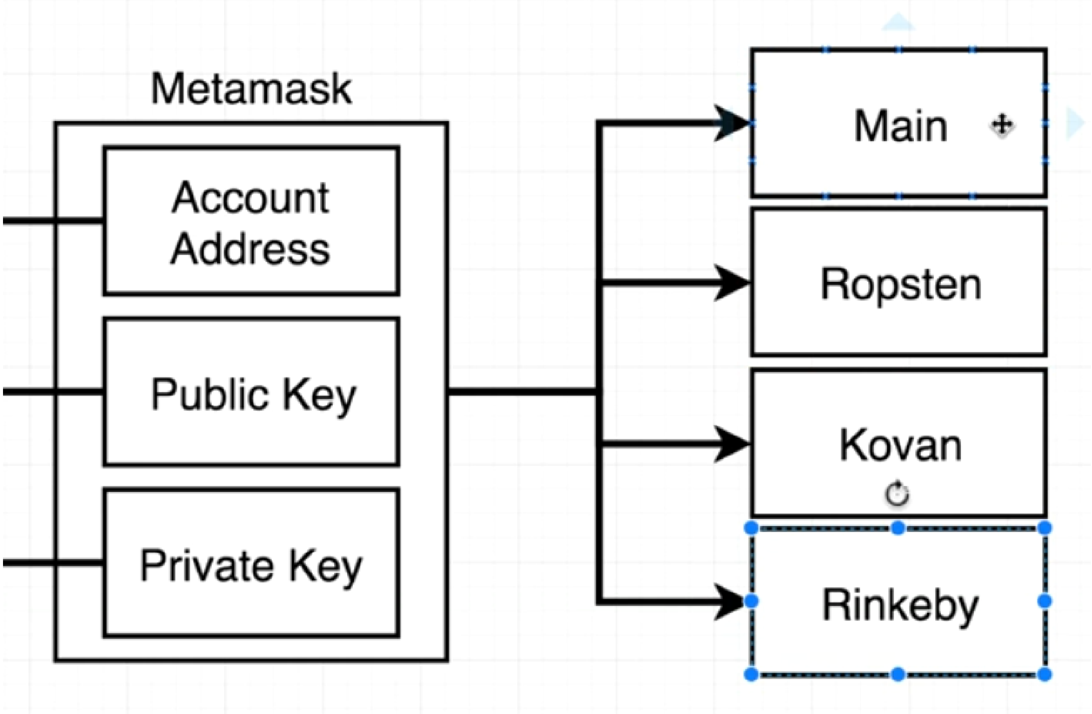


- Account on Ethereum Network -
    - We created an account on the Ethereum network using Metamask.


Metamask
Account Address
Public Key
Private Key
0xcf76e19404FFC87443bDA6A707299ae81A6c5c20
0x383723a06541a4f145439f6fcceed2f5e419dc8c
0xb1c6f656b2661bb15e00d1e7c41d2ce50a299c0963ab45b479e28422610561a3

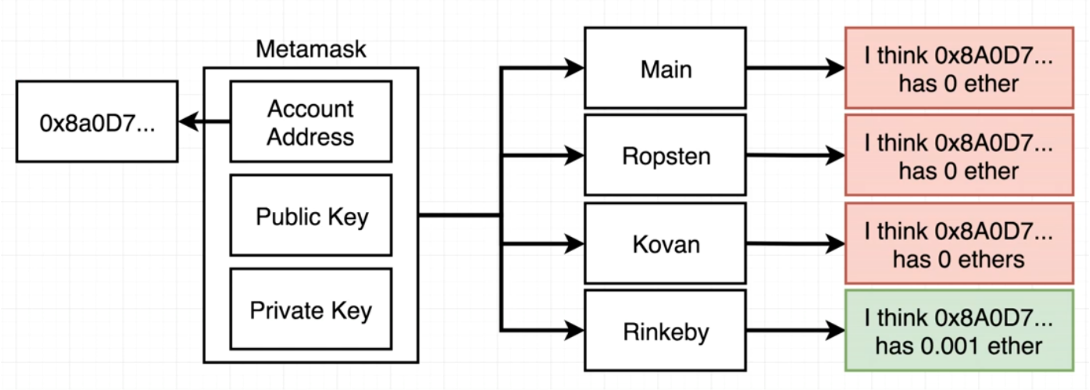
    - Each account constitutes of 3 things -
        - Account Address -
            - Unique Identifier for our acc.
            - Like email.
        - Public Key
        - Private Key
            - A very large number (10^76 ke order ka)
            - Shouldn't be shared with others in order to protect our account. If someone has your private key then they have access to your account.
        - These keys form a password of sorts, used to authorize the sending of funds from your account to some other account.
        - The Account Address, Public Key and Private Key are stored as hex numbers.
        - **IMPORTANT -** Our account address remains the same over all the Eth networks like the Main net, Rinkeby test

net, Kovan test net, etc.

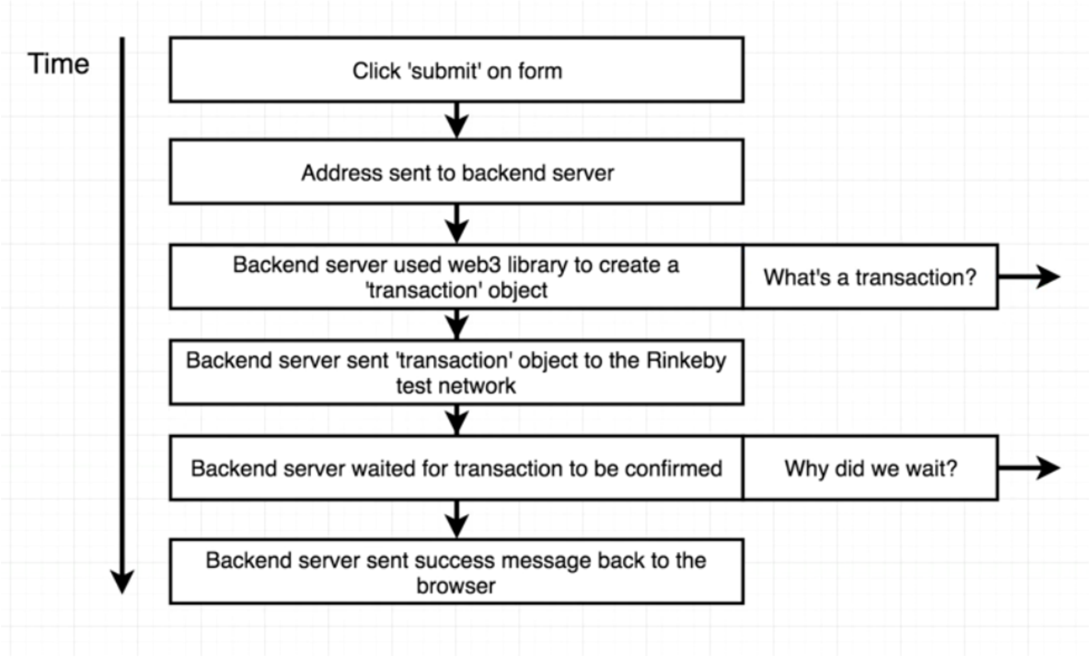


What is a transaction?

- Mere account mai jitna balance he ek network par that is completely independent of the balance I have on some other network.

