denet

Ver. 5

DFILE Digital Paper

Denis Shelestov | Rafik Singatullin

Aug 2021

Disclaimer

This document is part of a big digital documentation, which in turn will be supplemented over time and updates on the project. Everything that is described in the document is a concept and does not guarantee 100% work under conditions other than those described in the document. As long as this disclaimer exists, the status of the document is considered “Under construction” and may be edited or changed in the process.

Abstract

DeNet.File.Token — DFILE — decentralized storage staking token, made for support grows up the storage market for the next 10 years by DeNet team. In this paper we give technical information, that the token is used for storage, consensys, farming and staking.

DeNet Team work with storage ecosystems based on free capacity since 2017

Introduction

If you read the paper carefully, you will understand some reasons why we choose EVM networks, instead of building own blockchain.

Proof Of Storage contract help nodes and customers got consensys and automated scalability work without 100% uptime from customers. But node, need to be online always.

Also in this paper you will find technical part about VDF, Node and Token

New blockchain vs EVM

Decentralized storage will works with DFILE in EVM blockchains, start’s from Ethereum Kovan testnet and may scales into Ethereum Mainnet, Binance Smart Chain, Polygon Network, PoA Network, Ethereum Classic and other EVM supported blockchains.

Reasons for exclusion new blockchain development

A lot of time for developing own blockchain.
If we look at EVM blockchains we will see that the development of the blockchain is still ongoing by many hardcore teams, and there are still vulnerabilities and problems with scaling

Blockchain attacks.
Any new blockchain has next problems - high risk with double spent or 51% attacks.

Higher requirements for all miners.
If we look at FileCoin, miners need 128GB RAM to start a basic node and a lot of disk space for blockchain support.

Lack of liquidity if you are a new product.

Needs create own bridges to liquidity blockchains. Centralization or consensys problems.

Reasons for using EVM

DEX and Liquidity is ready. Any project can fast grow up in network, where users have a wallet with liquidity tokens.

A lot of token's is ready.

DAPP's is ready.

Community is ready.

Constant development and renewal of networks.

The ability to easily configure bridges between different EVM networks.
If chosen network have a lot of network fee (for example Ethereum in Mar 2021 had 1000 gwei gas price), miners may change network for one touch.

Smart Contract Support.

Token Standards Support.
ERC20 and other popular token’s.

Wallet availability.
For any EVM network, customers can use any wallet that work’s with any EVM network.

Testnet availability.
Any miner, customer and developer may use any EVM testnet. And it will work

PoS

Proof of Storage - PoS or PoStorage in next articles - consensus algorithm for storage nodes, who storing customers information.

Problem

Need to made consensus algorithm that works next conditionals

- Minimum system requirements
- Minimum blockchain using
- Infinity-like scalability
- Decentralized
- ZeroKnowlage Proofs (no sending full file for proof)
- Proving file storage without customer-side
- Proving file storage without smart-contract knows about file

Solution

Description of POS. At this moment we have 2 versions of realization PoS

- With node NFT Token (Working now in Kovan Testnet)
- With VDF (In progress/dev)

In this paper we will focus on the first version of PoS.

Definition

- **User** — customer — data uploader
- **FS** — fs — User File System
- **RH** — Root Hash from merkle tree
- **8kb** — part of file
- **Nonce** — User Nonce for fsRH from uploader
- **Digsig** — Digital Signature from user or node
- **PoS** — Proof Of Storage
- **SC** — Smart Contract
- **VDF** — Verifiable Delay Function
- **VDF_PROOF** — Proof of Verifiable Delay Function

Proof

$$proof = f\,(MerkleTree8kb, RHFS, nonce, digsig)$$

Realization with Node NFT Token.

That proof can be approved by smart contract or any user without any other data.

Conditionals to send proof

Node need to check next conditionals

- Is it profitable
- Is it possible now

Profitable.
As your know, ProofOfStorage works with any EVM network, and need's to send transaction from time to time. A lot of EVM networks have fees for creating transactions. In part PoS Benchmark you can get average information about not optimized Gas using proof, without VDF.

Proof of Storage supports any ERC20 like tokens for payments per proof.

Basic conditionals to make transaction:

- $Reward_{ex} > TX_{cost}$ & $Reward_{rest} > Reward_{min}$
- $TX_{cost} = Gas_{used} \times Gas_{price}$
- $Reward_{ex} = f_{swap}(T_{reward}, Gas_{token}, TX_{cost})$
- f_{swap} — requires Token A, Token B and Min returns.
- *Reward* — Amounts of returns T reward per proof
- Gas_{token} = Token or Coin for tx payments in chosen network, for example in ETH it will ETH.

- Reward_{min} — *Minimal earnings for one proof TX in miner config*
- Reward_{ex} — Amount of Gas token, after swap token from reward
- T_{reward} — Token ERC20 *(as example)*
- Reward_{rest} — difference from amount of requirements Treward in swap and total amount of T_{reward}

Possible to send proof

- $f_{balanceOf}(Node_{address}, Gas_{token}) > Gas_{used} \times Gas_{price}$
- $f_{depositOf}(Node_{address}, T_{reward}) > Reward \times T_{mindeposit}$
- $Nonce_{proof} >= Nonce_{last}$
- $f_{sha256}(proof, node_{address}, block_{nubmer})\% \; base_{difficulty} < user_{difficulty}$

Besides economical conditionals, node need to have actually merkletree of FSuser and latest nonce with user digital signature. Also node need valid proof.

$T_{mindeposit}$ — some number from 1 to 300

Create proof

Verify

Verifier can be node or smart contract on EVM, need to host last nonce for approve digsig and RHFS.

Refund

Full refund per using DFILE token for PoS in progress / dev.

Benchmarks

DFILE launch ProofOfStorage in Kovan testnet and got next results with first version of PoS.

Type	TxHash in Kovan Network	Gas Used
Send Proof	0x78cf94658f9eee1ebbb181f75235f04 8585deb1310d24861349ebe617f23fa06	303,898
Make Deposit	0xeaae8d517a9dab2796d0ba46c7477 8617f51eb3f055501237dd9415168c2bf61	85,135
Close Deposit	0xed01aea41f32f280d7ee46a1fa8ae452 9d4a886997dbd443c37228d8f6a20c2d	38,323

VDF

Staking

Solution

Node using ProofOfStorage for verify stored data. Any user can create own node application for work with POS.

Node NFT

Current have only concept model for creating defend option for PoS. and may disabled by VDF or other math algorithm.

NFT minting everytime, when node connect to network.

Basic parameters

- Owner Address
- Total Stored data size
- Total Proved data size
- Total Earned
- Deposited balance
- Network Mask
- IP Address
- Amount of proofs

Deposited Balance need’s to calc max node earn amount per proof.

Deposit balance = dep_balance

max earn per proof = 0.003 × dep_balance

Crypto-Security

FDS

FDS - Full Digital Signatures using