

Shariah-compliant block production on proof-of-stake blockchains

Description:

In the evolving landscape of blockchain technology, there is a growing need for solutions that not only embrace technological advancements but also adhere to ethical and religious principles to ease the path for mass adoption of this new technology. Our invention addresses this need by introducing a Shariah-compliant block production mechanism on Proof-of-Stake (PoS) blockchains.

The primary objective of this invention is to develop a Shariah-compliant block production on Proof-of-Stake (PoS) blockchains. To ensure this, our system is designed to conduct a Validator Node in a manner that avoids any direct involvement in transactions that conflict with Shariah principles. This means specifically excluding transactions related to lending and borrowing, gambling, and any activities associated with prediction markets. Our goal is to provide a Validation Node structure that aligns with ethical and Shariah guidelines, offering a unique and shariah-compliant way for users to participate in PoS blockchains while ensuring compliance with the underlying consensus mechanisms of the blockchain.

Below is a primer to Proof-of-Stake Mechanism that is the bedrock for major PoS blockchains:

To ease the readers' understanding we will take the example of the Ethereum (ETH) blockchain to explain the workings of a Proof-of-Stake (PoS) blockchain.

Ethereum's Proof-of-Stake (PoS) mechanism uses validators which replace miners in the Proof-of-Work mechanism previously. They play a crucial role in processing transactions and creating new blocks in the blockchain.

To become a validator, one must:

1. Install & run:
 - a. an execution client: listens to new transactions broadcasted in the network, executes them in EVM, and holds the latest state and database of all current Ethereum data.

- b. a consensus client: implements the proof-of-stake consensus algorithm, which enables the network to achieve agreement based on validated data from the execution client.
- c. a validator client: part of consensus client allowing a node to participate in securing the network.
- d. Stake 32 ETH tokens as a form of security deposit.

Validator Actions:

There are two primary roles for a validator:

- 1) checking new blocks and 'attesting' to them if they are valid
- 2) proposing new blocks when selected at random from the total validator pool.

If the validator fails to do either of these tasks when asked, they miss out on an ether payout.

Validator Attestation:

A validator is expected to create, sign and broadcast an attestation during every epoch. An epoch has 32 slots, each slot being 12 seconds so $32 * 12 = 384$ seconds or 6.4 minutes.

The attestation is for a specific slot in the epoch. The purpose of the attestation is to vote in favor of the validator's view of the chain, in particular the most recent justified block and the first block in the current epoch (known as `source` and `target` checkpoints).

Block Production:

Validator accounts propose blocks and are occasionally responsible for proposing a block. A single validator is pseudo-randomly chosen to propose a block in each slot. Only one block proposer is selected in each slot.

The block proposer creates a block by collecting data from its own local database and view of the chain. The execution client has access to a local pool of transactions that it has heard about on its own gossip network. These transactions are executed locally to generate an updated state trie known as a post-state.

After the data is collected; in a beacon block, it's signed, and broadcast to the block proposer's peers, who propagate it on to their peers, etc..

Validator Rewards:

Validators are rewarded for submitting attestations.

The attestation reward is dependent on two variables, the `base reward` and the `inclusion delay`. The best case for the inclusion delay is to be equal to 1.

`attestation reward = 7/8 x base reward x (1/inclusion delay)`

The base reward is calculated according to the number of attesting validators and their effective staked ether balances:

`base reward = validator effective balance x 2^6 / SQRT(Effective balance of all active validators)`

The execution client, integral to this system, not only listens to transactions but also performs complex computations to maintain the integrity and state of the blockchain. Similarly, the consensus client plays a pivotal role in ensuring network security and agreement on the blockchain's current state.

Ethereum PoS works differently from delegated PoS (dPOS) blockchains where users delegate their tokens to increase the share of validators. However the same principle can be applied on dPOS blockchains to make validator's block production sharia-compliant.