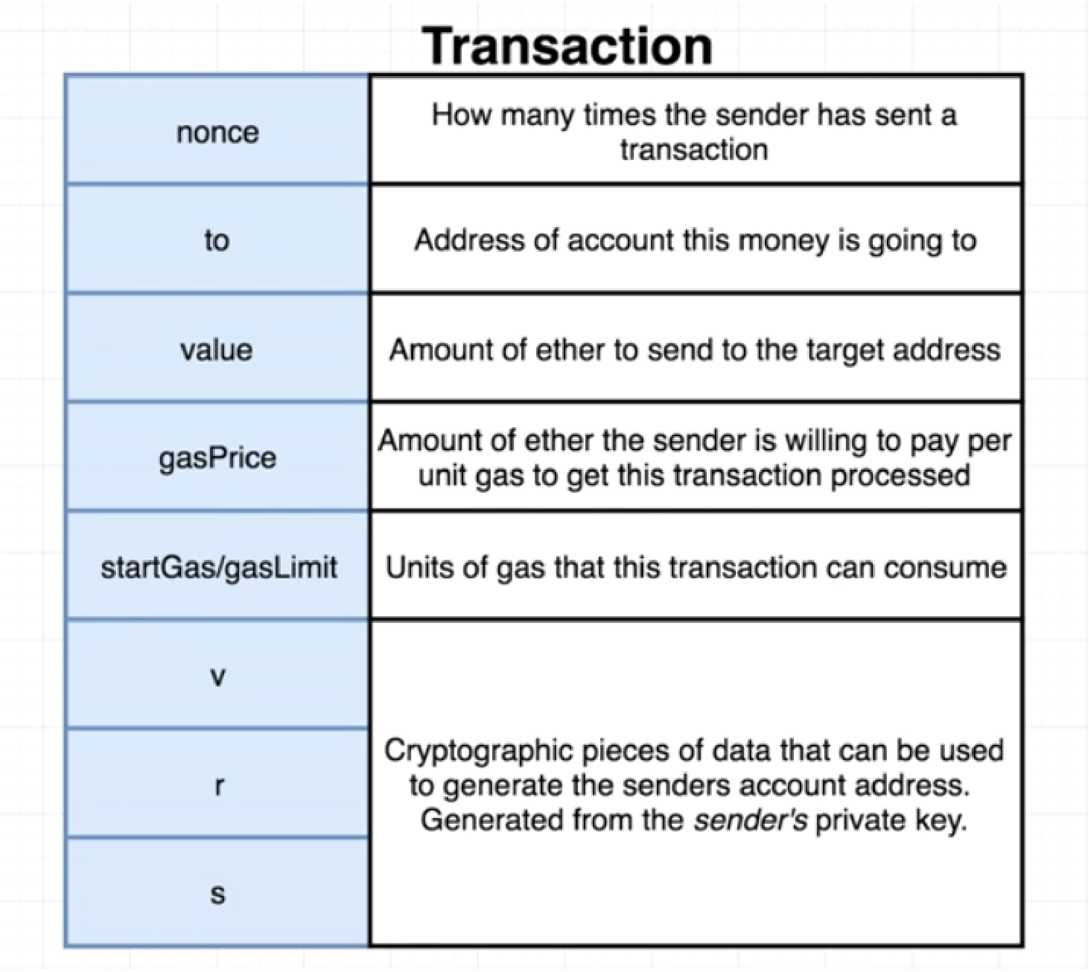

Metamask
0x8a0D7...
Account Address
Public Key
Private Key
Main
Ropsten
Kovan
Rinkeby
I think 0x8A0D7... has 0 ether
I think 0x8A0D7... has 0 ether
I think 0x8A0D7... has 0 ethers
I think 0x8A0D7... has 0.001 ether

- What happened when we request for some rinkeby coins -


Time
Click 'submit' on form
Address sent to backend server
Backend server used web3 library to create a 'transaction' object
What's a transaction?
Backend server sent 'transaction' object to the Rinkeby test network
Backend server waited for transaction to be confirmed
Why did we wait?
Backend server sent success message back to the browser

Transaction

- A transaction is a **record** that describes an **attempt by an account** to **send money** to **another account.**
  - We create a transaction object then send it to the network (Main net, Rinkeby etc) to get processed.



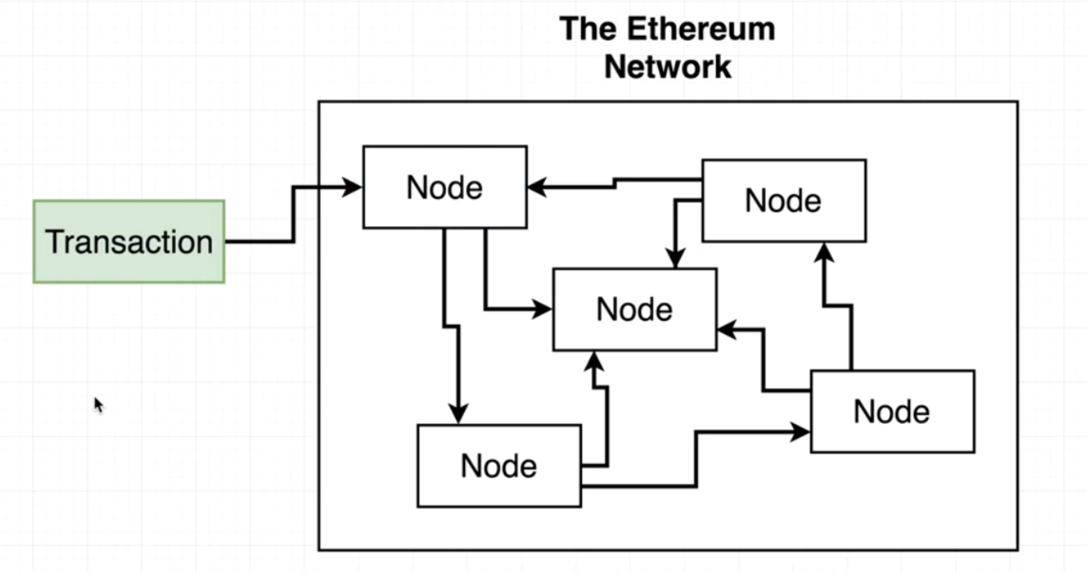
Transaction

nonce	How many times the sender has sent a transaction
to	Address of account this money is going to
value	Amount of ether to send to the target address
gasPrice	Amount of ether the sender is willing to pay per unit gas to get this transaction processed
startGas/gasLimit	Units of gas that this transaction can consume
v	Cryptographic pieces of data that can be used to generate the senders account address. Generated from the *sender's* private key.
r	
s	

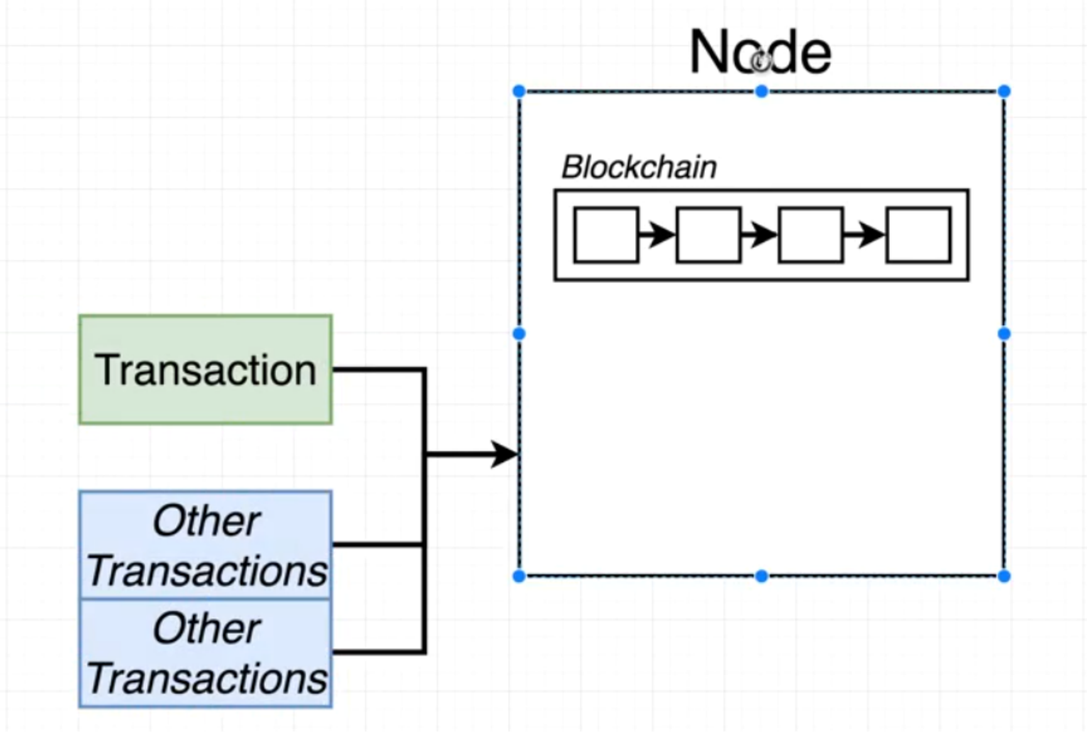
- Sender’s Private Key is used to generate v, r & s but reverse is not possible, this is for security reasons.
- v, r & s can be used to generate the Account Address though.
- v, r & s will be valid if they can be used to generate valid Account Address.

