DeFile - A Decentralized File Storage System (Rev. 1)

Jonas Korene Novak
info@dcrubro.com

August 2024

Abstract

A decentralized file storage system would allow users to freely store any data, without the need to rely on service providers requiring trust. Users could store any data, safely, without anyone else being able to read or modify them, as they would be protected with the user's keys. The protocol would utilize storage providers to give up their excess storage space, for users to use and pay fees to store that data using the network's native payment token. This token is freely transferrable between users, and new tokens are constantly minted to nodes to keep incentive to store data, while adjusting the mint amount as the network grows. Distributing the data via a Peer-to-Peer network would practically ensure that no data is lost accidentally, while distributing the chain and trimming out unused data to save storage space. The protocol utilized a Proof-of-Validity consensus system, which ensures that data is only being modified by the user who originally submitted the data. Network nodes are free to leave and rejoin the network, taking the most common chain as the valid one in order to continue their work storing user data.

Disclaimer

In this article, the DeFile team is not trying to sell or promote any product. It merely exist to provide the basic idea of how the product could function, and to receive feedback on it.

This article does not define this final form and function of the presented product. It is merely a basic description of the basic idea of the product. This is subject to change.

Preface

Before reading this article, we highly recommend familiarizing yourself with the general technology behind blockchains and cryptocurrencies in general, as this article derives greatly from those. We recommend reading the articles in the References section.

1 Introduction

Data storage on the internet has become too dependant on cloud-based services. Although this works for most people, this introduces many issues or trust between the service provider and the customer. These issues of trust include the issue of data privacy, service price increases, accidental data loss by said providers, etc. Although many systems and policies are in place to mitigate these issues, such as GDPR, data backups and more, the underlying issues still persist. Most of these problems can be avoided by users buying their own storage solutions, such as NAS units, extra storage drives and more, these solutions can quickly become extremely costly to the end user and are not very scalable.

What is needed is a secure system, requiring no trust, with data privacy and no data loss, without the need to spend high amounts of money to store that data. We propose a system utilising Satoshi Nakamoto's blockchain system to connect service users and storage providers to securely store data and reward the storage providers, without the fear of having your data lost or stolen, and without the need to pay high amounts to store that data.

2 Data Storage

Data storage via the DeFile protocol (DEP) uses the ability to write any string of data into a block. The data is written directly into the contents of the block, which is then hashed and pushed to the end of the blockchain like normal.

However the problem with the current blockchains is that they're by definition immutable. This feature is not suitable for our use, and has led to many downfalls of projects using blockchains, be it the Ethereum or the Solana blockchain as data storage. This level of immutability is achieved by including the whole block as the input for a hashing function, which makes any change to the block's data change it's hash, and break the chain. By removing the block's stored data as an input for the hashing function, we make the block's stored data mutable, while leaving the rest of the block immutable. This change also makes the protocol faster, as the data is arguably the most size-heavy part of the block, which removes a lot of data from needing to be hashed.

Over time, as the size of the chain grows, using linear or binary searching of data when requested may become inefficient, resulting in slow loading speeds for the user. We can overcome this issue by storing the block headers, which among other things include the block owner (user) in memory. With the availability of memory today, storing a couple bytes of data in memory constantly should not be a problem, even at major blockchain lengths.

3 Data Distribution

Like all blockchains, the chain is distributed to all nodes connected to the network. This alone would suffice for distributing data to many different nodes, which ensures that even if a couple nodes go offline, the user's data is still safely stored on other nodes. However this alone would not allow for major scaling, as the size of the blockchain would become too big for any single node to handle efficiently.

We propose a different solution where each nodes handles a specific part of the chain. This way if we run 100 nodes on the network, we can distribute parts of the chain evenly between the nodes, while still having enough backups around. When a new node joins the network, it will start pulling any available blocks, until the amount of data that the node can store, is spent (other networks like the Bitcoin network use a version of this method called "Pruning").

We can also implement a method, where new nodes call on existing nodes, to get the least stored blocks, and store those to preserve enough backups of the data, however that could impact scalability.

If required, we can then reconstruct the full chain from these nodes.

4 Space Reclamation

Our current system poses a major flaw. Since the data is being stored in the chain, over time, the blockchain's size would grow to a very big size, which even with chain distribution would pose major problems.

Our proposal is to introduce a data trimming/nulling system, where every block has a set expiry date. When this date is reached, if the user has not made a request to extend the

block expiry date further, an aggrement is made between all nodes with the said block, to trim/null out the block's data, effectively deleting it from the chain. This trimming would also not require as much of the block header to be stored in memory, further saving memory space. This way we can control the rapid expansion of the chain, and remove unneeded data, that is not serving a purpose.

