# The Ideals of Building Decentralized and Democratized Modern Platform for Monitoring and Evaluating Kelp Growing Sites in British Columbia

Xi Sun V00025479

Ting Liu V00045129

Zheng Yin V00915261

Cloud platforms are popular because they can offer users with functions such as real-time collaboration and multi-device access. However, in the cloud, the ownership of the data belongs to the server, not the user. How to ensure the privacy of user data is a very important aspect that needs to be considered in the future, especially in a community context. In the pursuit of better tools, we explore a distributed collaborative system of the future according to the needs of stakeholders. We will show that it is possible for users to retain ownership and control of their data, while also benefiting from seamless collaboration and access from anywhere.

## Platform Strategy and Technical Support

This platform enables both collaboration and ownership for users, such as working offline, collaborating across multiple devices, ensuring the security, privacy, long-term preservation, and user control of data[1].

Since we have analyzed the methods of data collection and processing in the previous reports, we will mainly discuss how to build the storage system and data models in this report. First of all, We adopt decentralized storage, which is a storage business model that uses distributed storage technology to store files or file sets in pieces on the disk space provided by different suppliers, as the system database. Decentralized storage can achieve strong privacy protection, low storage costs, data redundant backup storage, and high speed operation. We need to consider several points-scalability, consistency, and transparency, when designing a distributed storage system. Sharding strategy and metadata storage are the core of the database[2]. We hope to establish a multi-level sharding strategy, which uses hash at the top layer, while the data is stored in order in each hash-based sharding unit[2]. Each range shard will use the Raft algorithm to ensure data security and high availability on multiple physical nodes, and data will be divided into range shards according to the key range[2].

Furthermore, the key difference between the traditional system and our system is the change of the responsibilities of servers and local devices. Our system treats the copy of the data on the local device as the primary copy, while servers play a supporting role[1]. All operations can be processed on the local disk, and data synchronization with other devices occurs in the background. Considering that some users may operate data offline in remote areas, the system can synchronize with other devices sometime later when a network connection is available. This connection can be made via the Internet, Bluetooth or local WiFi[1]. Servers, as “cloud peers” instead of “central authorities”, will hold secondary copies of data in order to assist with access from multiple devices. For example, a cloud peer that stores a copy of a document will forward it to other peers when they come online[1].

Additionally, the collaborative multi-user is built using a JSON data model. Even in the offline state, users can view or modify the application status on their local device. The algorithm keeps track of any changes made and synchronizes the changes with other devices in the background when the network is reconnected. By including all the added items in a consistent order, it can be concurrently modified and merged on different devices and different objects. When a conflict occurs due to multiple users updating the same attribute of the same object at the same time, the algorithm will track the conflicting value and leave it to the application or user to resolve[1].

At last, a “time travel” interface will be introduced into the platform, allowing users to go back and view the state of the document before the merge, and automatically highlighting details that other users have recently changed. Traversing the history of merged documents in a linear fashion can become a general tool for collaboration[1].

## System Security

Our system has better privacy and security built into the core. One of the advantages is that the local device only stores the user's own data, avoiding the centralized cloud database to save everyone's data. The system uses end-to-end encryption to ensure that any server that stores a copy of the file only saves but cannot read the encrypted data[1].

The encryption infrastructure, transaction records, broadcasting methods, and storage certification mechanisms in the decentralized storage system effectively ensure the integrity of the data and reduce the fraud space and privacy leakage risks of malicious nodes. Since the shared network of decentralized storage is a peer-to-peer network, it is not susceptible to single points of failure and server disconnection, and it is not vulnerable to hacker attacks. Meanwhile, decentralized storage will encrypt the fragments of the stored content, so that no other nodes, including operators, can see the specific content of the storage. Thereby, the specific storage content of the user will not be leaked due to subjective reasons.

## Data Modelling

Our platform will create various models for stakeholders, such as location models, monitoring models, predictive models, etc. For example, in location models, users can use imagery and temperature data to evaluate potential kelp growing sites in BC. The monitoring model can analyze the growth of kelp by modelling the data (water temperature, wave action, harvest characteristics, etc.) collected by the drone to support kelp health and productivity. In addition, periodic data can be obtained through algorithmic analysis to establish predictive models. These models will predict the latest growth changes, growth cycle, yield and harvest time of kelp based on the possible impact of climate or environmental changes. In the future, our platform can also be extended to the monitoring and breeding of other economic crops and organisms in the coastal waters, ensuring and driving the economic growth and environmental protection of the surrounding ocean.

## Future Work

All in all, there must be many shortcomings in our design, and there are also some technical problems, which need to be solved one by one in the future. More work is needed to realize the platform in practice, such as supporting offline work, making use of on-device storage, improving the algorithms, developing programming models and user interfaces. There are still many obstacles in building collaborative software that respects users’ ownership and agency. All of these will be topics that we will continue to study in the future.

## References

[1] Martin Kleppmann Adam Wiggins Peter van Hardenberg Mark McGranaghan April 2019. (n.d.). Local-first software. Retrieved from https://www.inkandswitch.com/local-first.html#future

[2] 4, P. O. (2020, September 13). Building a large-scale distributed storage system based on Raft. Retrieved from https://www.cncf.io/blog/2019/11/04/building-a-large-scale-distributed-storage-system-based-on-raft/