Why did we wait?

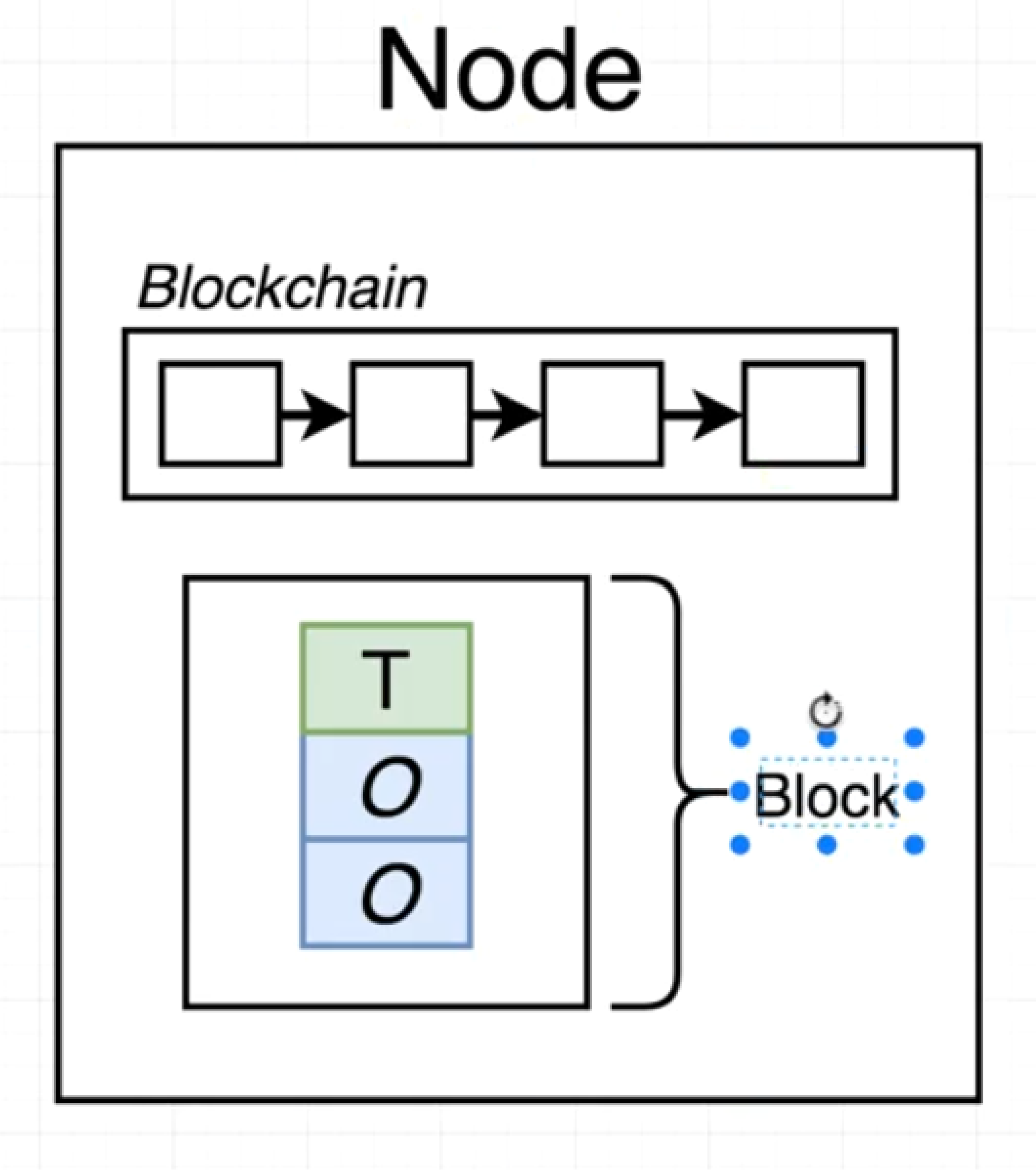
- If you noticed there was some wait time involved in the transaction.
- The transaction object was sent to one specifc node of the network.



- Transaction object comes a Node, highly possible that some other transactions came at the same time as well.
  - Note - All the nodes have a copy of the entire blockchain ie the database containing all the records of the transactions that ever took place.

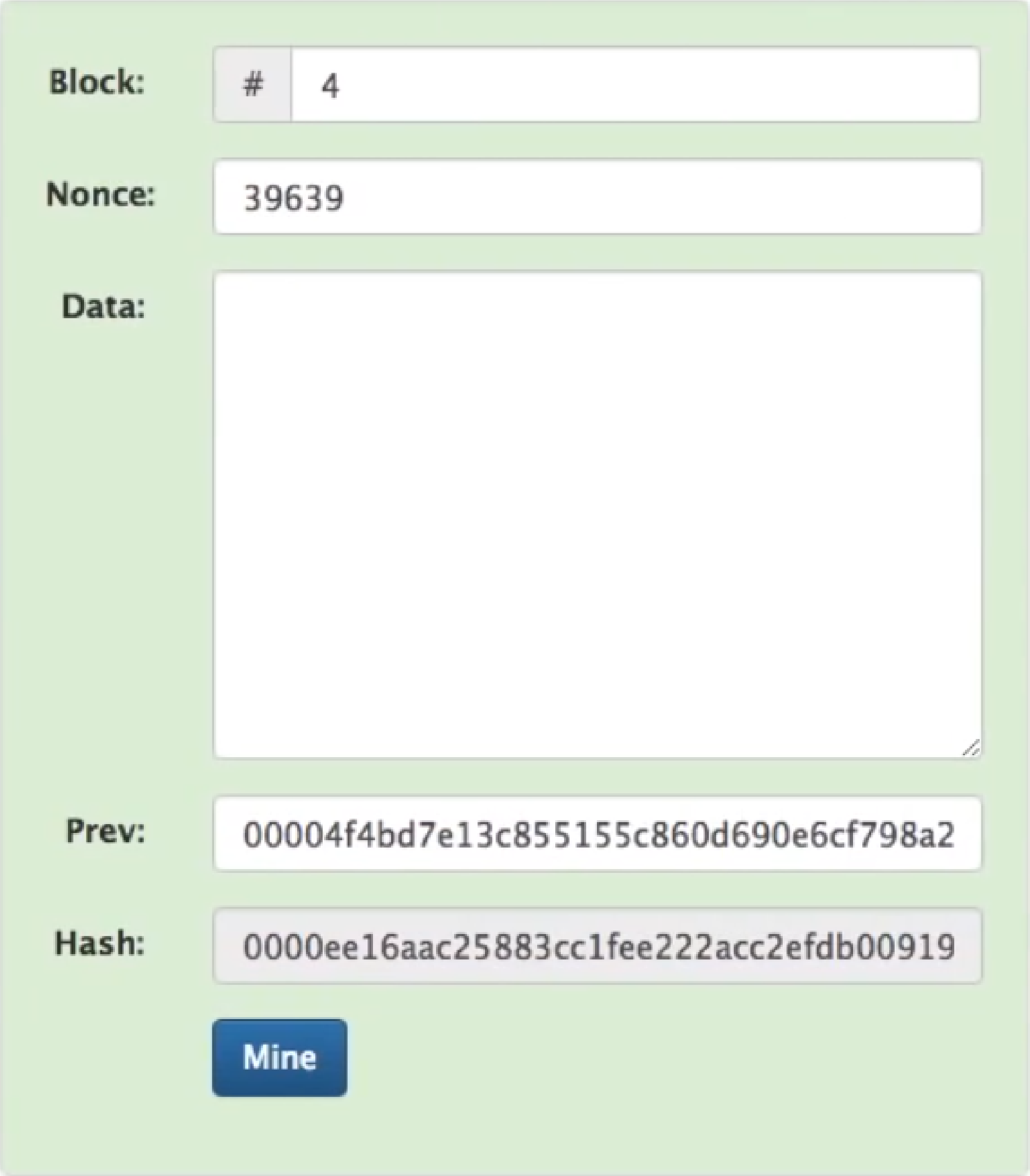


- All the transactions are stored in form of a list called a Block inside of the Node.
- Then the Node runs some validation logic on the block.
    - **This validation logic takes time, hence, the wait we experience.**
    - This process of running the validation logic on the block in a node is called **Mining**.



