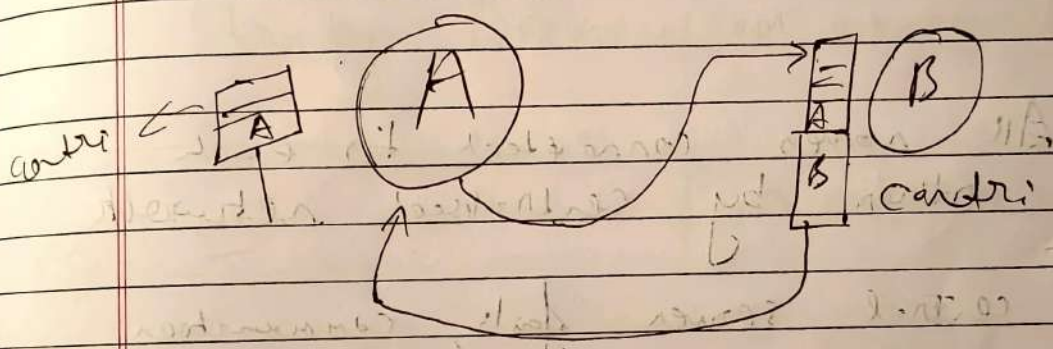
BLOCKCHAIN

Not complex at all.

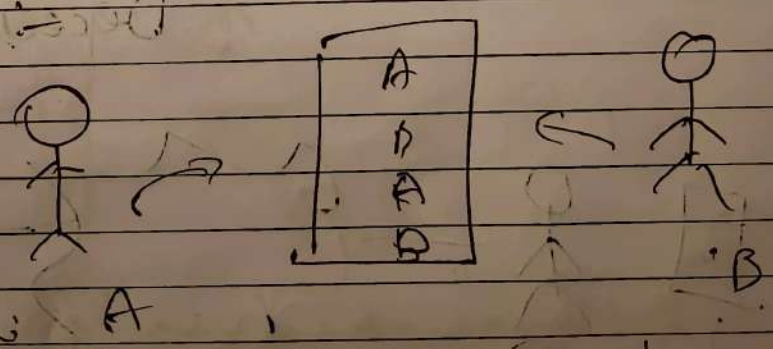
eg: 1. when microsoft come into existence to make report



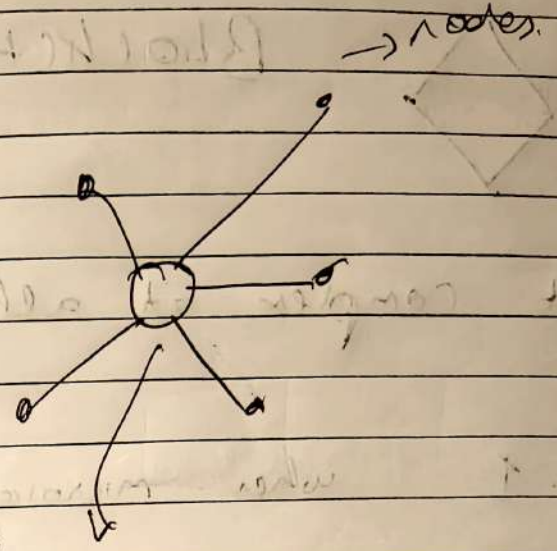
Not a simultaneous nature rahi hai.

Its one after the another.

So the work on some cloud / shareable network.



we see changes happens simultaneously.



Central server

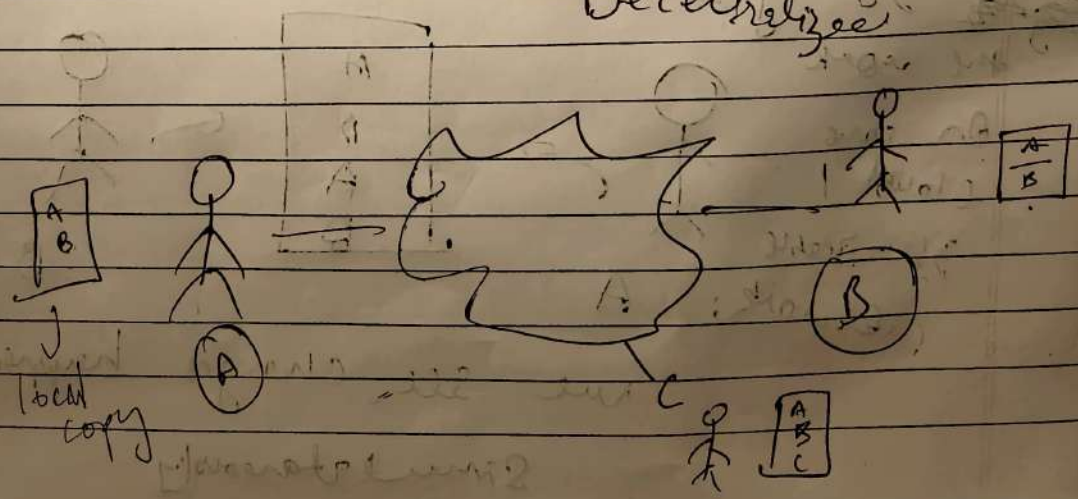
All nodes connected to each other by centralised network

