DotOracle
DotOracle Network

Decentralized Oracle and Cross-chain liquidity network for Polkadot Ecosystem

Context

* Ethereum has been becoming the de-facto standard for liquidity mining and decentralized trading.
* The rise of Uniswap and AMM models and the demand for trading on DEXes leads to the increase of transaction gas cost on Ethereum, making transaction fees unacceptable for normal users.
* The rise of Polkadot system with interoperable parachains that decentralized inter-communication between parachains possible, making Polkadot become a fast evolving ecosystem.
* MoonBeam has been rising as the best parachain that provides an EVM-compatible platform in the Polkadot Ecosystem.

DotOracle Proposal

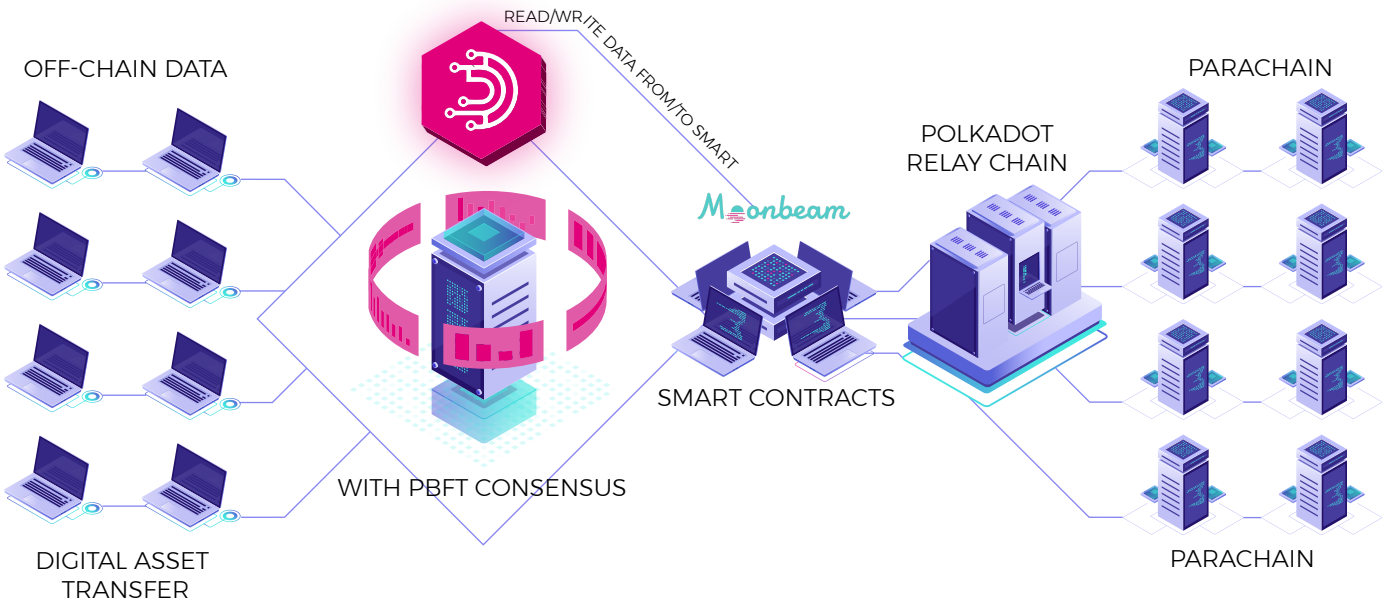
DotOracle proposes a decentralized network that allows:

* To connect the real world to the Polkadot ecosystem by providing a decentralized oracle service to transfer information from the real world to Polkadot in real time.
* To create a liquidity network layer that transfers digital assets back-and-forth between Polkadot and other smart contract platforms, especially Ethereum.

