XinFin Mainnet - DPoS Requirement

DPoS - Delegated proof of stake

Simple Summary

Design and development for a delegated proof of stake consensus mechanism for the XinFin public main network. (Public network shown in Figure 1.3)

Abstract

All blockchain systems are fundamentally a deterministic state machine to achieve consensus, the process of agreeing on the order of accurate transactions and filtering out invalid transactions. While mainstream blockchain system has been focused on Proof-of-Work (POW), Xinfin is interested in using DPOS, a robust, secure and efficient consensus algorithm that can produce an equivalent ordering of transactions, to solve the problem of scalability, privacy, and interoperability. The main issue with all consensus algorithms is that block producers can cause censorship as all blocks must be valid according to the deterministic open source state machine logic.

XDC01, the first protocol by Xinfin, will have masternodes maintained by institutions to ensure the problems discussed will be mitigated. Masternodes will constitute the validator nodes in case of a delegated proof of stake consensus. With 3 years of successful operation on BitShares and a year of Steem we have experienced all manner of network conditions and software bugs. DPOS has successfully navigated this environment and demonstrated its ability to maintained consensus while processing more transactions than any other blockchain.

Summary of DPOS Algorithm

Delegated Proof of Stake (DPOS) was invented to be an improvement to Proof-of-Work by requiring less hashing power, being less centralized, increasing scalability and being more secure while still providing the same effects as POW.

Witness

In a DPOS system, every individual who participates in the consensus can be elected to be a Witness, an individual who get paid for validating and creating blocks. As the community grows, the ability to remain the witness becomes more competitive which will help increase security as there will be pressure to excel and participate more to the network. In the case of malicious actors, the community can vote to remove an actor to ensure the security of the system and recognize new valuable members. Regardless, the incentives of income and repetition will prevent witness in the top tier to become malicious.

Staking/Voting

In a DPOS system, users will vote to select the witnesses that they believe in the most. The witnesses that gain the most vote will become top tier witnesses, letting them earn income for their services. If a user has a large amount of staking power, the user can delegate his/her vote to others to vote in behalf of the user.

Delegates

In a DPOS system, users will also vote to select the delegates, trusted parties responsible for maintaining the network. Instead of validating transactions and creating new blocks, delegates oversee blockchain protocol. Delegates allow for governance, the ability to produce, maintain, or change the inputs that make up a blockchain, in DPOS systems. Once the delegates propose a change, the users of the system will vote to determine if to adopt it. An example could be changing the average time for a block to be created.

"To help explain this algorithm I want to assume 3 block producers, A, B, and C. Because consensus requires 2⁄3 + 1 to resolve all cases, this simplified model will assume that producer C is deemed the tie breaker. In the real world there would be 21 or more block producers. Like proof of work, the general rule is that longest chain wins. Any time an honest peer sees a valid strictly longer chain it will switch from its current fork to the longer one."

For more examples and different scenarios of DPOS:
https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper

Conclusion:
Delegated Proof of Stake ensures DPOS will be a more efficient system that provide security even if a minority of the producers are corrupt.

The advantages of Delegated Proof of Stake:

1. System is more scalable than Proof of Work and Proof of Stake.
2. DPOS can continue to function when a majority of producers fail.
3. Producers can be replaced with a community vote.
4. System is more energy efficient than POW hashing algorithms.
5. Users are required token to participate in the consensus making it less centralized (no mining pool).