The Invention: Shariah-Compliant Block Production for Proof-of-Stake Blockchains

This invention presents a novel approach to blockchain validation in Proof-of-Stake (PoS) networks, specifically designed to comply with Shariah principles. At its core, the invention redefines the block production process by incorporating ethical filters that ensure all transactions in a block adhere to the Shariah. The mechanism consists of the following key components:

Transaction Selection from Memory Pool: Each validator, when chosen randomly as a block proposer, selects transactions from the Memory Pool (mempool). These transactions are then verified for authenticity using a local execution client and a consensus client. The unique aspect here is the method of transaction selection, governed by Shariah-compliant criteria.

Allow-List Approach: The primary feature of this invention is the use of an allow-list. This list is meticulously maintained to include all publicly identifiable transactions that meet Shariah compliance standards. The inclusion criteria are broad yet specific, covering:

- ERC20 token transfers that are non-algorithmic, non-derivative, and non-synthetic.
- All native Ethereum (ETH) token transfers.
- Transactions involving Decentralized Exchanges (DEXs).

This list is not exhaustive but indicative of the types of transactions deemed permissible.

AI and Machine Learning Integration: To enhance the accuracy and adaptability of the validation process, the system integrates advanced AI and machine learning algorithms. These algorithms are trained to continually evaluate and categorize transactions based on Shariah compliance, adapting to evolving interpretations and transaction patterns.

Blacklist Approach: Complementing the allow-list, the invention also employs a blacklist to explicitly exclude non-Shariah-compliant transactions. This blacklist approach acts as a secondary filter, ensuring the utmost adherence to Islamic financial principles.

Modified Blockchain Client: The heart of the invention lies in the modified blockchain client, which is programmed to process transactions using the allow-list and blacklist approaches. This client is responsible for ensuring that every transaction in the block adheres to the defined Shariah-compliant criteria.

This innovative approach to block production on PoS blockchains not only aligns with the core principles of Islamic finance but also offers a dynamic, adaptable system capable of evolving with the interpretations of Shariah compliance. By combining ethical considerations with advanced technological mechanisms, this invention paves the way for broader adoption of blockchain technology in markets that prioritize Shariah compliance.

Claims:

Claim 1: A method for producing Shariah-compliant blocks on a Proof-of-Stake (PoS) blockchain, the method comprising: identifying and processing transactions based on Shariah compliance through any of the following mechanisms: an allow-list, a blacklist, or artificial intelligence (AI) algorithms; wherein the method ensures that each transaction included in a block adheres to predefined Shariah principles.

Claim 2: The method of Claim 1, wherein the Shariah-compliant transactions are identified using an allow-list approach, the allow-list approach comprising including

transactions in a block based on a predefined list of approved transaction types that comply with Shariah principles.

Claim 3: The method of Claim 1, wherein the Shariah-compliant transactions are identified using a blacklist approach, the blacklist approach comprising excluding transactions from a block based on a predefined list of prohibited transaction types that are non-compliant with Shariah principles.

Claim 4: The method of Claim 1, wherein the Shariah-compliant transactions are identified using artificial intelligence (AI) algorithms trained to categorize transactions based on their compliance with Shariah principles.

Claim 5: The method of Claim 1, wherein the allow-list and blacklist approaches are utilized concurrently to identify Shariah-compliant transactions, providing a dual-layer verification process for each transaction.

Claim 6: The method of Claim 4, wherein the AI algorithms are continuously updated based on transaction patterns and changes in Shariah compliance criteria to dynamically adapt to evolving interpretations of Shariah compliance in blockchain transactions.

Claim 7: A system for block production on a PoS blockchain, the system configured to execute a modified blockchain client that filters transactions based on the method of Claim 1, ensuring the inclusion of only Shariah-compliant transactions in the blocks.

Claim 8: The system of Claim 7, wherein the blockchain client collaborates with an execution layer to process transactions using the allow-list, blacklist, or AI methodologies for Shariah compliance verification.

Claim 9: The system of Claim 7, wherein validators participating in block production are rewarded according to a Shariah-compliant framework, with rewards determined by the validators' adherence to the methods outlined in Claims 2, 3, and 4.