Our Vision: Adoption of cryptocurrency

* The decentralized oracle of DotOracle aims to blur the boundary between the real world and the information within blockchains. This will be achieved by our fast decentralized oracle.
* DotOracle will also act as a decentralized liquidity bridge network that allows to quickly transfer back-and-forth digital assets between Ethereum and Polkadot Ecosystem. In the current state-of-the-art, Ethereum has the majority of liquidity, despite having many problems such as scalability and transaction fees. DotOracle's vision thus aims to balance the liquidity between Ethereum and Polkadot, allowing users with small digital assets to access liquidity with less fees than on Ethereum.

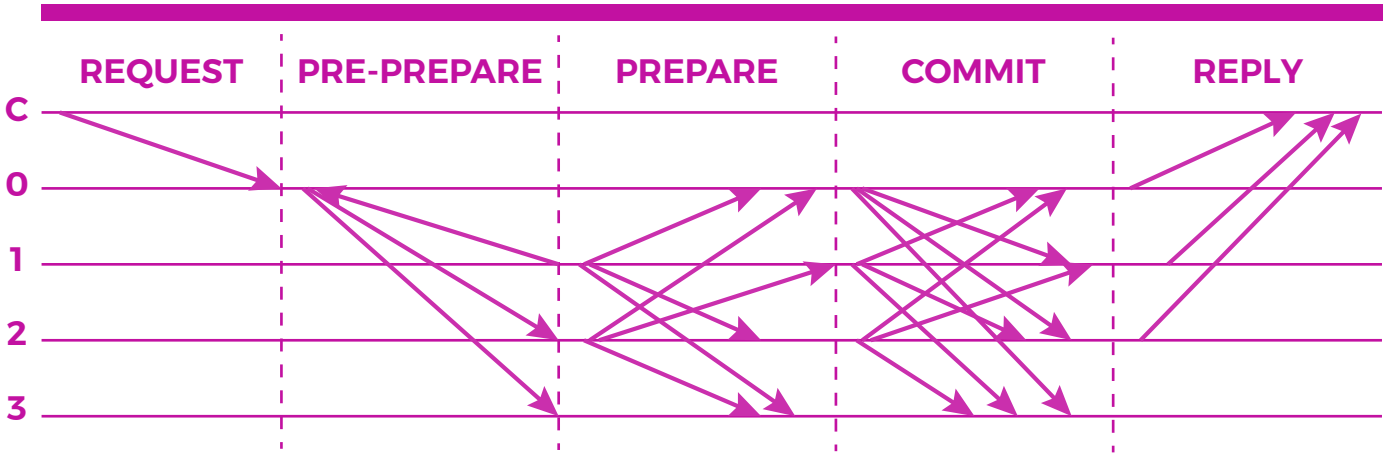
Technical solution



DotOracle network is a chain-less decentralized network. To achieve fast confirmation of the network, we propose to use the Practical Byzantine Fault Tolerance (PBFT) distributed consensus protocol combined with the Elliptic Curve (EC) Multisignature scheme. The use of the latter is to reduce the communication complexity of the original PBFT protocol.

DotOracle connects the real world and other blockchains to the MoonBeam parachain of Polkadot. The reason we choose MoonBeam is because it provides an EVM-compatible platform on a Polkadot parachain. The EVM makes it easy to integrate with other smart contract platforms. Moreover, as MoonBeam is a parachain of Polkadot, all information/digital assets on MoonBeam will be easily transferred to other parachains in the Polkadot Ecosystem.

To ease the description of how PBFT works with the EC multisignature scheme, we utilize the following figure extracted from Tendermint PBFT.