If central server fails, communication breaks

→ so this is centralized

So we propose Blockchain

Decentralized

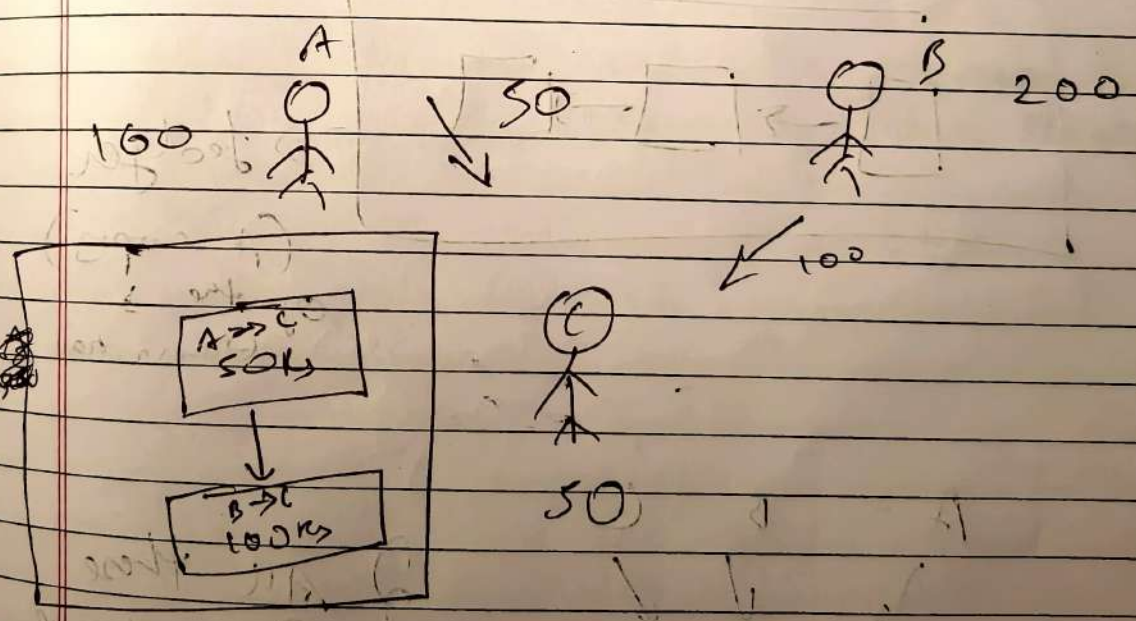


If anyone updates their copy, the same you will be able to see all others copies.

So this blockchain network is maintaing the proper updation of the copies in all users of Brooks.

Strictly Decentralized

eg. 2 :- Blockchain



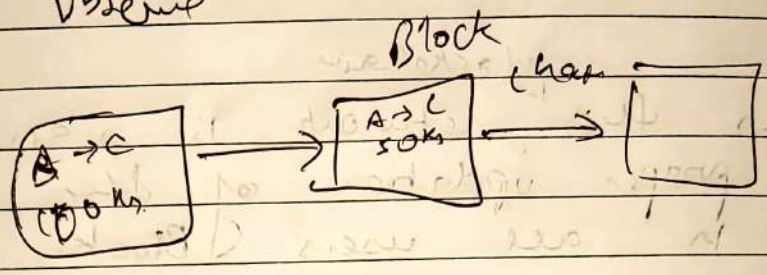
C asks for money to A & B.

→ A says i can give 50.

→ A block is created automatically which contains this information.

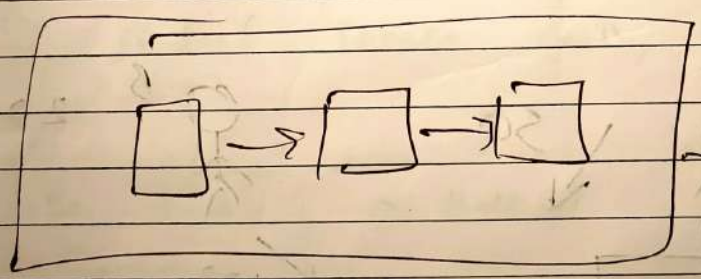
→ D also gave 100 Rs
Block is created.

→ You observe

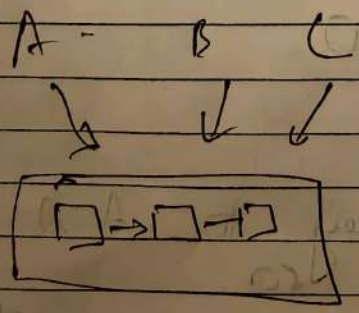


A block is created on each transaction.

These are connected & chained together.



→ Ledger
(A copy) of the transactions



All these guys have a copy of these Blocks

Page No:
Date.

Imp:

B → C 100

B remains with Rs 50. (if he had 150)

now B give 100 to A (while he has 50 only)

In such cases the transaction will not be allowed by the other member (A & C).

Because you have 50 & you send 100.

How? Because all the members have a copy of this ledger.

do you know

Page No:
Date.

Blockchain Definition :-

→ Now we have an idea of Blockchain
Lets understand some of it

All the Blockchain stories have some common words

1. Open (To ALL)
2. Distributed (on Nodes)
3. Ledger (public ledger, A Blockchain)
4. Efficient (Secured, speed, scalability)
5. Verify (eg: A sends 500 who have 100 only)
6. Permanent (Immutable, cant be changed, Tamper proof, Persistent)

Blockchain is a open kind of distributed ledger where all your transactions happen smoothly, efficiently & should be verified & validated + Information is being stored permanently.