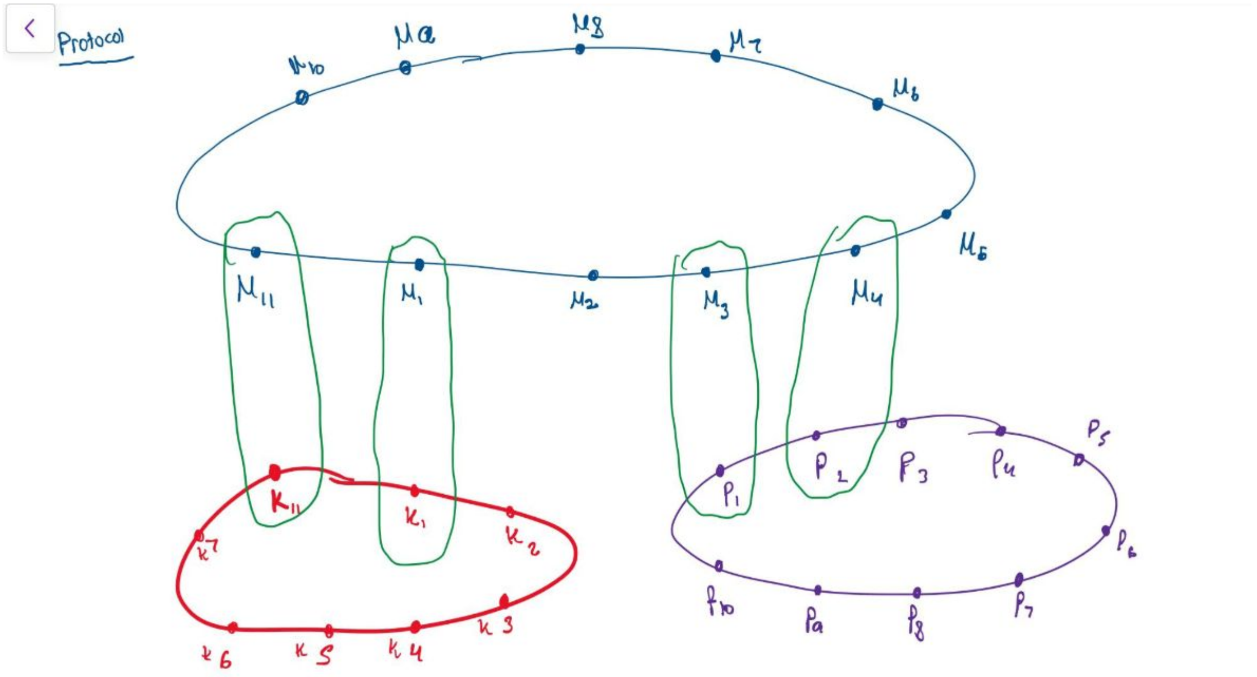
For more about DPOS:
Bitshare Use Case
Steemit Use Case
Hackernoon Summary
Lisk Use case

XinFin

Summary

Design a private network that is independent of the Ethereum chain to ensure fungible and non-fungible tokens are able to be transferred between private and public networks without fees. By making the private network independent of gas, any cryptocurrencies will be able to use the transfer back to the fiat value market.

Architecture of XinFin



Blue chain - XDC Private Network
Red Chain - Trade Finex
Purple Chain - Any other blockchain
Green Circle - Linked Nodes

The image above illustrates the architecture of XinFin. Nodes that are connected are the same user (Example: M_{11} and K_{11} are the same user but on different blockchain). In theory, tokens on the TradeFinex can be transferred to any other blockchain using XDC as the medium. XDC will be a DPOS network to ensure there is no centralization on the mainnet.

Implementing the XDC Arictecture using current projects

<u>Tendermint (General Idea):</u>

Tendermint is software for securely and consistently replicating an application on many machines. Tendermint is a partially synchronous BFT consensus protocol derived from the DLS consensus algorithm.

Properties of Tendermint

1. Simplicity

2. Performance
3. Fork-accountability

For a more descriptive understanding of Tendermint.

Validators:
In Tendermint, validators either have non-negative amount of voting power or nodes that have **positive voting power (these are the validators)**. Validators participate in the consensus by broadcasting cryptographic signatures, or votes, to agree upon the next block. The voting power of a validators is **determined at genesis or are changed deterministically** by the blockchain. **Non-validators can delegate their staking token (Atom)** to any validator to earn a portion of blockfees and atom reward.

The Voting Process:
Voting for consensus occur during each of the block additions

1. During each round has a **round-leader (proposer) who will propose** a block
   - The proposer is determined by the ordered list of validators based on voting power.
2. Afterward, the validators then vote to accept or reject the block

Limitations on the Number of Validators:
Because a Tendermint blockchain gets slower with more validators, there is **a limit to how many validators there are** to ensure very fast transaction confirmation times.

The system works like this: **On genesis day, the maximum number of validators will be set to 100, and this number will increase at a rate of 13% for 10 years, and settle at 300 validators.**

Becoming a Validator:
Anyone can become a validator at any time, except when the size of the current validator set is greater than the maximum number of validators allowed. When the maximum number of validators allowed is exceeded, **a new user can become a validator** if and only if **the user has the amount of Atom greater** than the current smallest validator. When a new validator replaces an existing validator in such a way, **the existing validator becomes inactive** and all the atoms and delegated atoms **enter the unbonding state.**

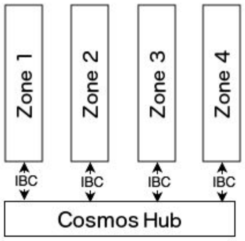
Penalties for Validators:

For any intentional or unintentional deviation from the sanctioned protocol:
This will result in the **validator losing its good standing** and **its bonded atoms** as well its **proportionate share of tokens** in the reserve pool will get **slashed.**

Being Inactive:
After the *ValidatorTimeoutWindow* blocks, if a validators commit vote is not included in the blockchain more than *ValidatorTimeoutMaxAbsent* times, that validator will become inactive, and lose *ValidatorTimeoutPenalty* (DEFAULT 1%) of its stake.

