Sharder Technical White Paper

Sharder – A Cross-chain Distributed Storage Protocol

V1.1

Website: https://sharder.org

Telegram: https://t.me/sharder_talk

Twitter: https://twitter.com/SharderChain

Medium: https://medium.com/@SharderChain

Email: hi@sharder.org

Github: https://github.com/Sharders

foundation@sharder.org

Preface

Today's world is a data-driven society. The development of information technology and smart life leads to an explosion in data growth. On one hand, the growth in storage capacity lags far behind that of the data growth so the demand for storage is far from being quenched, on the other hand, there is a great amount of storage space belonging to individuals and corporations that lie idle and are wasting away. Additionally, there are many pain points in current centralized storage systems, such as lack of encryption, proneness to data breach and abuse, mutable, impermanent, and expensive.

Blockchain is a new information technology that integrates distributed storage, consensus ledger, peer-to-peer transfer, encryption algorithm, and incentive mechanism. Its congenital attributes such as decentralization, open-source, autonomy, anonymity, traceability, immutability will effectively solve the pain points of centralized storage systems. The Sharder Protocol is a cross-chain distributed storage protocol based on blockchain 3.0 and aims to greatly optimize the current blockchain technology, starting with the application of distributed storage and expanding to many other commercial applications.

Technology Innovations: Sharder initiates cross-chain distributed storage protocol, and creatively introduces the role of a Watcher and Prover in the network. It develops its own Sharder-UTXO model, which is compatible with the UTXO model of Bitcoin. The Sharder Chain running on Sharder Protocol is advanced with robustness, security, privacy, and availability of the system.

Sharing Economy: The cross-chain distributed storage network based on Sharder Protocol provides the subscribers with a secure, efficient, cheap, and permanent storage service; while allowing the subscribers with redundant storage space to share their extra space for incentives. Similar to Airbnb, the multi-chain scheme based on the Sharder Protocol constructs a distributed, secure, convenient, and permanent sharing network.

Sharder Miners: Sharder will release its micro node miner (Sharder Hub) and the all-in-one storage-miner (Sharder Box), which allow the miners to get multifold rewards by contributing both computing power and storage space to the network.

Commercial Applications: Sharder Protocol is open-source and free, any public chain or storage network could deploy Sharder Protocol, and anyone could develop DApp on it. Bean Cloud is our first application that provides the storage, certification, and security service for the vast e-contracts generated in governments, banks, medical care, e-commerce, etc. There are more ahead. Sharder is developing Sharder Matrix, Sharder Brain, One Fair and other applications on data, AI, and exchange. Besides the conventional data such as photos and documents, Sharder is hopes to store any other data, for instance: biological data (including gene information, growth log, medical records), and even thoughts and memories. Our mission and vision is: Store Your Stories. Your support will help our dream come true.

Administration: In the spirit of openness, transparency, and democracy; Sharder Foundation is in charge of the R&D of Sharder Protocol, Sharder community management, and the promotion of Sharder products and culture.

Contents

1 Abstract

Sharder Protocol is a cross-chain distributed storage protocol. The name “Sharder” comes from the word “shard” in computer science. Sharder client could be deployed on various public chains, storage networks, personal nodes, etc. Sharder Protocol defines multiple objects, action functions, check mechanism, consensus mechanism, contribution quantification, and authorization mechanism, etc., and designs data encryption, data sharding, multi-chain architecture, file system, smart contract, free market, security, availability, flexibility, etc.

Hopefully this technical white paper is helpful for all users, even those without background knowledge in computer science, programming, mathematics, or blockchain, to understand how Sharder Protocol constructs a decentralized storage network.

2 Summary

Vision: To build a global, secure, private, online 24/7, cross-chain, distributed storage network – Sharder Network, and furthermore, construct a cross-chain shared storage ecosystem that reforms the way people store and exchange valuable data.

Token (SS): The cryptocurrency embedded in Sharder Protocol. A total of 500 million SS are issued. Please learn more at the official website: https://sharder.org.

Sharder Chain: The very first commercial blockchain that deploys Sharder Protocol, the very first Sharder Pool (Sharder-Pool$_0$), and the cornerstone of Sharder Network.

Sharder Network: The decentralized and distributed network comprise of various Sharder Pools that deploy Sharder Protocol. In addition to quality data service with a low price, myriad Dapps are developed on top of Sharder Network, such as Bean Cloud (data storage, security, and certification), Sharder Matrix, Sharder Brain, One Fair. Sharder Network will eventually reform the way people store and exchange valuable data.