5 Incentive

In order to give the nodes hosting all of the data an incentive to store data, we can create a native token such as Ethereum's Ether token and Solana's SOL token. This token would be used to pay the nodes hosting all of the data. In order for a user to store data on the network, they would be required to pay a storage fee to the nodes. This fee would be dynamically calculated based on the amount of data being stored. Along with the storage fee, we can charge a maintenance fee to the user, which pays the nodes to extend the block expiry dates. This maintenance fee can be charged to the user every network cycle, which we can define as 1 month (or any other number we agree on).

In order to provie an incentive to users to only store data for the amount they need, we can put the maintenance fees into a limbo state, and are slowly payed out to the storage providers, as the network cycle inches closer to its deadline. If the user submits a request to delete the data out of a block, we can free the tokens back to the user, based on the percentage amount of the data that was deleted. This way, we can make sure that the users are only paying for the data that they're using.

Along with these fees we can also mint fresh tokens to storage providers. In order to prevent hyperinflation, we can agree on a number, which is the starting mint reward to storage providers, and adjust it based on the network size. This way we can make the reward lower as the network grows, which ensures that a stable amount of freshly minted tokens is constantly being put into circulation, while not having an insanely high inflation rate. Of course this assumes that as the network grows, the demand for tokens will increase, thus making a smaller reward as worth it as the old one in that older time period.

6 Security

Data security/privacy is an extremely important part of the protocol.

The first thing we can do for the security of data, is to make the data unreadable by anyone who is not the original owner/submitter of the block. Like on every other blockchain, the user is required to make a private/public keypair. When submitting data onto the chain, we can require pushing the submitters public key into the immutable part of the block, which effectively assigns the block's owner. When submitting this data, we can encrypt the data using the submitters public key. This makes the data unreadable by anyone who does not possess the private key of that keypair, which by norm is the case.

The second thing we can do for the security of data, is to make the data only mutable by the person who originally submitted the data to the chain. We can do this via digital signatures. When submitting the new encrypted data, we can sign the new data using our private key, which is then compared to the public key that is encoded in the immutable region of the block. If they match, the nodes update the modified block accordingly.

7 Consensus

The DeFile protocol uses a Proof-of-Validity consensus system, which is mentioned in the previous section. This consensus system simply requires the user to sign their blocks as required before submitting them to the chain via a node.

While a node is absent from the network, the chain will become larger than when they wrote to their own local storage. When reconnecting to the network, we can contact several other nodes with our current last block id. Those nodes can then respond with every block on their local storage after that block, and the requesting node can then accept the chain that appears most commonly between all of the recieved chains as the valid one. This way, as long as most of the nodes stay honest, we can be sure that every node will be up-to-date with the latest data.

We can also use this consensus to transfer native tokens between users. In each block, we can assign another immutable section of data, this time for transactions. In this section we can write various transactions, paying nodes fees for storage, or transfering tokens between users. For transfering tokens between users, we can utilize the same system of paying fees, since they both effectively transfer tokens from one user to another (referring to a node a user, for the lack of a better term).

This consensus system saves a lot of computing power, which would be otherwise wasted in a consensus system like Proof-of-Work.

8 Conclusion

The DeFile protocol seeks to fix many of the of the underlying issues with conventional data storage solutions. We first described the idea of a mutable blockchain, which allows for users to sumbit and modify data at will, as long as they can prove their ownership of the block. Further, we implemented a security system, where all data on the chain is encrypted, making it inaccessable to any user who does not possess the keys to that data. We implemented a data control system, to control the amount of data stored on the chain, via trimming out unused data to free space, and by distributing between nodes evenly as the network grows bigger. We proposed an incentive for nodes to stay honest and store data, by creating a native token, that is used to pay fees to the nodes and to freshly mint a token reward to the nodes, which scales as the network grows. The network uses a Proof-of-Validity consensus system, which serves to allow faster speeds than other consensus systems, such as Proof-of-Work, to allow computationally inexpensive storage.

Revision Changes

- Removed the "Disaster Prevention" chapter, as it was no longer a protocol flaw.

References

1. Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", https://bitcoin.org/bitcoin.pdf, 2008
2. Vitalik Buterin, "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform", https://whitepaper.io/document/718/ethereum- whitepaper, 2014
3. Anatoly Yakovenko, "Solana: A new architecture for a high performance blockchain v0.8.13", https://solana.com/solana-whitepaper.pdf
4. Protocol Labs, "Filecoin: A Decentralized Storage Network", https://filecoin.io/filecoin.pdf, 2017