(This is to ensure that Validators are completing their responsibilities.)

Cosmos Architecture



What is Cosmos:
A system able to have multiple **parallel** blockchains to **interoperate** while **retaining** their security properties. The system is broken into the Cosmos Hub and Zones.

Zones:
Blockchains that are **powered by Tendermint Core** to provide a **high-performance, consistent, secure PBFT-like consensus engine**. This is the key to scaling Cosmos.

For zones to communicate between each other, they must **post Merkle-proof as evidence** that the information was sent and received. This is called **Inter-Blockchain communication** (IBC). Zones cannot transfer more tokens than it has but can receive tokens from other who have them The security and the consensus mechanism is **different for each zone and does not have to follow the protocols of the Cosmos Hub**, so it is **the responsibility of the users to trust the zones.**

Cosmos Hub:
The first zone on Cosmos that is a **multi-asset proof-of-stake cryptocurrency** using a simple **governance mechanism.** It is connected to all the other zones and will **communicate** with the other zones **using a IBC protocol** (inter blockchain communication). The purpose of a hub is to **enable quick and secure communication between the zones** thus token transaction can occur easily.

For this system to work, all inter-zone token will go through Hub and Hub will **track the total amount of token in each zone.** The hub is **responsible for preserving the global invariance** of the total amount of each token across the zones.

Security of the Cosmos hub is **paramount important because it is the central ledger** so it must be secured by a globally decentralized set of validators that can withstand the most severe attacks.

Note that the every zone is isolated from each other which allow for future compatibility with new innovations.

The Atom Token:

While the Cosmos Hub is a multi-asset distributed ledger, there is a special native token called the atom. **Atoms are the only staking token** of the Cosmos Hub. Atoms are a license for the holder to vote, validate, or delegate to other validators. Atoms are **required as transaction fees** which will be **rewarded to delegators and validators.**

Transaction Fees:

Validators **can accept any token type or combination of types as fees** for processing a transaction. **2% of the transaction fee will become ReserveTax** (DEFAULT 2%) and will go toward **the reserve pool to increase the reserve pool and increase the security** and value of the Cosmos network.

Incentivizing Hackers:

To encourage users to report and discover exploits, the Cosmos Hubs **rewards the user when the user submit a *ReportHackTx*.** When the exploit is confirmed, the validator and delegators will become inactive, **HackPunishmentRatio (default 5%) of everyone's atoms will get slashed**, and HackRewardRatio (default 5%) of everyone's atoms will **get rewarded to the hacker's bounty address.**

Governance Specification:

The Cosmos Hub is not defined by the protocol as it can **coordinate changes to the blockchain**. To ensure all proposal is viewed, validators are responsible for **voting in a timely manner or they risk being deactivated**.

NOTE: Each zone **have their own constitution and mechanism** (Ex. Cosmos Hub would have no roll-back but certain zones could).

For each proposal, voters may vote with the following options:

1. Yea
2. YeaWithForce
3. Nay
4. NayWithForce
5. Abstain

To understand how the system will work:

A strict majority of Yea or YeaWithForce votes (or Nay or NayWithForce votes) is required for the proposal to be decided as passed (or decided as failed), but 1/3+ can veto the majority decision by voting "with force". When a strict majority is vetoed, everyone gets punished by losing VetoPenaltyFeeBlocks (DEFAULT 1 day's worth of blocks) worth of fees (except taxes which will not be affected), and the party that vetoed the majority decision will be additionally punished by losing VetoPenaltyAtoms (DEFAULT 0.1%) of its atoms.

Light Client:
By using Tendermint, validators only need to **keeps up with changes to the validator set** to verify transactions. This makes it very useful for mobile and internet of things use cases.
ABCI (Application Blockchain Interface):
ABCI allows for blockchain applications to be programmed in any language, not just the programming language that the consensus engine is written in. This means that communication between **different** types of blockchains can occur.