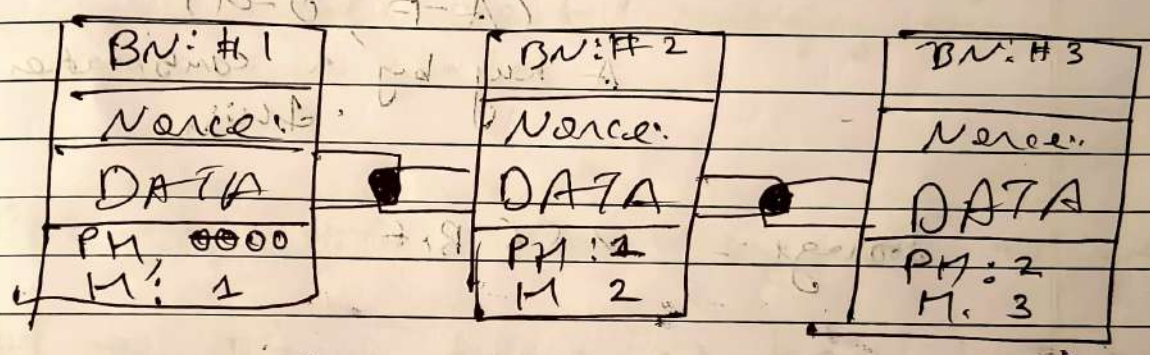
Now !!

Page No:
Date.

whats in the Block.?

Lets take a Block & open it.

Lets understand Block first & then Blockchain.



BN :→ Block number

Nonce :- Number used only Once. IMP number

DATA : Soul of the Block, All security this is for crucial data. Transaction, medical etc. gov etc. nodes, participants etc.

PH :- Previous HASH

H :- Hash (SHA-256)

Page No:
Date.

Hash Value :- A unique identity for a Block.

Example :-

My fingerprint is my unique identity.

---

length of Hash :- 64 digit, character long

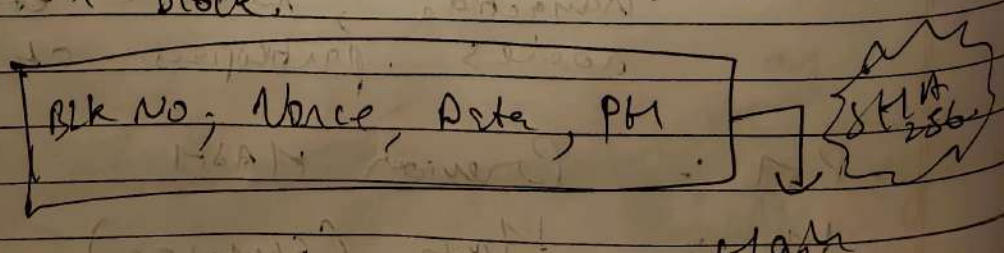
Value :- Hexadecimal
(A-F, 0-9)
A key by a combination of this.

Storage :- 256 Bit

On what basis the Hash is being generated,

what input ? to get Hash

Hash is a unique identifier for a block.



Nonce : Very Important number.

Block number 1 → can't be changed.

Data → we keep data safe, to maintain integrity. we don't modify in later

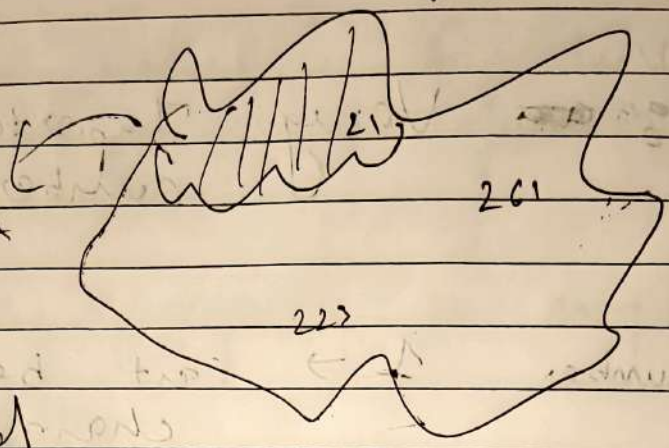
PH → we can't change it as well

H → If, want a hash value for my Block, I am only dependent on my nonce.

So nonce uses Brute force method until the targetted Hash is achieved, that Nonce number keeps on changing until hash is made.

Page No:
Date.