Basics of Blockchain -

- Section 1 - vid 13.
- Lets say 5 blocks ki blockchain he and maine right-most block ka data change kar diya -
  - Data change karne se uska hash gadbada jaega, hash recalculate hoga, yani us block ko firse mine karna padega.
  - Video wise correct hash is such that starts with 0000.
- Now consider that the 2nd block from left ka data change hua -
  - Aisa karne pe left se 2nd, 3rd, 4th aur 5th blocks sabko firse mine karna padega.
  - har ek block ke pass prev block ka hash tha.
- **This is how a blockchain resists data change.**
- A block of a blockchain -
  - The nonce value will be recalculated upon data change such that the hash starts with 0000. This process is called mining.



Block: # 4

Nonce: 39639

Data:

Prev: 00004f4bd7e13c855155c860d690e6cf798a2

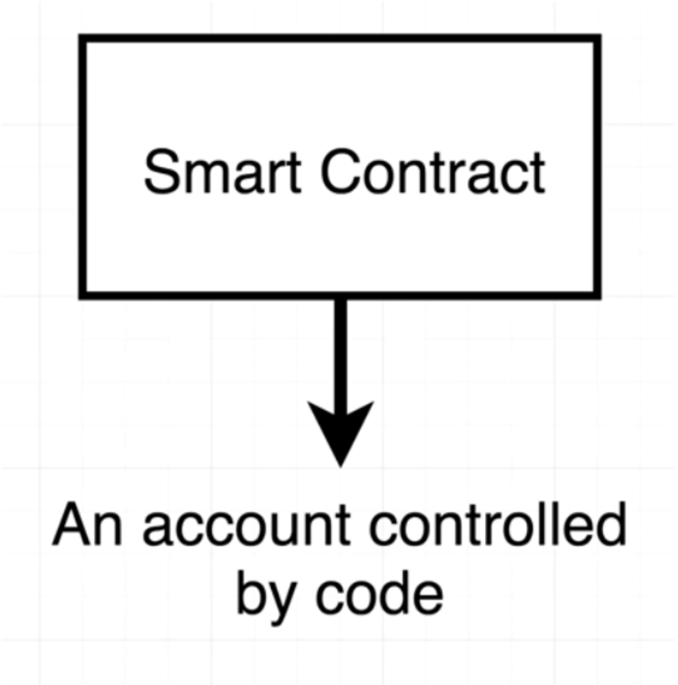
Hash: 0000ee16aac25883cc1fee222acc2efdb00919

- Now, decentralized networks mai sare clients/peers ke pass puri blockchain ki identical copy hoti he.
- To inorder to tell that a particular peers copy is **old/wrong** we can simply check the hash of the last block, if it doesnt start with 0000 or the hash wont match with that of copies on other peers that means there is some modification on this copy.
- So, its really easy to tell if some modification has been done with a copy.

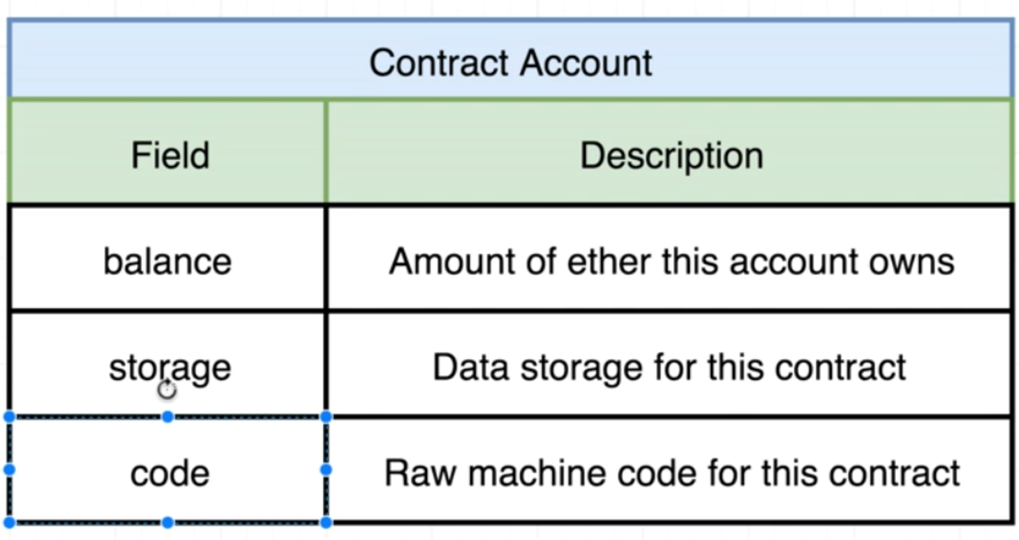
- **IMPORTANT -** The point that **valid/correct** hash has to start with 0000 is not exactly correct.
  - For Ethereum based blockchains the hash (a hex number) should be smaller than a target value.

- **Block Time -**
  - Its the time required by a block to find a valid hash (a hash less than a specific value).
  - Ye ek transaction ke corresponding he.
  - The **target block time** of Ethereum Network is **15s**.
  - If a transaction takes more than 15s of block time then the target value is increased in order to bring the block time down. Similarly, if a transaction takes less than 15s then the target value is decreased. Basically, block time ko 15s pai manage karna chah rahe he.
  - Now ye dynamic change in target value is required because the computation power of the Eth network has is always in flux, ie its not static, keeps on changing.

Smart Contract -

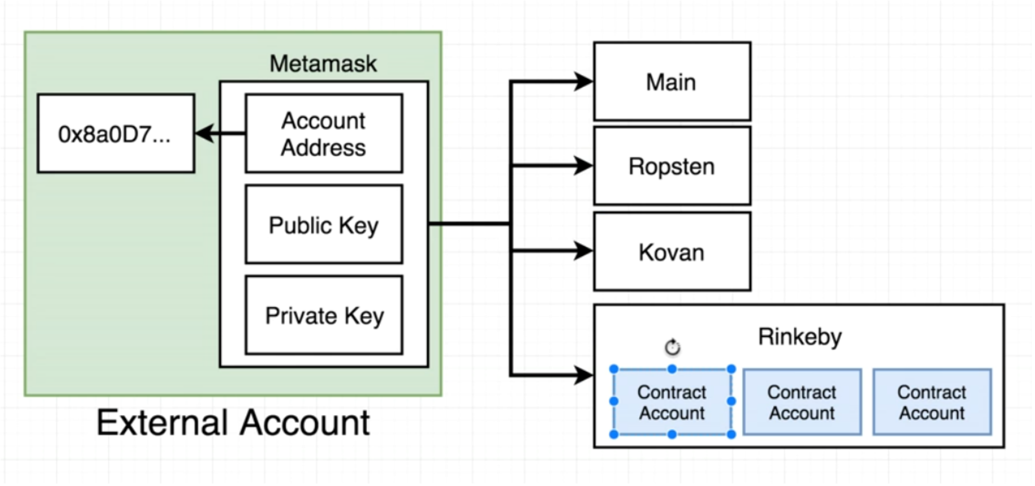


- Properties in a contract account -
  - Storage - Its the data corresponding to the application which uses this smart contract.

