TODA PROTOCOL - A LEDGERLESS BLOCKCHAIN

TODA FOUNDATION

1. Introduction

There is currently a need for a strong governance value exchange platform that scales globally to millions of transactions per second, powering billions of devices while increasing security and minimizing transaction fees. The TODA protocol is designed to meet those needs, is fully distributed, has a governance model which becomes more and more decentralized as it grows, and is optimized for mobile devices without the need for any centrally controlled system. It can also run on combination of nano-micro cloud instances, one representing each mobile device.

2. Design overview

In a fully distributed and decentralized system, nodes will need a map to efficiently navigate the network and locate objects. A virtual binary tree, we call it the Todatree, specifically a BStree structure is a virtual structure on the protocol level that contains every object in the entire system that exist, will exist or even objects that move from one place in the tree to another, they move using unique IDs from the tree as their identifiers and occupy points in the tree that are also unique. We use it because it is extremely fast to compute at each nodes level to locate a certain leaf's parent in a certain branch etc. and it is easy to any computer scientist to comprehend it and imagine its navigation and the possibilities of things that can be done in a deterministic way.

The smallest Unit is at the leaf?s level we call it TODAQ or Quark, and the smallest unit that can be transferred by users we call it TODAcoin, TODAUnit or Atom. It has an atomic structure in a sense that it has a unique number through its life and quarks actually form it within that tree structure, each quark has a unique number and can contribute along with a total of 2^{32} Quarks to form one single unique coin.

One application of TODA protocol a value to be exchanges similar to a currency, and the provenance of each unit/coin is from a sub-branch at level 96 the Provenance Wallet - A Coin can travel within the Todatree and have a new location every block only when transacted. Every movement builds a chain of Wallets.

3. Atomic Coins

- Each unit/coin contains its own history as a wallet chain of past transactions. If UserA sends UserB coin 115, the coin would look to the user like this:
  `C115_W(UserA).BlockNumber.MerkleProof_W(UserB).BlockNumber.MerkleProof`

- TODACoin is the smallest unit that can be transacted by users
- TODAQ or Quark is the smallest unit non-divisible, it is mainly used for exchange rate of 0.03% for the payer cost, 0.03% for the payee and 0.04% of new quarks that are issued every transaction that can be sent to the transaction wallet of users who win the PoAW. We are suggesting 2^{32} quarks = one TODACOIN.
- The Quarks are actually the leafs of the branch TODACoin in todatree*
- Quarks are managed within one coin until filled up, so the management of that is basically an increment to the coin extension. Say for example UserA owns 99.9% fractions of C113 that were gained by validating other coins in transactions, it would be represented by C113.4290672329_UserA, then if UserA validates another coin being transacted and wins the transaction fee as per PoAW then user A Qualifies to collect 0.1% in transaction fee on that coin which is 4294968 Quarks at that point the C113 is filled up and now can be spent by UserA. Similarly if instead UserA sends 100 coins to UserB, UserA will have to pay the 0.03% or Quarks which will reduce C113 from 4390672329 to 4161823310. UserA will then have C113.4161823310. Another implementation would be using a hashcash of next block?s merkleroot concat of existing transaction hash to charge 0.03% of the times 1 coin at a time.
- Basically, a coin will have to be transacted 2500 times so its impact aggregate is creating 1 new coin

4. Todatree

- Like a BST of height 256. The leftmost branch of level 160 can be viewed as a tree itself, and transactions within it can be processed more efficiently than in the full tree, but we keep 256 by design for ease of future expansion.
- This tree is used as a map so nodes know how to navigate the system and where to expect objects to be.
- We call this tree the Todatree. We refer to level 96 as t96 and it is the level that represents the wallets.
- At the very beginning, the very first wallet on the left, has the provenance of all the coins
- Quarks with the zero denomination are at the leafs level t0 of that wallet, higher denominations would be a branch, for example if we want 1024 quarks, we get 1t11 and a coin is 1t32 or 2^{32} quarks. Since it?s a fully saturated Balanced BST the quarks have unique numbers at the provenance level and at the locations that they can occupy in other branches that are symmetric to their provenance, each wallet keeps drawing from exact same branch (coin) when it wins transaction fees until fully completed before it moved to the next one. The reason for that is that 2^{32} quarks must be filled up to constitute one specific coin before it can be transferred from transaction wallet to actual wallet to then be saved or consumed.
- Coins have a unique number for the life of them. Their number is that of the leaf they come from, and their movements are leafs they occupy belonging to other

wallets, hence wallet chain is within each and every coin is calculated at the system level as it is better from usability perspective to show wallet chain than point occupied chain.