Flash universe



I want my hash to be of this area

eg: 223, i don't want it

eg: 261, i don't want it

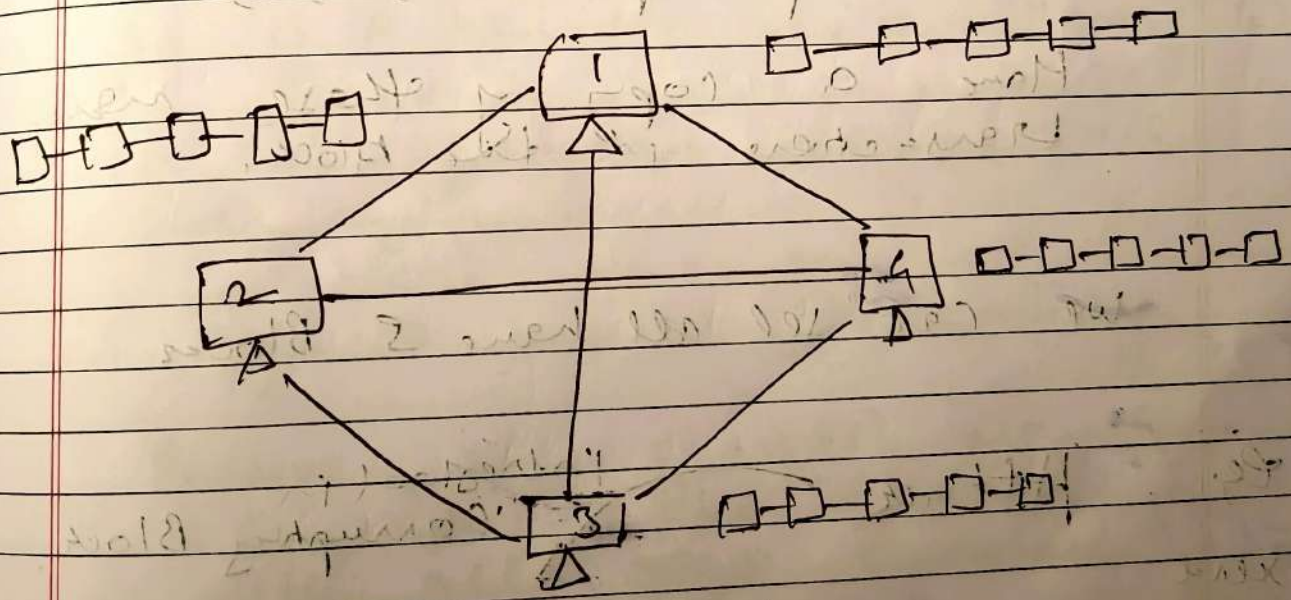
21, its in my area so i take it & use it for hash

Thats why range is used

Page No:
Date.

Lets understand P2P

P2P - Peer to Peer.
Distributed Network



So since the is no centralized, its distributed.

Now its just a scaled down version of a network.

or a big scale.

→ when transaction is correct, cert, a block is created,

1 Block is made for each

So all participants 1, 2, 3, 4

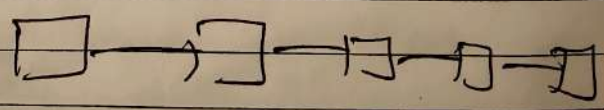
Have a copy of those many transactions in the Block.

We can tell all have 5 Blocks

eg. Attack

Interested in Corrupting Block.

serve



remember :- each Block has hash value.

If he tried to change

A = 100 Rs

to

A = 10 Rs (Modified data, ~~Corrupt data~~)

Since data changed, hash will change, all rest Blocks will change,

So if A had 100 now its showing 10

it will affect rest transactions, all block affected

~~A~~ its not a good thing.

Because all participants are able to match their Blockchain with all other Blockchain.

& when majority sees that its changed

the ~~data~~ Attacker changed one is discarded & the correct Blockchain is ~~protected~~ protected.

So attacker has to modify the majority of Blockchains. to get his thing done.

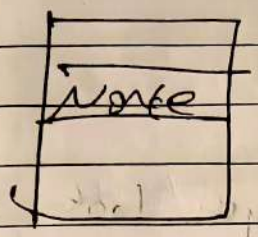
So attackers will have only seeds, to modify more than half of network

→ Because Blockchains are continuously checking on each other making attack impossible.

That's what makes Blockchain security secured & efficient.

Page No:
Date.

Nonce & Mining in Blockchain.



Mining:

Lets understand from Blockchain POV.

To understand this lets understand

Nonce:- The hero of Mining more

→ we change this to get targetted hash value.

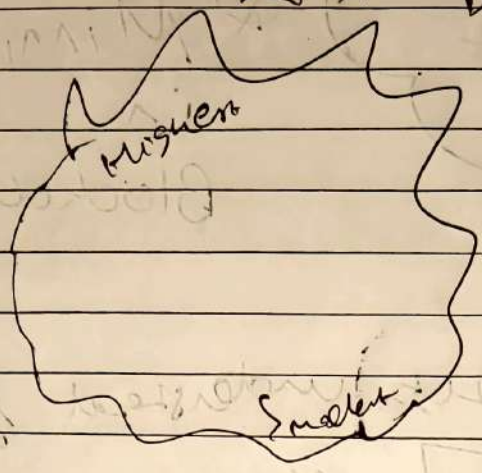
we don't change things in Block, only nonce.

A Block has unique hash Id.

→ unique hash for big long.

Block

All Blockchain hash value (possible)



If you look at extreme ends

(Lower) 00000000 ⟷ FFFFFF (Higher end)

64 Char

(0-9, A-F)

Total 16

(Iske upper value nahi hogi)

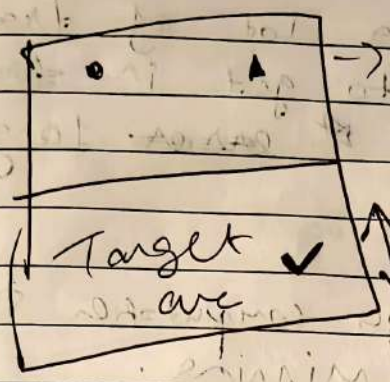
Smallest hash value

Smallest ⟷ Highest

(Iske neche value nahi hogi)

Now here in Blockchain

we fix a targed zone.