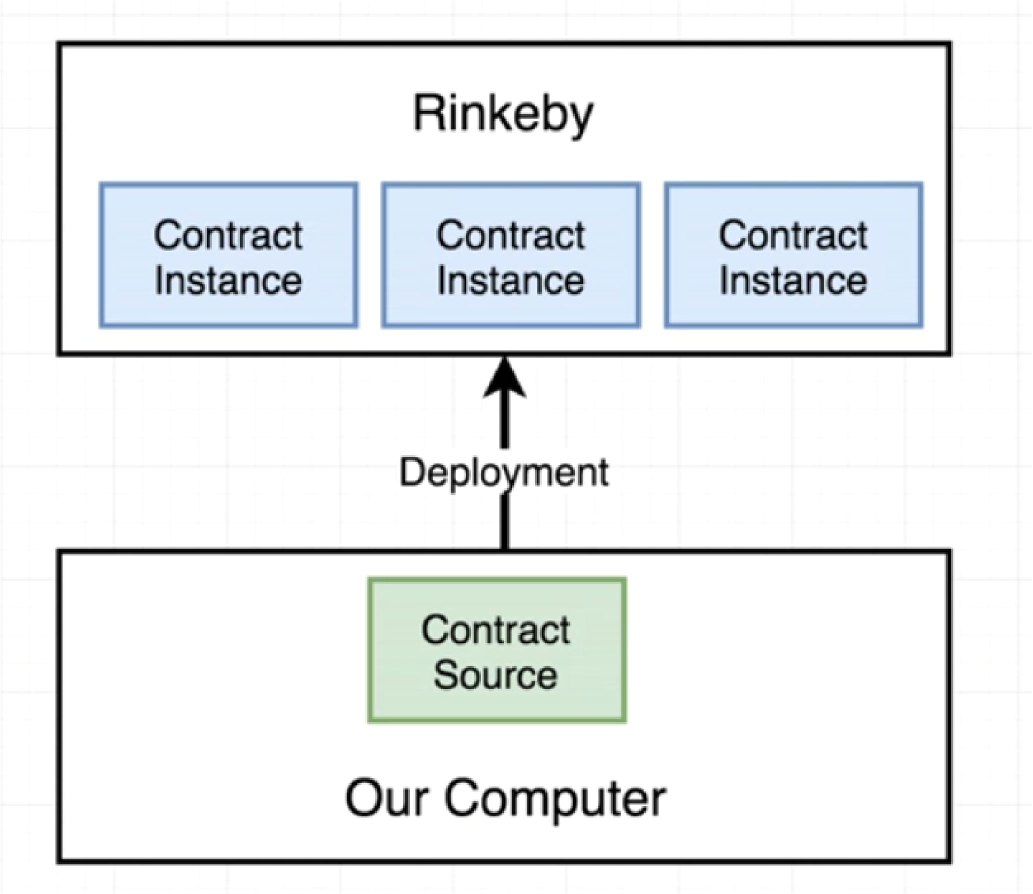


Contract Account	
Field	Description
balance	Amount of ether this account owns
storage	Data storage for this contract
code	Raw machine code for this contract

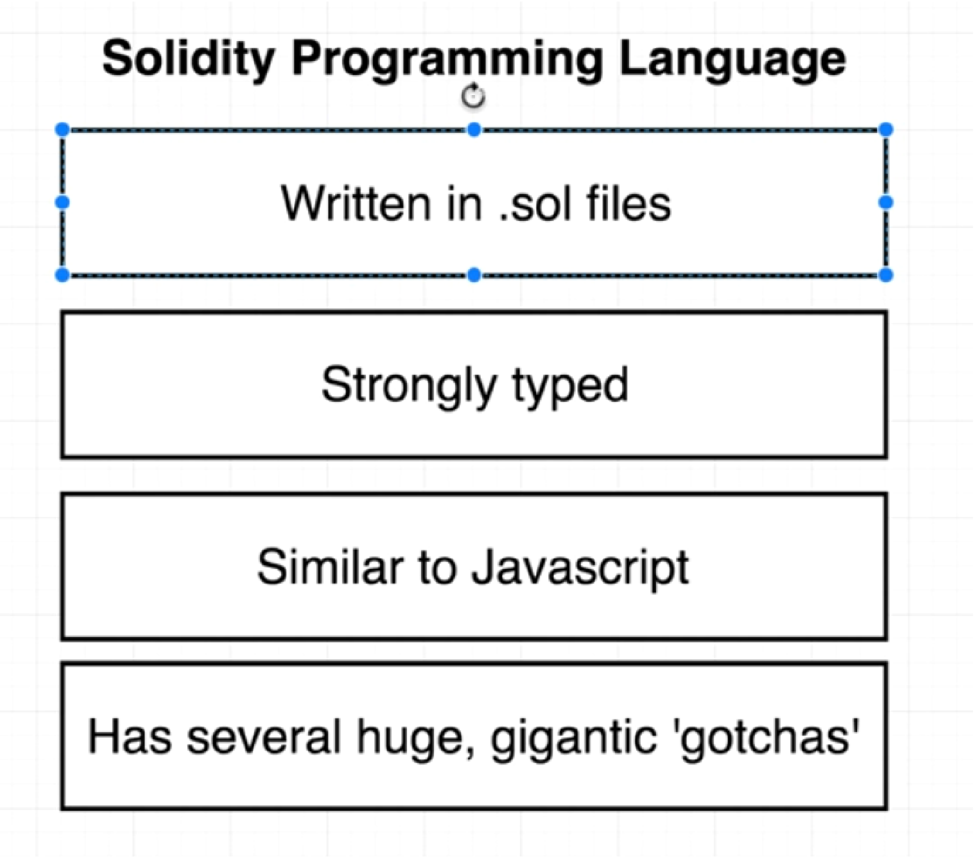
- External Account is not associated to any network. But the Contract Account is linked to only a particular network. To agr kisi aur network pe wo contract account chahiye to phirse deploy karna padega on that network.



- Contract Source is like a class, then I deploy this Contract Source to the network, in this way we create an instance of this Contract Source called a Contract Instance, this is like an object of the class.
    - Multiple instances deploy kar sakta mai ek network pe.



- **We use Solidity to write Smart Contracts.**



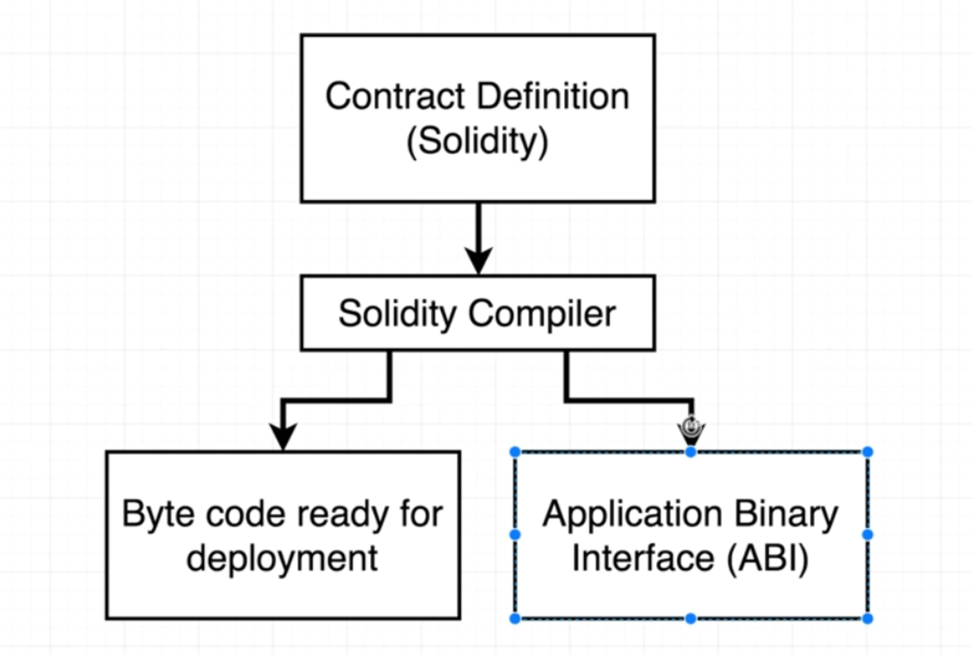
Solidity Programming Language

Written in .sol files

Strongly typed

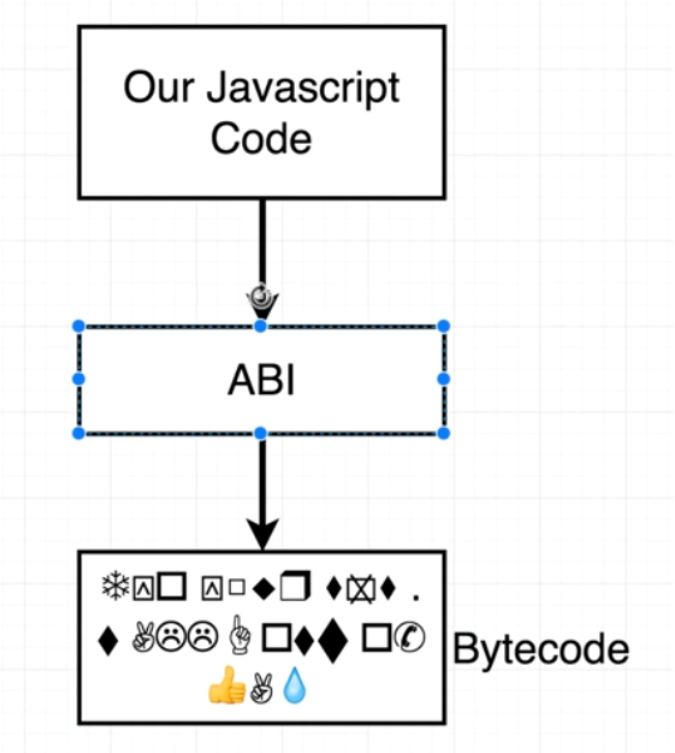
Similar to Javascript

Has several huge, gigantic 'gotchas'


Contract Definition (Solidity)
Solidity Compiler
Byte code ready for deployment
Application Binary Interface (ABI)

- Ethereum network mai direct .sol code execute ni hota.
- Sol code gets compiled to Byte Code and ABI.

