Blockchain has had a very important impact in our life but still have its problems for example their scalability or gross energy consumption or poor or limited performances or its immature governance mechanism.

So there was some solution that were trying to improve these things in Bitcoin and other blockchains

For example SegWit or some blockchain that has tried to solve some of other blockchains problem such as lightning that has tried to solve the scalability problem of Bitcoin transactions using microtransactions but this network may not be suitable for more generalized scalability.

The ideal solution to these problems is that to let parallel blockchain works together and keep their own protocol which using the PoW is very difficult.

Merged Mining for example allows the work done to secure a parent chain to be reused on a child chain

But it makes it vulnerable to attacks if the majority of the hashing power do not merge-min the child chain.

Cosmos is a network of blockchains that can work parallel which are called zones.

The first zone connected the cosmos network is called cosmos hub which is a multi-asset PoS cryptocurrency with a simple governance mechanism that can simply be upgraded. the cosmos hub can be extended with connecting other Zones to it and those Zones can interoperate much simpler considering money or time consumption .

These zones communicates with each other using Inter-Blockchain communication (IBC) a simple communication protocol like virtual UDP(User Datagram Protocol : A protocol that is being used to communicate throughout the internet) or TCP(Transmission Control Protocol) for blockchains.

So the tokens can be transformed from one zone to another securely and quickly without needing for exchanging liquidity between zones instead all the tokens transfer through the cosmos hub which keeps the track of tokens in the inter-zones.

The hub isolates the Zones in case of the failure of one of the Zones and since any zone can connect to the Hub so Cosmos is flexible for future-compatibility of blockchain innovation

**THE consensus PROTOCOLE :**

Cosmos Network uses Tendermint consensus protocol.

Tendermint :

Validators: In the classical Byzantine fault-tolerant each node has the same power and voting right but in Tendemint Consensus Protocol the zones has non-negative different amount of voting power.

The zones that helps in validating and consensus that have positive voting power are called **validator.**

The validators’ voting power are determined in the genesis block or depending on the zone can be determined later for example the voting power for PoS such as cosmos hub may be determined by amount of staking tokens bonded as collateral.

Consensus: Tendermint is a partially BFT consensus protocol derived from DLS consensus protocol.

Tendermint is notable for its simplicity and its performance.

For cosmos in order to consensus on a block it needs a fixed set of validators which will become to consensus for one block at a time. That block will contain transactions and the consensus will be determined in rounds in which that every round has a round leader or proposer that propose a block and that leader will be choosed base on the voting power of validators that have consensus on that block and then the validators will decide to either accept the proposed block or going to the next round.

Tendermint’s security derives from its use of optimal Byzantine fault-tolerance via super-majority two third voting and a locking mechanism.

In this way two things will be ensured :

1. More than a third of majority must be Byzantine in order to violate the safety in case that more than one value is committed .
2. And if a group of validators succeed in violating the safety they will be recognized by the protocol. this include both voting for conflicting block and broadcasting unjustified votes.

Despite the strong guarantee Tendermint has a very good performance and the tests has shown that it can process thousands of transaction per second with commit latency of one to two seconds and this will hold even if the chain is in hash adversarial for example when validators crashing or broadcasting malicious crafted votes.

**Light client** (a node that is not full node and is only communicating with the blockchain through the full node):

Tendemint simplified light client security makes it a good case for the mobile development and internet of things.

While bitcoin light client should sync the block headers and find the one with the most proof of work with the Tenderemint light clients only has to store the set of the validators and then verify the more than two third precommits in the latest block to determine latest state.

**Preventing attacks**:

Tendermint has protective measures for preventing certain notable attacks, like long-range-nothing-at-stake double spends and censorship.

**ABCI:**

The Tendermint consensus protocol is implemented in a program called Tendermint core which is an application-agnostic consensus engine that can turn any blackbox application in to a distributedly replicate blockchain.

The Tendermint core communicates with blockchain via application blockchain interface(ABCI). Thus ABCI allows the blockchain application to be written in any language not only the language that consensus engine has been written with. ABCI makes it possible to easily swap out the consensus layer of any existing blockchain stack.

Here an example for someone that wants to program a Bitcoin-like coin in cosmos.

What will Tendermint engine will be responsible of:

1. Sharing the blocks and transaction between nodes
2. Establishing a immutable order of transaction **(The blockchain)**

What will ABCI be responsible for:

1. Checking for digital signature and transactions
2. Maintaining UTXOs database
3. Preventing transactions from non-existing fund
4. Let the nodes query the UTXOs database

**Inter Blockchain-Communication (IBC):**

Let’s consider three blockchain on the cosmos blockchain

“Zone1”, “Zone2” and the “Hub” and suppose we want to send a packet from one zone to another first there will be a proof posted in the receiving chain.

That proof state that the sending chain wants to send a packet destinated for the receiving chain to check this proof it must be able to keep up with the block’s header.

This mechanism is similar to the sidechain protocol which requires that two chain be aware of each other using a bidirectional stream proof-of-existing datagram.

The IBC protocol can be defined using two types of transactions:

1. IBCBlockCommitTx : that allows to any observer to see its recent block hash.
2. IBCPacketTx: Which allows a blockchain to prove to any observer that indeed the packet was published by sender’s application via a merkle-proof to the last recent block-hash.