Sharder Market: The free market where storage, data, and digital assets are exchanged.

Dapps: Bean Cloud: Data storage, certification, and security. Sharder Matrix: Personal biological data storage. Sharder Brain: Big data service such as data security, data dispatch, data analysis, data search, data alert. One Fair: A transparent, public, free, and peer-to-peer market where inert data and value could efficiently and securely circulate and exchange.

Open-Source: Sharder Protocol is an open-source project under the GPLv2.0. The Github is https://github.com/Sharders.

3 Design Philosophy

Imperfect Nodes Presumption: It’s a loose network architecture that allows single node failure and occasional downtime while the whole network remains robust.

Ownership & Privacy: Data is encrypted and private. The data owner has full ownership and access of the data. No one without authorization can access to the data.

Quantified Contribution: All contributions in the network is quantified and observable based on measuring methods such as PoST (Proof of Storage & Time) and PoR (Proof of Replica).

Eventual Consistency: Data objects are allowed to have different states at different nodes, can also rapidly converge to the entire network consistently.

Monitoring & Recovery: The entire network availability and data object states are closely monitored and could be spontaneously recovered to some extent.

Audit & Supervision: Data could be monitored and audited in specific situations with the consent of data owners.

Extendable API: User-friendly API with high extendibility.

4 Sharder Protocol

First of all, Sharder Protocol constructs a distributed storage network, which provides: cost-effective storage space, reliable data storage, and transparent on-chain information. Eventually, all the storage space, data stored, and digital assets will together form a free market. Sharder Network is accessible not only to conventional storage resources, but also to global public chains (such as Qtum, Ethereum), storage networks (such as IPFS, Aliyun, Baidu Cloud), and personal storages (such as hard disks and cloud disks), etc. Any DApp could be developed for free based on Sharder Chain.

4.1 Sharder Protocol Overview

- Advocates openness and transparency.
- Comprises of roles, network, data, contribution quantification, incentives, and multi-chain mechanism.
- Defines quantified proofs of PoR (Proof of Replica) and PoST (Proof of Storage & Time).
- Adopts data sharding, multi-replica, and data erasure to ensure data security and availability.
- Connects to various public chains and networks and forms multi-chain ecosystem to circulate data and value.
- Initializes the authorization mechanism of Sharder-PAIR and Sharder-UTXO to meet requirements of the auditing and supervision of corporations or regulators.