→ Block hash universe

If hash in this area, then only accept else deny.

eg If nonce = 29

some hash • → not accepted

If nonce = 23

some hash ▲ → not accepted

If nonce = 362

some hash ✓ → accepted

To do this, Miners are involved

& this process is called Mining.

We are trying to generate a hash value in a target zone

Page No:
Date.

so miners do this!

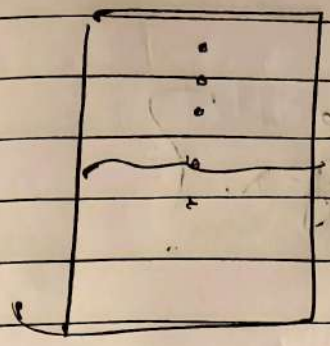
Miners → do a lot of trial error to get in ~~trial~~ ~~error~~ target ar

Required :- High computation to do mining.

Lot of computations are carried out.

miner changes the nonce & get the proper hash, thats how they guess.

Nonce is not predictable, very highly unpredictably that what make it challenging for miner



nonce = 17
= 18
= 19
= 20
= 21
= 22 → pata gaya.

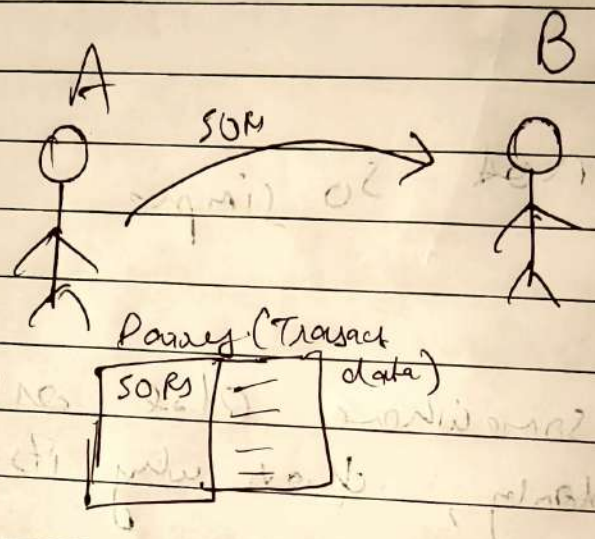
not so simple

All are somewhere else only.
randomly, that's why its hard.

That's all about nonce & related with mining

What is immutable Ledger?

eg:



Transact data, very important :-

Security damage :- page tax, stolen, change term

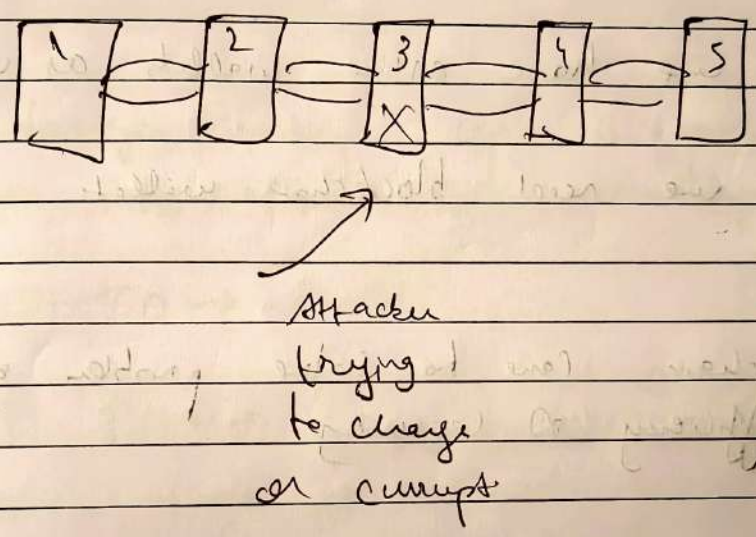
Here it was easy to do this.

What if we put Blockchain in this place

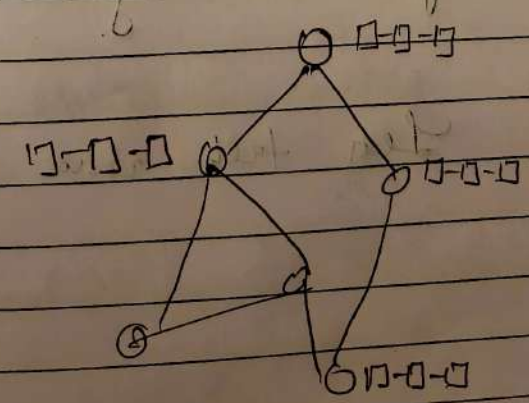
Immutable → not possible to change

Ledger → List of records

Immutable → such a record which can't be changed.



So here when he was changing, at the same time transactions are happening.



Since all nodes are connected & each local copy is present with all!

Since all have copies, after majority blockchain that will be accepted automatically.

Blockchain Wallet

(1) Wallet → open → take out money.

(2) take money → open wallet → put it

So we have online wallets as well.

→ why we need blockchain wallet

→ Blockchain came to solve problem of efficiency & security

→ Same is the purpose of wallet

→ Its fast because no centralized party

→ Transaction fees is very low.

Lets talk about Keys.

Manage crypto wallets.
↓.

Public key :-

To transfer money upi is public, open to all, send money.