Using Bitcoin as an example: if one wanted to create a Bitcoin-like system on top of ABCI

I. Tendermint core would be responsible for
   1. Sharing blocks and transaction between the nodes
   2. Establishing the blockchain (immutable order of transactions)

II. ABCI application would be responsible for
   1. Maining the UTXO database
   2. Validating cryptographic signatures of transactions
   3. Preventing attacks such as double spending
   4. Allow clients to query the UTXO database

Polkadot

What is Polkadot:
Polkadot is a scalable heterogeneous multi-chain.

- Polkadot provides **no inherent application functionality** at all.
- Polkadot should be considered **equivalent to a set of independent chains**.
- The foundation of Polkadot revolves around the the **relay-chain** in which a large number of **valiatable, globally-coherent dynamic data-structures may be hosted side-by-side.**
  - This is called parallelised chains or **parachains.**

"Polkadot provides a rather bare-bones piece of infrastructure leaving much of the complexity to be addressed at the middleware level"

The Philosophy of Polkadot:
The goal for Polkadot is to **provide a foundation to build the next wave of consensus systems** right through the risk spectrum from production-capable mature designs to nascent ideas. By providing security, isolation and communications, Polkadot is able to let **parachains select their own properties.**

Polkadot foresees:

1. Conservative, high-level chains (Ex. Bitcoin and Z-Cash) co-existing with lower-value **themechains** and test-nets with zero (or near zero) fees.
2. Fully-encrypted, "dark", consortium chains operating alongside—and even providing services to—highly functional and open chains such as those like Ethereum.
3. Experimental new VM-based chains such as a subjective time-charged wasm chain being used as a means of outsourcing difficult compute problems from a more mature Ethereum-like chain or a more restricted Bitcoin-like chain.

The main principles of Polkadot:

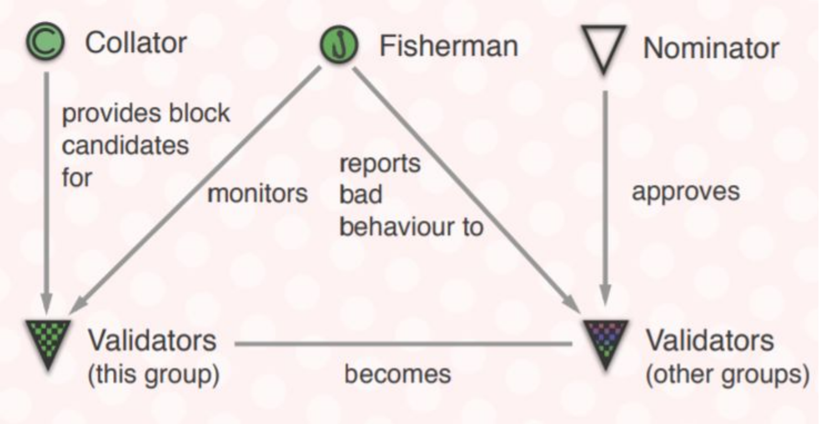
1. Minimal
   a. It should have as little functionality as possible.
2. Simple
   a. No additional complexity should be present in the base protocol than can reasonably be offloaded into middleware.
3. General
   a. No unnecessary requirements, constraints, or limitations should be placed on **parachains**.
   b. It should be a test bed for consensus system development which can be optimised through marketing the model.
4. Robust
   a. Should provide a fundamentally stable base-layer. This means decentralizing to minimize the vectors for high-reward attacks.

Polkadot is aiming very high and trying to cater to a variety of projects which in some way makes it lose its identity. It is unclear what Polkadot wants to achieve because the philosophy it provides is too broad. While it has its set core principles, Polkadot does not restrict itself so the vision the team has for the Polkadot can easily change throughout the developmental process.

<u>Participation in Polkadot</u>

Polkadot Network's Basic Roles:

1. Collator
2. Fisherman
3. Nominator
4. Validator



Validators:
Validators are the highest charge and **helps seal new blocks on the Polkadot network**. The role of validators is **contingent** upon (1) **a sufficiently high bond** being deposited (Proof-of-Stake) or (2) **nominations by one** or more validators **to act for them**.

Validators must **run a relay-chain** client implementation with **high availability and bandwidth.** The nodes are responsible for ratifying a new block on a nominated parachain by **receiving, validating, and republishing candidate blocks**. The nomination **is deterministic but is virtually unpredictable**.

Because validators **cannot maintain a fully-synchronized database of all parachains**, it is expected that the validators **will nominate the task of devising a suggested new parachain** block to a third-party (collator). When all new **parachain blocks** have been properly ratified, validators must then ratify the **relay-chain block** itself. This is done by **updating the state of the transaction queue, processing the transaction of the relay-chain transaction set and ratifying the final block.**

If a validator fails to complete the tasks stated, the validator will be punished.