Let's demonstrate an example when there is a user request transferring USDT from Ethereum network to MoonBeam. Here's how the DotOracle works when the user request is relayed to the nodes of DotOrale:

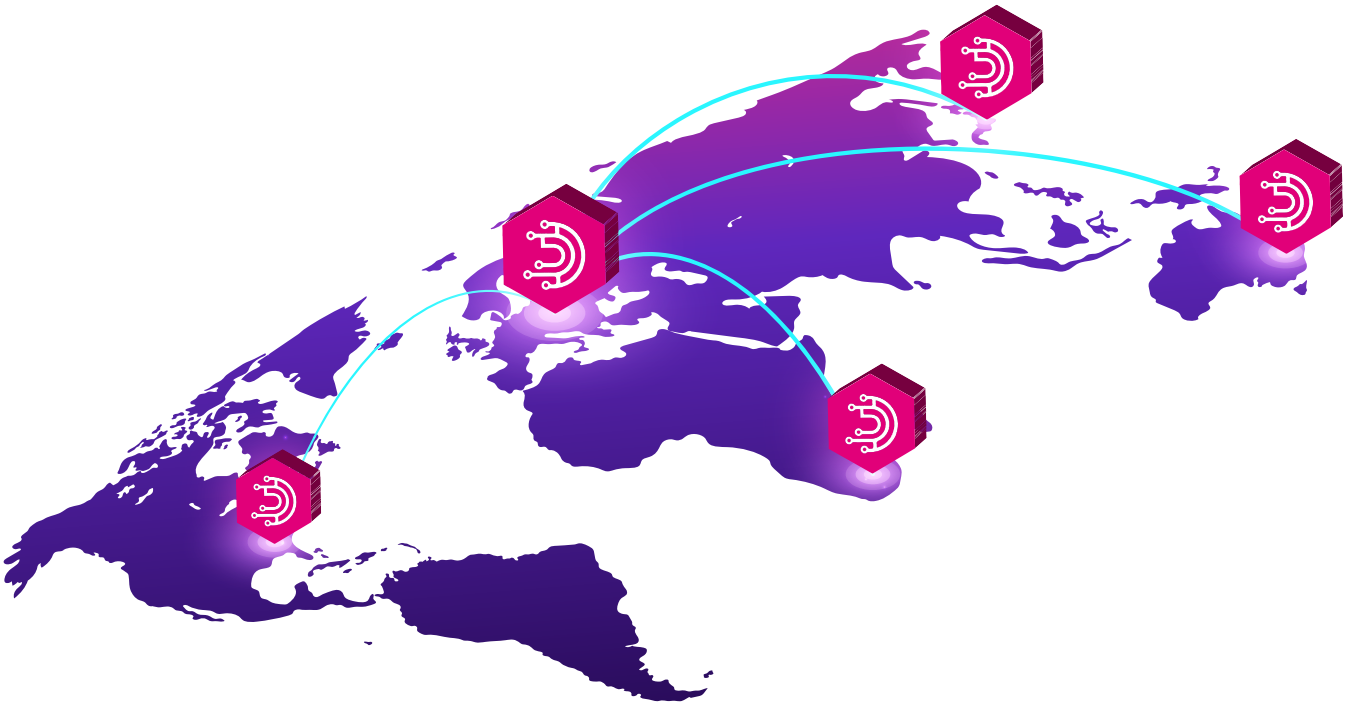
Step 1 - Pre-prepare: One of the DotOracle validators is selected as the leader of the network. Leader is changed for every request in a round-robin manner.

Step 2 - Prepare: All validators check the validity of the USDT transfer transaction on Ethereum. USDT should be transferred to and locked by the DotOracle smart contract on Ethereum. This smart contract is governed by the whole DotOracle network of validators, ensuring the decentralization and censorship resistance of the network. The leader validator then collects the validation results from at least ⅔ validators of the network. This is to guarantee that the transfer request must be validated by at least ⅔ validators of the network and the leader cannot manipulate the request. The leader then builds the EC multisignature which proves which validator has verified the transfer transaction.

Step 3 - Commit: The previously built multisignature is then broadcast to all validators, which then perform a second signature to confirm the previously multisignature.

Step 4 - Finalize: The leader relays the user request and the multisignature to the smart contracts on MoonBeam, which verifies the validity of the multisignature and issues the corresponding amount of USDT on MoonBeam if the multisignature is valid.

Bonding



Security is always the most important factor of any financial system. In order to ensure the security of the network, a bonded system is a must.