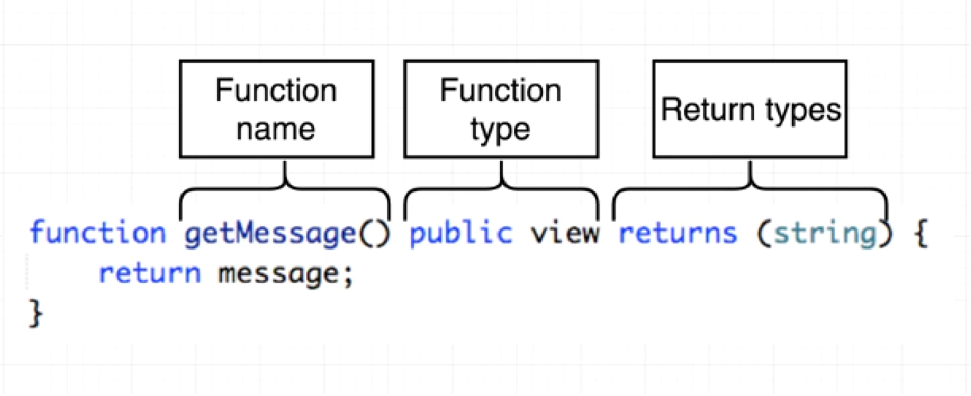
- Ye byte code hota he jo ethereum network pe execute hota he.



- Our JS code which acts as the frontend of our smart contract cant directly interact with the byte code deployed on the network, so, we use the ABI to convert our JS code to byte code.

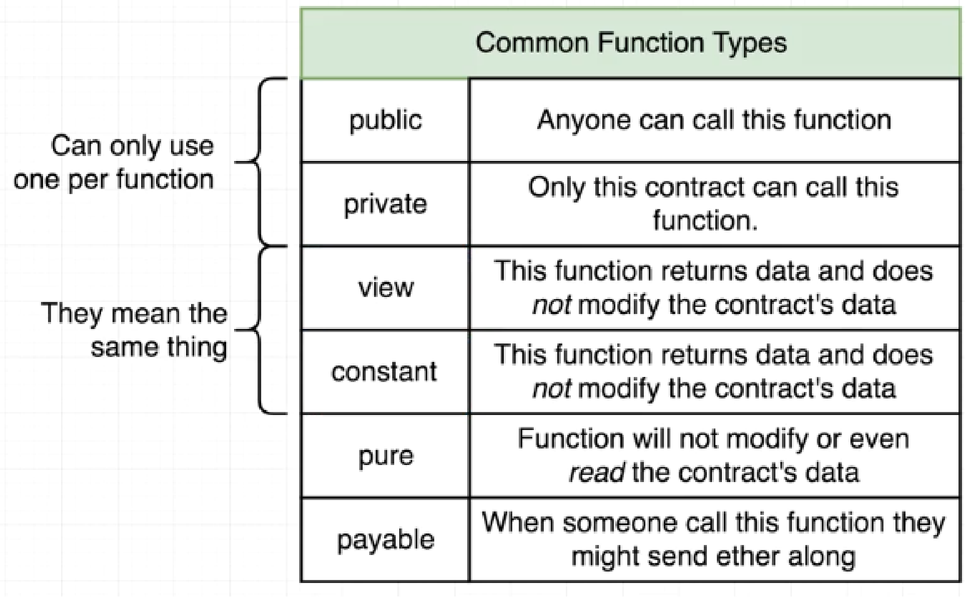
Some Solidity Syntax -

- Function syntax



```
function getMessage() public view returns (string) {
    return message;
}
```

- public and private types dont give any type of security, they are just access modifiers.
- view and constant mean the exact same thing, just that recent versions of solidity mai constant ko view se replace kar diya he.



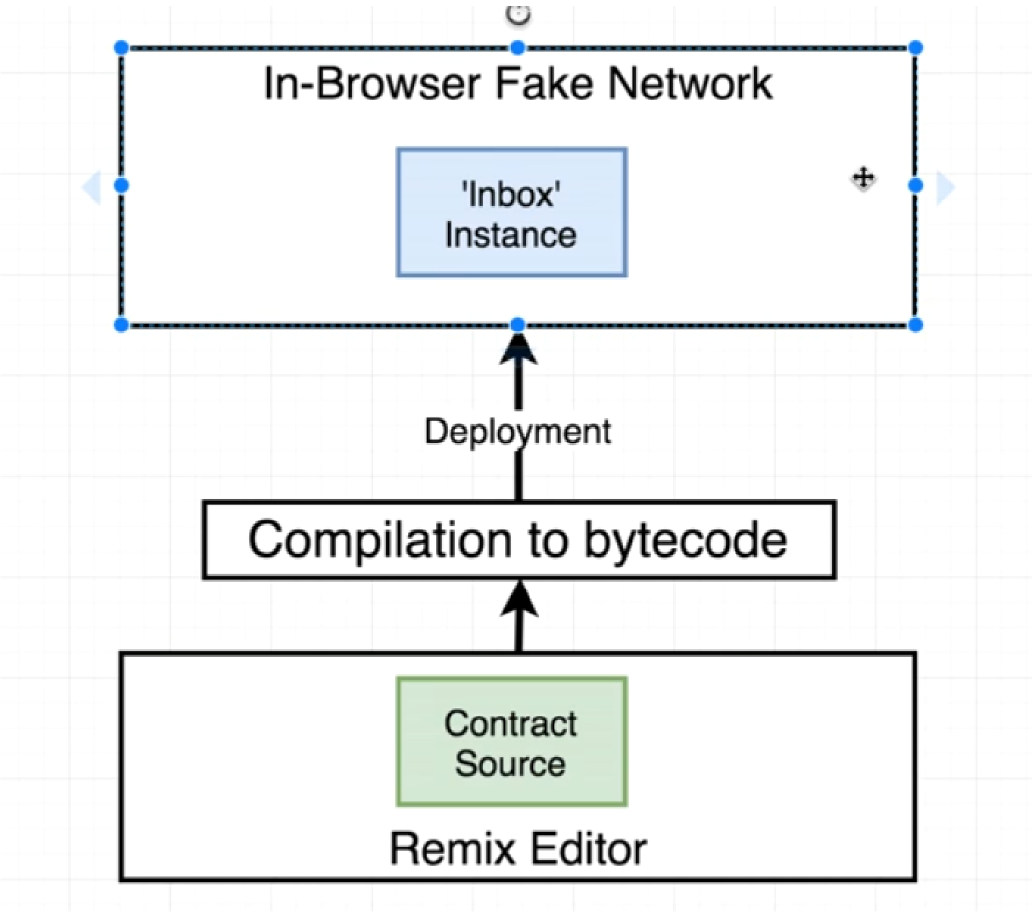
Common Function Types	
public	Anyone can call this function
private	Only this contract can call this function.
view	This function returns data and does *not* modify the contract's data
constant	This function returns data and does *not* modify the contract's data
pure	Function will not modify or even *read* the contract's data
payable	When someone call this function they might send ether along

