BrightChain

1. Abstract

BrightChain intends to become the de facto international and interplanetary standard for data storage, identity, and digital contract fulfillment.

At its heart is a concept introduced by The Owner Free File System which breaks a file up into source blocks and merges them with blocks of random data using an “exclusive or” operation and discards the source blocks. Added on top of that, we introduce identity/anonymity, reputation, block revocation and expiration. What the Owner Free Filesystem called “whitening”, we call “Brightening” and where BrightChain gets its name.

2. Problem

BrightChain addresses not one, but three central problems:

1. Wasted unused storage, lack of storage where needed.

Computers and devices with unused storage are everywhere, and yet no mainstream solutions exist to both make use of the wasted space, as well as to ensure that participating nodes have immunity to takedown requests.

2. Wasted Energy in Blockchain

In recent times, Blockchain has become a hot area for research and development, especially with the rise of Digital Contracts. However, most of those systems rely on a network that was designed around creating artificial scarcity for the sake of monetary and trade equivalence. There is a significant amount of energy waste in blockchain operations as a result.

3. Reputation, Anonymity Versus Accountability and the Parler Problem

January 6th, 2021 and the Parler network revealed a number of problems with the current state of Social Media, and the overall ability for both malevolent and misinformed people to go unchecked.

3. Solution

BrightChain addresses the three central problems as one.

Energy waste reduction and tracking is the primary goal of the network. Contributions in the form of storage, CPU, even content and more are tracked like most Blockchain networks track fees or Gas. BrightChain goes a step further and ties your reputation to your data, but gives a variety of options for both node operators to decide what they want to allow users to do, and users to decide what they want to make private or public. Private contributions have an energy waste factor that is intrinsically higher as it is less likely to be used heavily when compared to public blocks that can be discovered and shared more readily. Private blocks are more likely to be cold stored and need lower priority redundancy and access speed requirements. Operators can choose which private and public block sizes to store and serve for read/write separately so that sizes may be deprecated and replicated out without penalty.

The claim of immunity to takedown requests comes through all of the content being broken up into multi-use fragments of files. No one block contains anything but totally random data that may be a part of multiple files.

Deduplication of data is baked in at every level. Block IDs are simply their SHA-256 hashes. There can be no two blocks with the same ID and duplicate content. At the point that SHA-256 ceases to meet the needs of the network with collisions, new blocks may be placed with a newer algorithm and a client update. However, all blocks are random and their original meaning is lost, so an index of public handle-blocks is kept with the hash of the source file and maps it to the blocks that store its data. Private blocks are stored encrypted with the user's key. Moreover, all blocks at time of storage are contracted for a minimum "Keep Until At Least" and a minimum storage durability/accessibility.

Every time a block is accessed, its "usefulness" goes up and its staleness goes down. This will cause the replication system to alter where and how many replicas are kept, and will correspondingly "return" some of the energy anyone who has used a particular block to store their data.

BrightChain considers the negative comments and other data generated by social networks to be essentially bad, unwanted data that should be purged or expired out from the system.

It is not the goal of BrightChain to store every block forever, but to allow it and encourage it for things that are worth it, and let other things auto-extend themselves with use. As a block approaches its expiration and is accessed, the node may extend the contract without the original storer initiating it.

Proof of Work is used as a transaction validating throttle and good actors in the network have near-zero requirement while bad actors have their values temporarily or permanently bumped high by algorithms and essentially force them to work too hard to participate in the network.

The last piece is the identity aspect of BrightChain. Forward Error Correction is used in a unique manner to generate enough error correction data to require a fixed percentage majority reassembling of the shards in order to be able to recover the meaningful data containing the identity of the original poster before the blocks with the data expire out- a digital statute of limitations.

FEC is generated against the true identity, and either a registered alias ID or anonymous ID (all zeroes) is selected and replaces the original ID. The receiving node validates the FEC recovers the original ID and that the user is allowed to store the given block or perform anonymous actions before splitting the shards and giving them to a quorum that is the non-profit, multi-entity governing body of BrightChain. The FEC data is stored in the BrightChain and the corresponding handle-blocks are stored in the private vaults of the Shard-Holders. The original identity data expires out of the network normally if nothing happens. Otherwise the quorum must be requested to assemble the shards and agree to do so according to the bylaws, to be determined.

4. Current state

BrightChain is in its very initial stages.

- Raw file data is being persisted and retrieved from the blockstore with verified integrity, using the resulting handle-block, called a Constituent Block List or "CBL".
- A mechanism to store chains of arbitrary serializable C# objects in the blockchain, called ChainLinq is nearly complete and will facilitate using Linq to write Digital Contract based Applications or "dApps".
- The blockstore is non-replicated, non-expiring, and based largely off of Microsoft's FASTER KV store.

5. Potential paths forward

- Once ChainLinq is complete, the pieces will be in place to store the identity blocks with the keys and user data for the Blockstore API end of things.
- The identity, reputation and digital contract aspects are on the next 3 sprint boards.
- The replication will likely be done with Microsoft's Orleans in conjunction with FASTER.
- Much help is requested in the way of collaboration on the reputation math aspects especially, but any help at all is much appreciated.
- The BrightChain governing body is yet to be determined, but its size and members should be chosen carefully.

6. Conclusion

BrightChain offers the power of digital contracts on top of the .Net platform and makes the blockchain as easily accessible as Linq arrays, without all of the mining waste of traditional blockchains. It is a mathematically guaranteed overall positive experience for all participants.

Project URLs:

https://github.com/The-Revolution-Network/BrightChain
Main repo/wiki/discussion/project board

https://apidocs.therevolution.network/
Auto-generated documentation and GitHub wiki mirror.

https://github.com/BrightChain/BrightChain
Currently a mirror of The-Revolution-Network

https://brightchain.net/
Homepage/logins coming soon

Other URLs:

https://en.wikipedia.org/wiki/OFFSystem
Owner Free File System

https://aka.ms/Faster
FASTER project

https://aka.ms/Orleans
Orleans project

Last Updated 08/12/21 00:44 Pacific time

jessica@brightchain.net