(1) UPI ID → PUBIC

UPI PIN → 1, 2, 3, 4, 5, 6 (Private Key)

(2) email id → public

private key → password

---

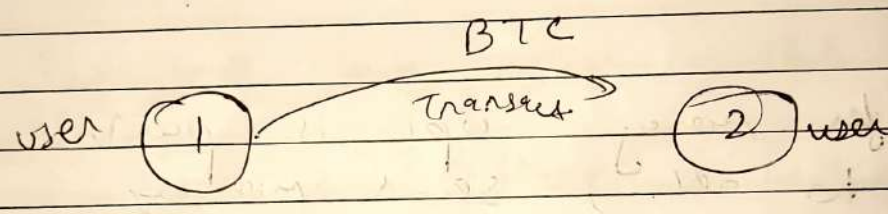
make blockchain wallet

Mempool

↳ Memory Pool

→ what is it ; where its going to be used

→ what the role , where it going to be used



Transaction data :-

Details about Transaction

Who, whom, How much, when.

every 10 mins blocks are being created & that how blockchain ~~...~~ gets long.

T_1

A

when A does transaction, its unverified.

Its possible, that some potential miner

Can be done to disturb the network.

Every user has its own mempool.

mempool → waiting area.

↓

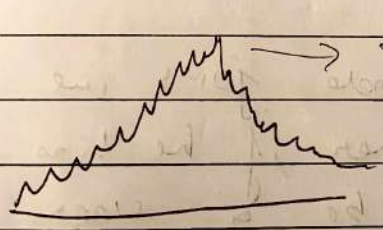
your number will come

you go when you have check

---

when transaction rate are normal

Transaction → mempool → Blockchain.



→ Traffic increases some times

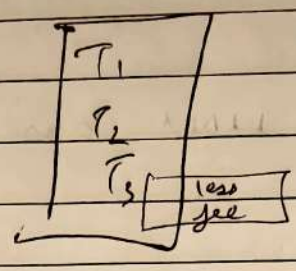
mempool also has some capacity

If you have to make new transaction, you have to remove the previous ones from mempool

With every transaction, we have transaction fee.

Higher transaction fees ones are treated first.

people use RBF → Replace by fee.



If someone wants their transaction to get authorized fast they use higher transaction fees.

Types of Blockchain.

1. Public → All people know everything, Blockchain its transparent in nature.

   → If a new node joins, we can call it miner if he has complete. Rest he can be a user.

2. Private → Private to some organisational company.

   → All employees will be node,
   → No entry to outsiders.

(3) Consortium / federated Blockchain :-

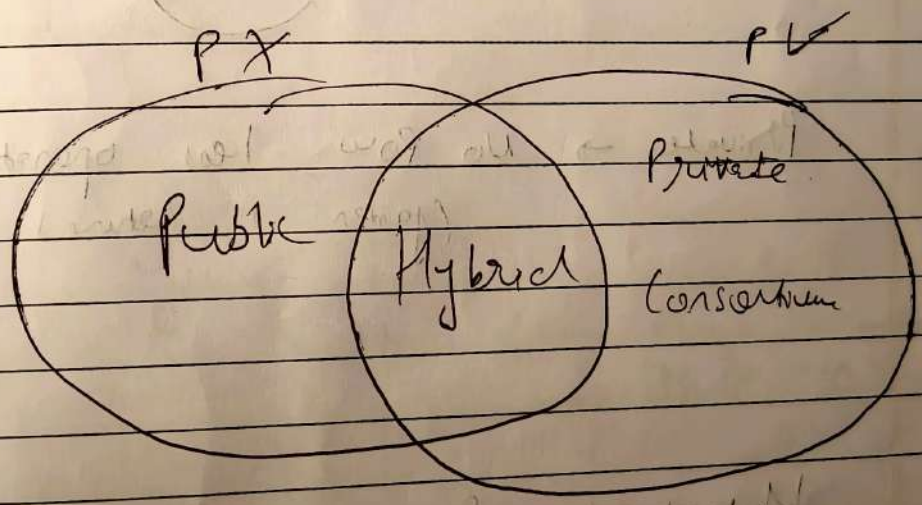
Made for a specific organisation

Then a group of companies joined

Decentralized nature grows.

Public is most decentralized

(4)



Hybrid :- Best of 2 worlds,
Has qualities of both

when we have P_b → Permissionless.
(Anyone can enter as reader here)

PV → Permissional Blockchain

Hybrid → future of Blockchain
flexible in nature

When to use whos quality its smart

eg:- Public → Takes time for operation
POW

Private → No POW, less operation times
(faster in nature)

Hybrid → Security high
Transaction fees :- very less.

Hard fork

→ Change / upgrade.

when you make some something, its always going to change & upgrade.

eg : Youtube has upgrades

There are a few times we have to Hard fork.

Yes
- old
- new. (some change necessary in app for all)

→ smart contract → can't change

→ ~~so~~ we had to change smart contract, we have to upgrade

2 communities

- ethereum. new. (new one because they need upgrades)
- ~~eth~~ classic

they are okay

→ Tokens :- given to both.

Miner → have 2 choices.

new rule to code

→ Reasons :- (1) correcting security risks
(2) adding functionalities
(3) Reversing Transactions.

Soft fork (do again)

→ Backward - compatible.

Some nodes are upgraded
&
Some did not upgrade.