Can only use one per function

They mean the same thing

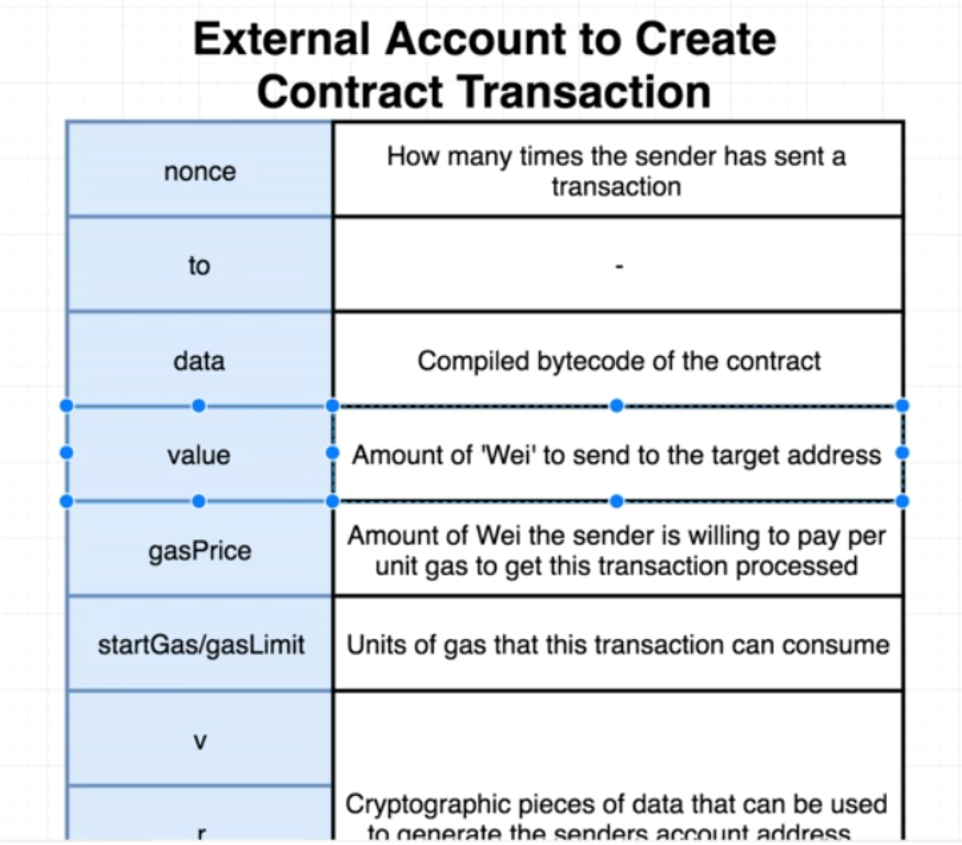
- IMPORTANT - We cant return a value from a function that modifies the contract data.

Working of Remix



Deploying a Contract on a Network -

- Deploying a contract on a network is done by **authoring a transaction** from the creator's account with the purpose of creating a contract on the network.
- Ye hone ke baad contract ka ek account ban jata he target network pe.
- This type of transaction which is used to deploy a contract on a network looks like the following -

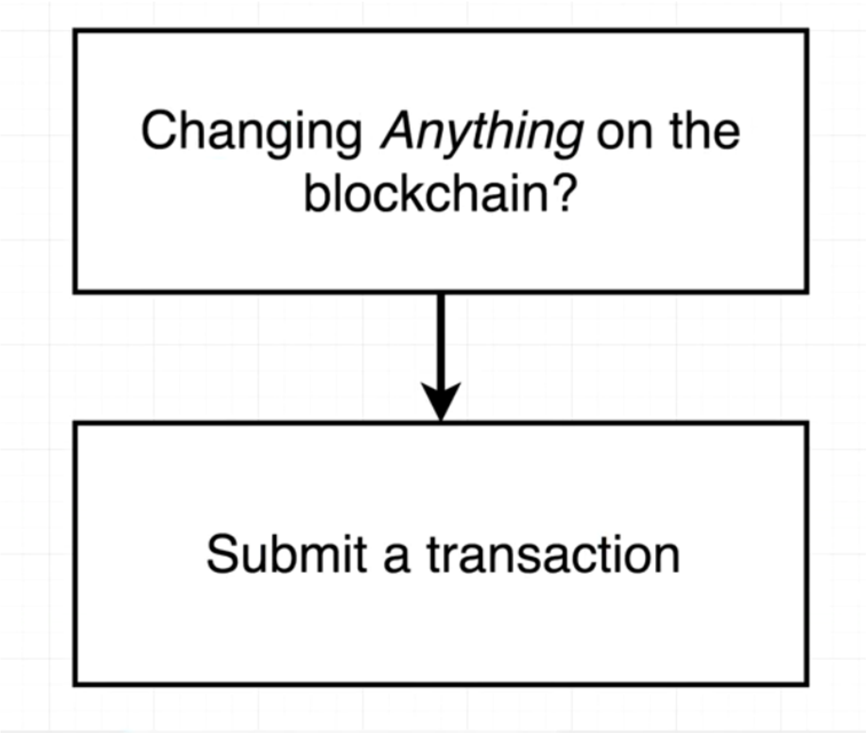


External Account to Create Contract Transaction

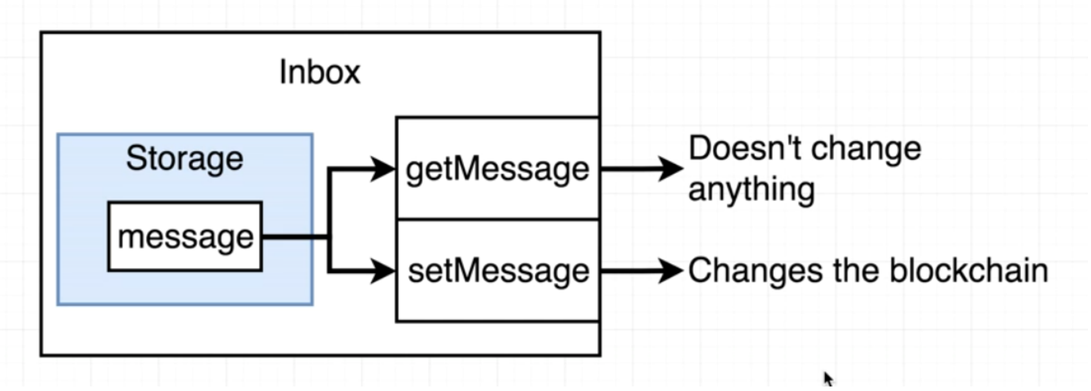
nonce	How many times the sender has sent a transaction
to	-
data	Compiled bytecode of the contract
value	Amount of 'Wei' to send to the target address
gasPrice	Amount of Wei the sender is willing to pay per unit gas to get this transaction processed
startGas/gasLimit	Units of gas that this transaction can consume
v	Cryptographic pieces of data that can be used to generate the senders account address
r	

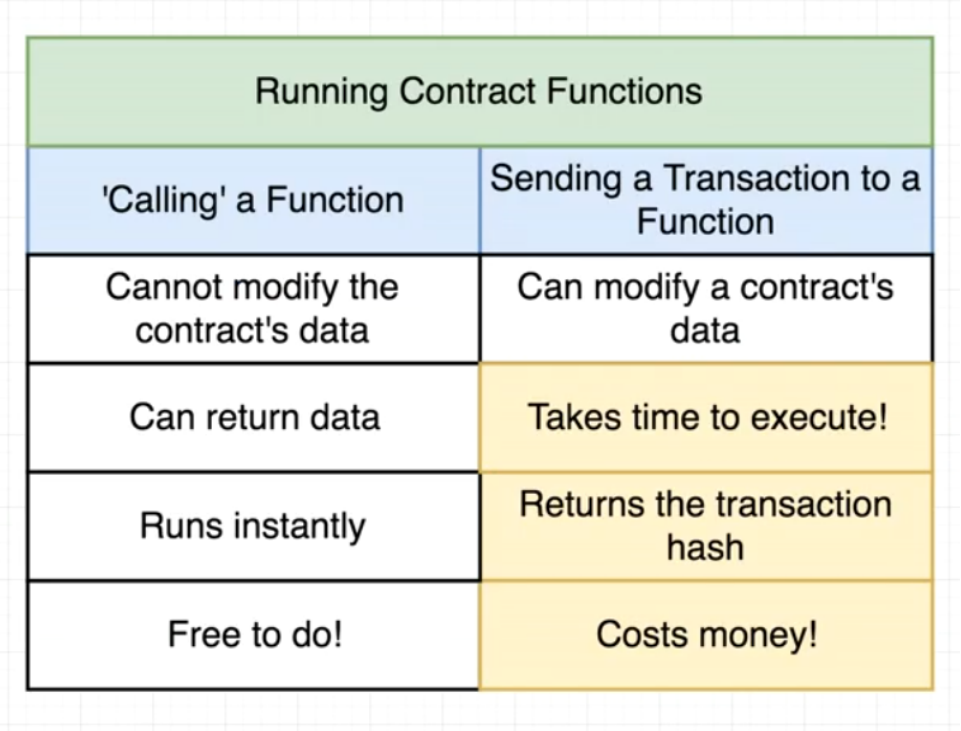
- Yaha v, r & s are just what were present in the transaction we saw earlier.
- IMPORTANT - Is type ke transaction mai **to** wali ppt is left blank.