1. For initial, unintentional failures
    a. THe network will withhold the validator;s reward.
2. Repeated failures
    a. The reduction of their security bond by burning their stake token.
3. Caught with malicious actions
    a. The loss of the entire bond.

Overall the role of a validator is very straightforward and very similar to validators of any proof-of-stake consensus protocol. The only difference is that the validators need to check every single paracha in which realistically seems impossible and could be exploited.

Nominators:
A stakeholding party who contributes to the security bond of a validator. Only role is to place risk capital and to signal their trust a particular validator. Based on the results of the validator the nominator contributes to, they will either increase or reduce their deposit.

While validators would be considered as the mining pools in Proof of Works, nominators would be considered the miners.

Collators

Parties that assist validators in producing valid parachina blocks. The purpose of collators is to maintain a **full-node** for a particular parachain. Under normal circumstances, collators will collate and execute transaction to create an unsealed block and using zero-knowledge proof, provide it to one or more validators responsible for proposing the block.

The relationship between collators, nominators, and validators is not solidify yet
This is because as the cost of maintaining a synced version of all such parachains increases, it will cause a separation between the validators and collators.

"Eventually, we expect to see collator pools who vie to collect the most transaction fees. Such collators may become contracted to serve particular validators over a period of time for an on-going share in the reward proceeds. Alternatively, "freelance" collators may simply create a market offering valid parachain blocks in return for a competitive share of the reward payable immediately. Similarly, decentralised nominator pools would allow multiple bonded participants to coordinate and share the duty of a validator. This ability to pool ensures open participation leading to a more decentralised system"

Simply, I do not know what I am supposed to expect from collators. While their purpose is to ensure validators can continue to work, their method of doing this is confusing and not concrete. There are many exploits and risk involved with using a collators which makes me worry of what the purpose and how to use this role. But if collators are not included, the Polkadot platform will likely not function at all.

Fisherman:
Fishererman are not directly related to the block-authoring process. They act as an independent "bounty hunter" motivated by a reward. Fisherman are rewarded through a timely proof that at least one bonde party acted illegally. To prevent this system from being exploited, the reward is minimal. The reward can be increased as more corroborating illegal signature from validators are provided.

Once again, the use of a fisherman in this architecture is confusing and in my eyes difficult to implement. The entire concept is very similar to any other blockchain protocol as it is just having nodes check each others but the way it is implemented is different and overall requires a large amount of logistics figured out.

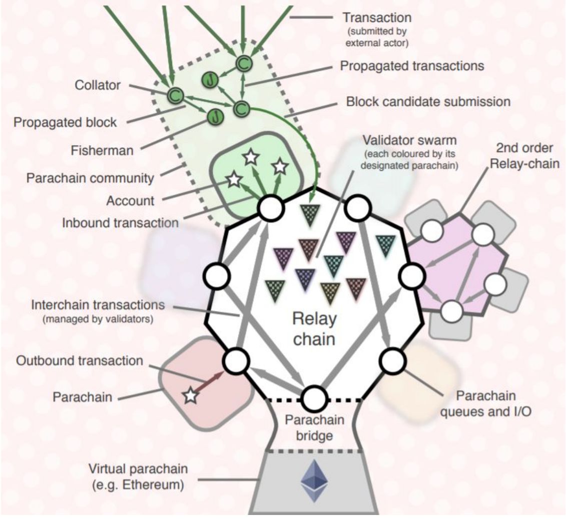
Design Overview (Architecture)

Consensus:

Proof of authority network- Provides an efficient and fault tolerant consensus over an arbitrarily defective network.

Using Proof of Stake, Polkadot is able to determine a set of validators and incentivise them to be honest.

The image to the right is a summary schematic of the POlkadot system where the collators are collecting and propagating user-transactions as well as propagating block candidates to fishermen to validators. Not only that but it shows how accounts will be posting transactions which will be taken out of the parachain and into the relay-chain to have them communicate.


Transaction
(submitted by external actor)
Propagated transactions
Block candidate submission
Collator
Propagated block
Fisherman
Parachain community
Account
Inbound transaction
Validator swarm
(each coloured by its designated parachain)
2nd order Relay-chain
C
J
Relay chain
Interchain transactions
(managed by validators)
Outbound transaction
Parachain
Parachain bridge
Parachain queues and I/O
Virtual parachain
(e.g. Ethereum)