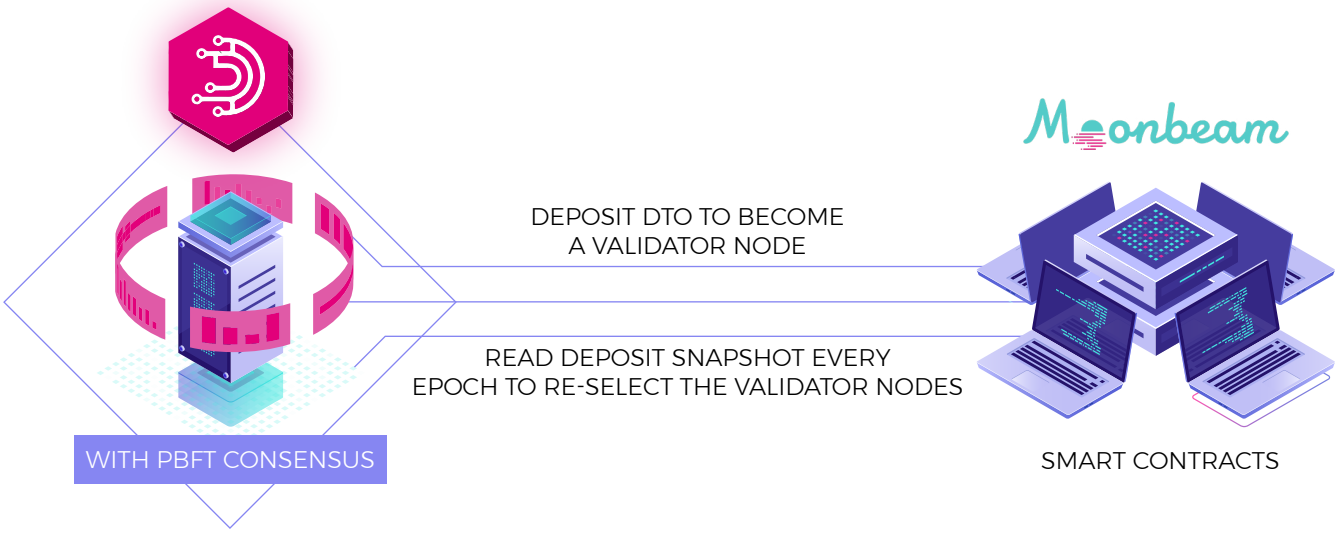
DotOracle will issue the DTO utility token.

Each DotOracle validator must lock/collateralize an amount of DTO in our smart contracts on Ethereum and MoonBeam.

To ensure that the network is trustless and that all validators follow the protocol of the network, a slashing mechanism will be implemented so that if a validator does not follow the protocol, the validator will be slashed and its bonded amount will be burned.

As in other networks, validators are rewarded in DTO for providing the service to DotOracle.

Epoch and network validator nodes selection



The network consists of a maximum of 49 validator nodes that follow the PBFT protocol to achieve the consensus whenever there are requests to the network. On the one hand, the number 9 is selected because it keeps the decentralization level of the network. On the other hand, 49 validator nodes combined with the consensus as described above allow the network to quickly decide the consensus, thus being capable of providing the network service in real-time.

Furthermore, we envision that each validator node will be operated by a financial organization such as an institution, a capital venture, or a DAO-based organization. By being operated by reputable financial organizations, the network will be more stable in responding to user requests.

Any one can operate a node and make it become a validator node by depositing a minimum of 200,000 DTO to DotOracle smart contracts on MoonBeam. If there are more than 49 nodes, top 49 most deposit nodes will be selected as validator nodes of the network.

The set of validator nodes is re-selected every epoch, which is equal to 3 days. This 3-day is called an epoch. This duration is selected as it ensures that the network validator nodes set changes based on the deposits of the nodes, while does not cause the network validator nodes set changes too frequently, thus making the network more stable.