Updating Data on Blockchain -



- Submiting a transaction is accompained by the mining process.

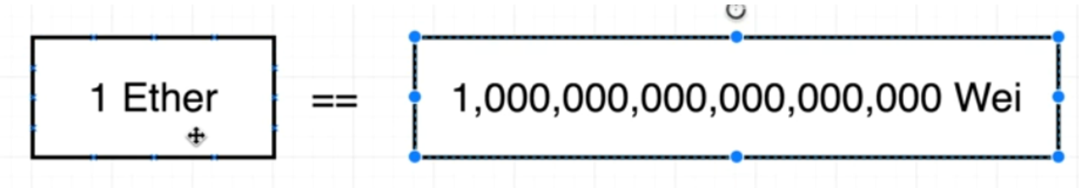




Running Contract Functions	
'Calling' a Function	Sending a Transaction to a Function
Cannot modify the contract's data	Can modify a contract's data
Can return data	Takes time to execute!
Runs instantly	Returns the transaction hash
Free to do!	Costs money!

- Whenever we are trying to update any data on the blockchain network we need to submit a transaction. So, functions which modify data on blockchain actually submit a transaction to the network.
    - IMPORTANT - **Such transactions always return a transaction hash.**
    - Hence, its not possible to return anything else from such functions.
- The functions which do not update any data on the blockchain dont have to submit any transactions, they are just normal functions.

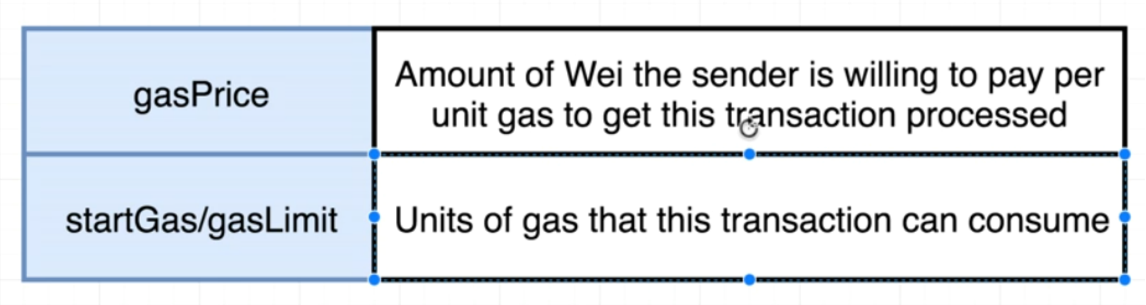
- Ether, Wei, Gwei are just units of measurement.



1 Ether == 1,000,000,000,000,000,000 Wei

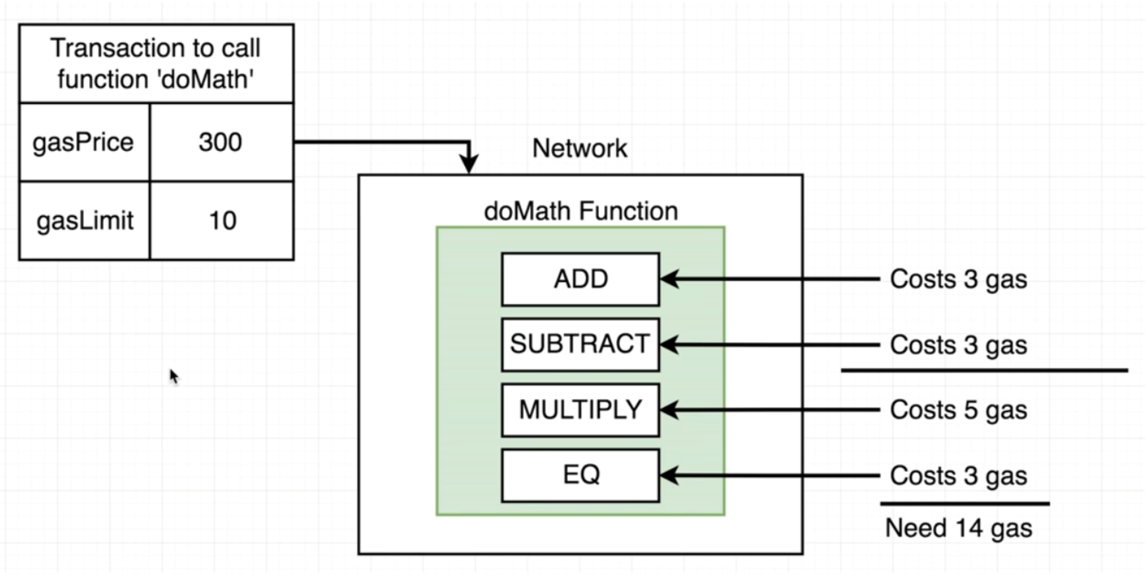
Gas and Transactions -

- As we want to modify data on blockchain → we have to run a transaction on the blockchain → this needs Gas (some amount in ether).
- Just like how it costs us some money if we host our centralized servers on AWS, Azure etc.
- This tells us the scope of the application we can build using Ethereum (web3 in general) as we will have to pay some price to just update data stored on the blockchain.

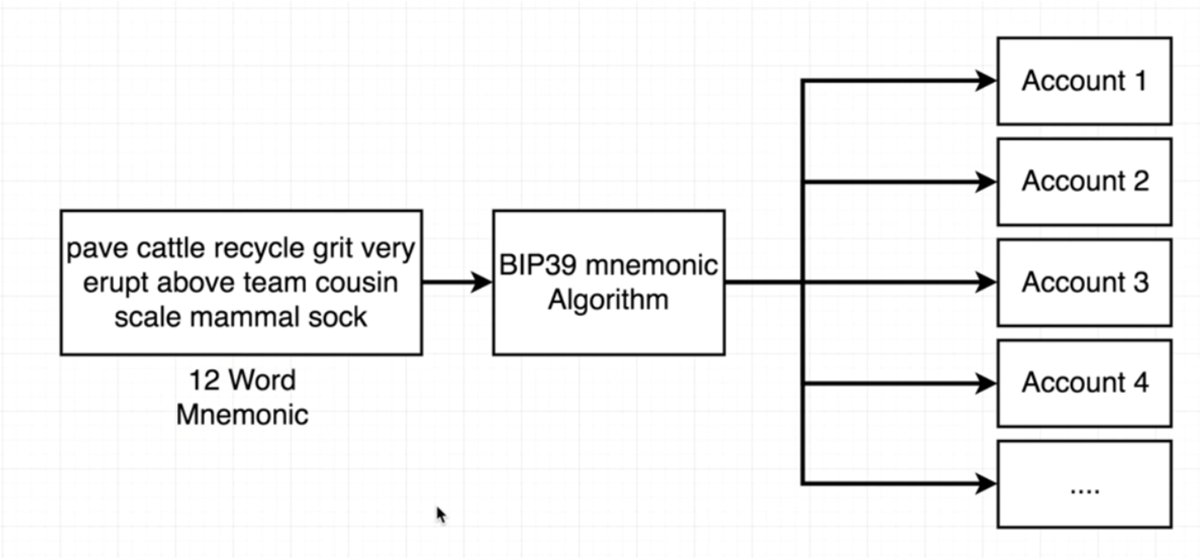


gasPrice	Amount of Wei the sender is willing to pay per unit gas to get this transaction processed
startGas/gasLimit	Units of gas that this transaction can consume

- **Example -**
- Add, Substract operation ke baad meri gas multiply operation ko execute ni karwa payegi, so, execution will stop after substract.
- Also, in case extra gas reh jati he after transaction execution then wo left over gas comes back to the sender.



Mnemonic Phrases -



- Apne accounts ke corresponding address, public key & private key yaad rakhne se acha he ki mnemonic yaad karlo.
- Using it we can retirives our accounts.