- Every block, the entire provenance W branch move to the right of the first wallet at t64 which is $W(2^63)$ (basically after the first block would, it would be $W(2^63 + 2^64)$) The reason for that is so when looking at any coin number you can tell its provenance block by conducting a quick binary computation, so the provenance W of a coin is linearly related to the block# so the calculation would be: $Provenance\ wallet = (W(2^63 + (block\# * (2^64)))$

5. Merkle Tree

- Generated every T seconds
- The root hash is generated by every active validating node in the network shared across the network
- Every coin is coupled with another coin to potentially form L1 of new merkle tree (Merkle tree and Todatree are not the same, merkle tree is another dimension that gets created every block, while todatree is static for the life of the platform) utilizing the previous Merkle root in a function to pseudorandomly select a range for the coupling of coins along with the nodes/managers of those coins
- Only managers with at least one coin that is being transacted in the current block can be going up to L2 and all the way up to

6. Merkle Root

- The non-transacting coin manager will be relying on the transacting coins managers to get back with the new merkle proof If a manager (every wallet is a manager to some other coins) does not have any coins transacting and therefore not incentivized nor authorized/expected to participate in building the Merkle tree, it will need to rely on the other nodes to relay what was the last Merkle root. The same applies to dormant or disconnected nodes, when they rejoin the network.
- Every participating wallet must have all Merkle roots in the system along with last coin assigned and last wallet assigned during each Merkle root

7. Beacon

- The Merkle root of last block is used as a beacon in existing block to generate pseudo-randomness that is deterministic and universal.
- For example, the Merkle root is used in a function to tell each wallet which coins to manage during the existing block
- To make the deployment secure consider each wallet to manage on average 32x the average amount of coins owned by wallets
- Coin does not change ownership on its own, there?s a minimum of 32 coins transferring together

8. Wallet/Node/Manager/Miner

- While wallets have unique numbers in the Todatree, at the t64 level they can also be part of a branch at a higher level than t64
- The wallet owning the coin is the only one able to crypto-sign it to change its ownership

9. Coins

- Each coin has a unique number at t0 level in Todatree that is atomic, so it is not divisible
- Each coin is a file, as coin changes ownership, it keeps a stamp of the location it occupied in its own file, so anyone can compute the wallet chain it was owned by

10. Decentralized and Distributed Consensus Mechanism

- For every coin, at least 17 managers must testify its transfer for it to be transacted and those managers must be the ones selected by the pseudorandom function so everyone knows who they should be during a certain block
- Testifying includes the check for authenticity of coin, ownership, cryptosignature, then racing with that coin up the Merkle tree for the PoAW

11. Economic Incentive / PoAW (Proof of Actual Work)

- The first validator node / manager for every coin to get the correct Merkle root will collect the transaction fee of 0.1%
- Tx fees are split between payer 0.03%, payee 0.03% and new issuance of coins 0.04%
- Tx fee charged to the payer and payee is taken off their Wallet in Quarks when in fractions.

12. Todatree structure

- T64 level is divided into 4 equal sections that are all on the same level
- The ones on the far left are for wallets with coin provenance but the ones on the far right of t64 are for wallets assignments

13. Wallet reputation

- Every wallet owner is automatically assigned a transaction wallet that can only receive coins(quarks) based on transaction fees earned due to PoAW (Proof of Actual work)
- Transaction wallets fill up quarks/coins in its branch of the Todatree from left to right. So verifying the one on the very far right you can deduce the reputation as you must leave Quarks in Tx Wallet at all times.

- Wallet reputation along with other impactful elements on Proof of Stake, like how many coins are owned by a wallet and other elements, how long have they owned the coins, the diverse provenance of their coins and the frequency and disparity of last transactions are some aspects that can play important roles in impacting the role one wallet can play to participate in PoAW
- Even when quarks add-up to become coins so they can get transferred to regular wallet and get consumed, the far right last (at least one quark)must stay in transaction wallet more like needle / acts as a reputation-meter.
- The combination of reputation, number of coins owned, and active in sending and receiving build up a score that increases the expectation of a node to validate more coins that are being transacted and participate in the potential gain of the 0.1% of transaction fee. The less the score, the less likely the other nodes would couple with while racing towards the merkle root and therefore the less likely to have the ability to game the system.
- The math can work in a way that inorder to game the system or initiate a successful sybil attack, you must have full control over more than 50% of the active/issued coins in the system and operate more than 50% of active Wallets.

14. Transaction stamp / Double Spending

- Payer and payee must cryptographically sign inorder for Units to be transacted
- Up to 32 from the pseudorandomly selected Nodes/Wallets by a function of the last Merkleroot so it?s deterministic and everyone knows who they should be for any active coin for the duration of a certain block. If a coin is transacted, they will all be replaced by a new set of 32
- The first of the 32 selected nodes that comes back with the correct merkle proof (including new merkle root) will get the 0.1% transaction fee - We call this PoAW
- TODAQ or Quarks, are managed in a value as an extension to the coin parent. Given that users can not send Quarks on their own but as part of transaction fee they can only receive Quarks or their Quarks gets subtracted when they issue a transaction.
- Although coin authenticity can be checked up instantaneously, in-order for coin receiver to ensure every coin received was not double spent, it must wait for the block to conclude and stamp transaction with Merkle root and retain Merkle proof for use in subsequent transaction. At the end of the block, the coin transacted will have new managers that are pseudo-randomly chosen by a function of the merkle root, those new managers must certify ownership to only one owner and therefore anyone in the system including new owner can easily check to see if coins are assigned to them and no one else.