At the beginning of every epoch, all nodes will read the state of the smart contracts for the deposits of all nodes. This process is to ensure that the selection of validator nodes is done by all nodes in a decentralized way.

Local and global consensus

Global consensus in DotOracle is an agreement of at least 2/3 of validator nodes of the network for user requests.

Local consensus in DotOracle is an agreement of a small subset of validator nodes of the network.

The DotOracle network serves for two purposes:

- Connecting off-chain data to the Polkadot Ecosystem
- Bridging liquidity and digital assets from different blockchains to the Polkadot ecosystem through the MoonBeam parachain.

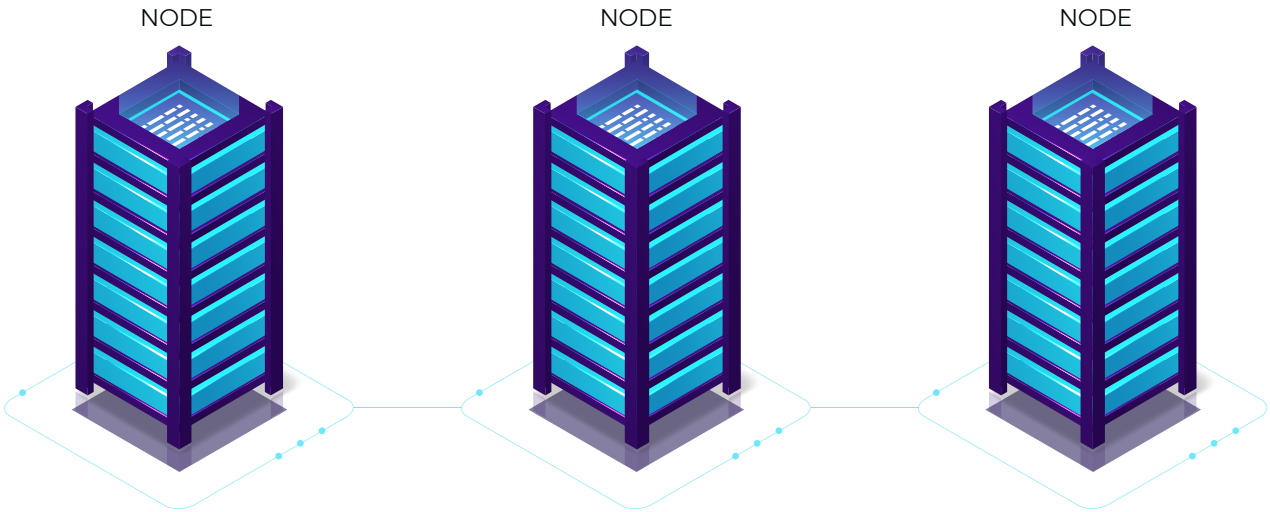
Global consensus is required whenever there is a user request for transferring liquidity/digital assets back-and-forth between MoonBeam and other blockchains such as Ethereum.

Local consensus is applied for off-chain data provision.

The reason global consensus is applied for liquidity bridge is because there should not be two different copies of the same digital asset on DotOracle. This means any application that uses liquidity bridged by DotOracle can be guaranteed that there is only one single version of USDT on Ethereum being transferred from Ethereum to Polkadot.

Any user application running on MoonBeam can use local consensus for the nodes to provide off-chain data to the network. To use local consensus, any application builder can select a subset of the validator nodes as candidates to read off-chain data and push to the DotOracle contracts. The subset of the validator nodes in this case will form a local consensus whenever there is a user request for off-chain data provision.

Slashing



Slashing is applied to a validator node if:

- The node does not participate in the global consensus 10 user requests consecutively.
- The node maliciously makes a decision against the agreement made by 2/3 validator nodes of the network.

In case of being slashed, the entire deposit of the node will be burned. This strong penalty slashing mechanism will incentivize every node to follow the protocol in order to be rewarded in DTOs by providing proper service to users.