

Peer To Peer Personal Cloud Storage Proposal

By: Richard Zins

With the invention of cloud storage the average consumer now has the ability to have all their essential files synced on all of their devices. Although this convenience comes with some drawbacks. Since all these cloud storage systems are controlled by their respective companies (i.e Apple controls Icloud, Google controls Google Drive, and Microsoft controls OneDrive etc) there is a possibility of the companies viewing your private files, censoring your files, or even losing your files in a potential failure in their systems. Albeit I think the odds of this happening at a wide scale is probably low, but there still is potential for the loss of confidentiality and integrity. That's why I think data confidentiality, data integrity, and transparency need to be built into our cloud storage solutions. Hence I would like to propose for my capstone project that I attempt to create an open source decentralized distributed cloud storage network based on peer to peer networking techniques.

In this cloud storage solution users will act as nodes in the network, and must contribute at least a small amount of their local storage to the network in order to use the service. Also there will be incentives for users with lots of excess storage to contribute more local storage than the average user. I believe that this solution will inherently solve the problems of data confidentiality, data integrity, and transparency for many different reasons. For starters given that the application will be open source there will be massive amounts of transparency for the community to audit and contribute the project's code base. Data confidentiality will be ensured by sharding the files, and distributing each shard to a different node. Hence not one user in the network could look at any other user's complete file (also the shards will be encrypted so they won't easily be able to view a shard). Data integrity will be ensured because each file shard will be copied and distributed to multiple different nodes on the network. That means if one user removed their device from the network there would still be copies of the shards elsewhere. This would mean natural disasters, and users who stop using the service are less of a concern. Lastly each user would get to vote on what updates will be rolled out to the network. Their vote will be weighted, and will have more weight based on how much of their own local storage they contribute to the network. This will democratise the network, and is a measure in place to make sure not one organization can take control of the direction of the project without all the other stakeholders being in agreement.

The technology stack behind this project is subject to change, but I do have a couple of main components that I plan to use. At the moment I plan to use the Go programming language because it is fairly fast, it was designed for use with complex network based applications for Google, and because of its built in concurrency features (i.e Goroutines). Also while implementing a custom peer to peer networking library would be fun I think it would take too much time so I plan on using either libp2p

(<https://github.com/libp2p/go-libp2p>) or noise (<https://github.com/perlin-network/noise>). Lastly it's important to mention the ADT I plan on implementing across the nodes of the network. This ADT is called a distributed hash table. I plan to implement the distributed hash table from scratch so it is optimised for my specific use case, but there are plenty of implementations to use if this proves to be too time consuming. I plan on implementing my distributed hash table in a similar way as described in this paper: <https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf>

In conclusion I think this project has the potential to become a good alternative to the big cloud storage providers that we are all used to using. Although I think it is important to mention that there are other projects that attempt to accomplish a similar goal. Some of those projects include Filecoin, Sia, and Storj. I hope my project reaches a level of sophistication that surpasses these other competitors, and that my project can gain adoption among a non-technical audience as well as technical one.