But their relations are good

Hard fork
is not backward
Compatible

→ If a software has new version, new rules

old ——— + new rules → new

we added new rules which will clash with old rules.

→ subset rule.

I want block size of 1Mb. (old rule)

I want block size of 0.4Mb. (new rule)

New rule is subset of old rule..

- we as user use updates software, majority guys upgrade, some don't.

Eg.

old 1MB

new rule 0.4MB

Old [0.3]

new

[0.6]

Page No:
Date.

Blockchain Layers.

Smart contracts, DApps, UI — **Application Layer** — (5)

Transaction validation, POW, POS. — **Consensus Layer** — (4)

P2P, Internode communication — **Network Layer** — (3)

Blocks, Inf transactions, we link blocks, Digitally signed, Immutable — **DATA Layer** — (2)

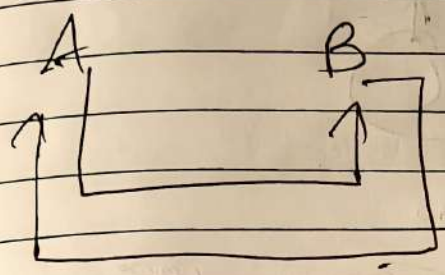
Node, computers, Infrastructure, Network — **Hardware Layer** — (1)

Page No:
Date.

Summary.

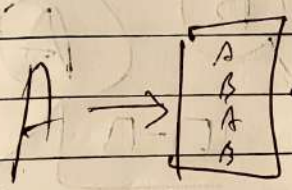
Microsoft

(1)

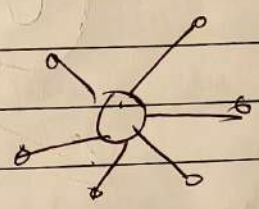


one after other

(2)

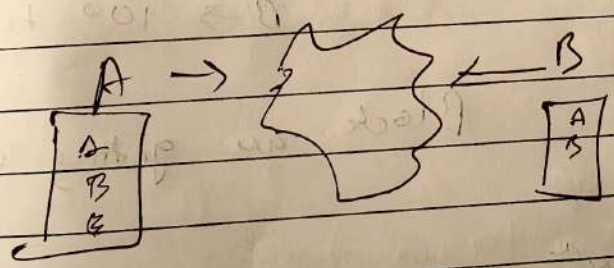


cloud.



→ centralised server node may destroy

(3)



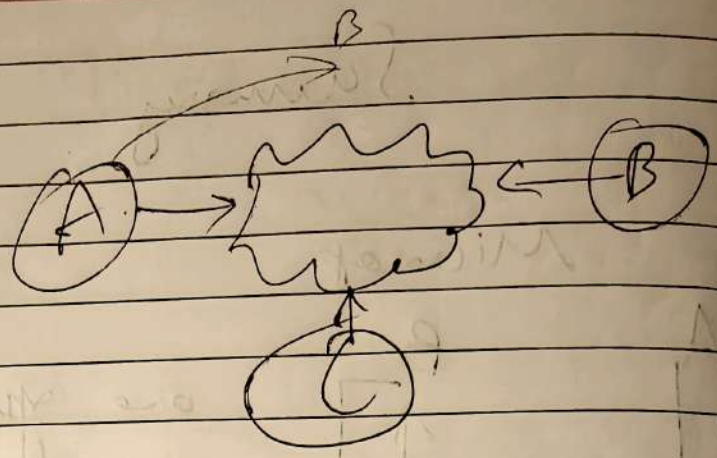
So here we propose blockchain

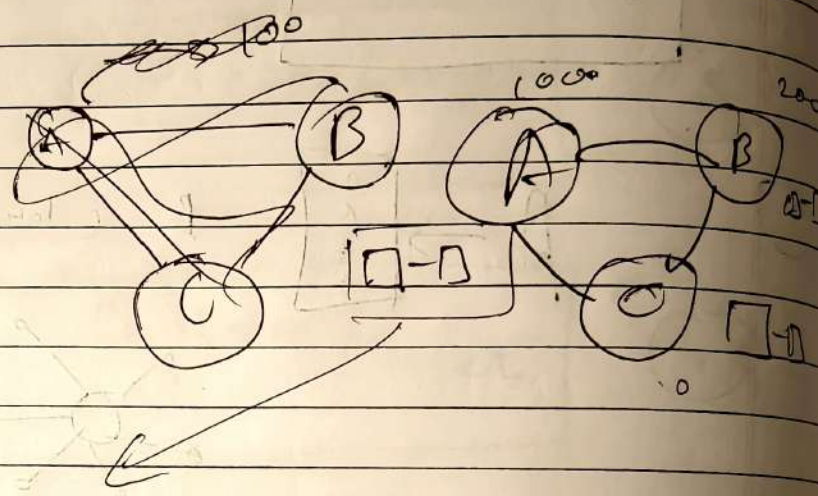
A decentralised approach.

copies will be maintained of your charge.

they will have

e.g. 2





These of are transactions nothing but ledger.

immutable ledger once added can't be changed

A → 50 to C

B → 100 to C

Block are getting created

If B → A 2000

In this case it won't be done because, it goes for validation & everyone has their own copy to verify, so not possible

not allowed by ABC

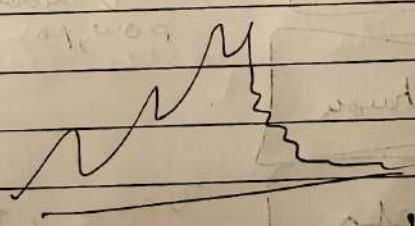
Page No:
Date.

Blockchain definition

1. Open to All
2. Distributed on nodes
3. Ledger (public, ledge, Blockchain)
4. Efficient (secure, speed, scalable)
5. Verify.
6. Permanent (Immutable, cannot be change tamper proof.

Mempool & RBF.

Transaction → waiting area → verifies → Block



High Real time, people do RBF

Replace by fees faster & with high fee

Date.

→ Blockchain wallet :- Public key
upi
Private key → Pin

→ Types of Blockchain :-

(1) Public (Permission less)

(2) Private (Permissioned less)

(3) Federated, Consortium

(4) Hybrid Blockchain Both features

→ Blockchain Layer

Application → Smart contract Dapps.

Consensus → ~~Transacto~~ Transactions Validation, POW, POS.

P2P, Internet connection

Network

Data → Transaction, Immutable ledger, Blocks, Digital signature

Nodes, Computer, Miner

Hardware

Page No:
Date.

Timestamp :

Block : #2

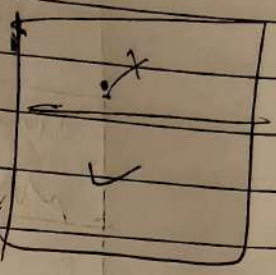
Nonce :

Data :

Prev Hash :

Hash :

Target



eg:
000000

Hash
A value starting with 6 zeros. then only it will be accepted.

Nonce : 32 bit.

$2^{32} \approx 4$ billion.

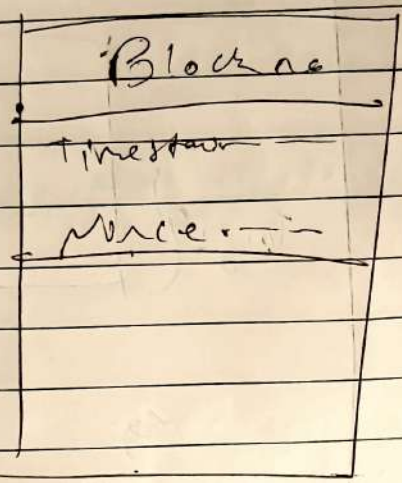
0 ——— 4bn.

Nonce range.

If we exhaust all the nonces, still its not guaranteed we will get unique hash value.

Date.

why because unique hash has very minimum value.



we have a new Thing called Time stamp.

1529533115
unix time

Eth random

10 digit number

1 Jan 1970, each second this updates once.

In how many seconds we can exhaust this 4bn

Page No:
Date.

Miners,

How many hash per sec.

100 mln Hash per second

$$\frac{400 \text{ bln}}{100 \text{ mln}} = 40 \text{ sec.}$$

In 40 sec it exhaust.

Q unit time is being updated each second

Eg.

Timestamp : |——|

Nonce : 0 • • 46

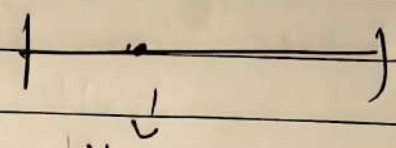
1 sec

it reached here

miner generates hash value, until acceptable zone.

Page No:
Date.

I see



it reached here

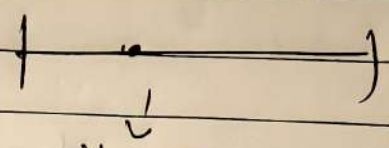
unit timestamp get updated

⟷ you can again use that, you can exhaust

So before you reach here, it will get updated.

So this help to solve such problem in a given range.

| see



it reached here

unit timestamp got updated

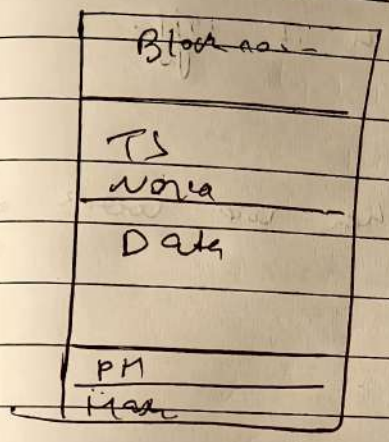
⟷ you can again use that, you can exhaust

So before you reach here, it will get updated.

So this help to solve such problem in a given range.

How Miners Pick transactions.

we solved Nonce & Timestamp.



Nonce → 0 —— 4bn

Timestamp → updated every second.

In previous we considered Avg Case scenario

100 MH/s.

we were generating at this rate

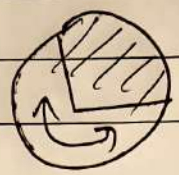
Best case scenario:-

A miner have very heavy computer

→ Bln, Trillion, hash per second.

In some fraction of second.

it finishes this, exhausts this.



in less than a second, it can finish it off.

This is a problem, when we work on high scales

what the solution

→ we take help of Data

But we can play with data

eg: Data =

Trans = T_1, T_2, T_3 } Same fee. } 3, 2, 1

The miner, the look at mempool

100's of transactions

we can't take all

Miners pick on priority of gas fee.

How will data help?

Configuration change:-

If we can't get hash value

→ Try with, low transaction than that.

& you change data, get new combination.

So our miner will never sit idle

Works Algorithmically.