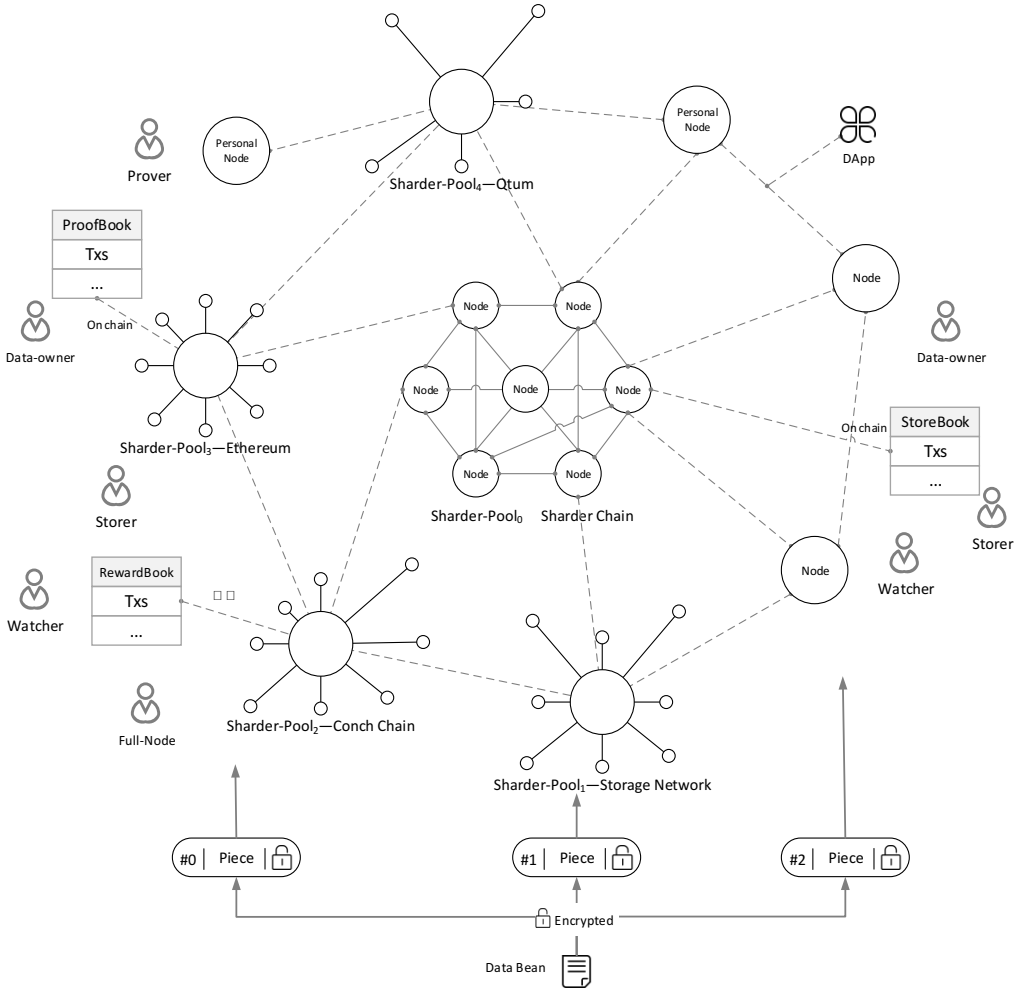

Figure 1 Sharder Protocol Overview

4.2 Definition of Roles

Sharder Protocol defines the roles of all participants in the network. A node could work multiple roles. Full nodes are assessed based on their PoC [3.8.3 Proof of Credit]

Type of Node	Forger	Watcher	Storer	Prover
Full Node	√	√	√	
Storage Node			√	
Watch Node		√	√	
Proof Node		√		√

Table 1 Relationship Between Roles and Nodes

Bean: In Sharder Protocol a data bean is the data object before sharding. Except for sharding, a bean is not divisible and is exclusive to the external system.

Data-owner: The owner of data beans. All data beans have the signatures of the owners, who have the right to check the data whenever it's necessary so as to ensure the data is securely stored on storage nodes.

Data owners would require different levels of security according to the importance of their data. In response, Sharder Protocol adopts proper security strategies, and selects

qualified storage nodes to do the job. Of course, a higher level of security incurs higher storage cost.

Storer: Provides disk capacity for data storage and gets rewards. Storers are also subject to inspection from data owners or Watchers and provide proof of storage. For instance, PoST proves that data is stored during the specific time period and is accessible anytime. Hereinafter "storage node" refers to physical nodes that deploy storage clients.

Watcher: Observes the entire network state, checks the security state according to the security strategy, and fixes existing or potential loopholes. Watchers must be constantly online. They are essential for the rapid convergence of the whole network, and perfect for data indexing.

On behalf of data owners, Watchers would randomly perform beat detection on Storers to ensure data security and availability. Most of the work is done off-chain. We hope Watchers are independent nodes that could check and balance Miners and Storers, further ensuring data security and avoid hostile attacks.

Miner: Similar to miners in blockchain networks, miners need to run a client (terminal or GUI) to store block information and process transactions and bundling. The network's stability, connectivity, and throughput rate highly rely on Miners. Generally full nodes work as Miners to ensure online stability and efficiency. Currently, only full nodes connected to Sharder Chain (Sharder-$Pool_0$) could compete for forging rights, which utilizes PoS or DPoS.

Prover: A role embedded in Sharder Protocol to convert data into digital assets and add public credibility to data. Prover's evidence data, combined with original data objects will be recorded on chain. All data is traceable and immutable.

Most of the time Provers are external institutes or organizations with public credibility. Once the Prover node deploys, any party could join the Sharder Network and work as Provers. For example, in digital copyrights, the most authoritative prover should be the National Copyright Administration. The Prover node could connect to the National Copyright Administration and work as its proxy in the Sharder Network. For data certification, notary offices and judicial organs would be Provers with credibility. On- chain data with notarization would be legally valid.

Full-Node: Stable online physical nodes with excellent bandwidth and processing power. In Sharder Protocol, full nodes could work as Miners, Storers, and Watchers.

Sharder Pool: A mini network (analogous to mining pool) comprises of multiple nodes that deploy Sharder Protocol. Once Sharder Protocol is deployed, the public chain or storage network becomes a Sharder Pool. All nodes should be affiliated to the Sharder Pool. If a node isn't affiliated to the Sharder Pool, it will be affiliated to the zero pool (Sharder-$Pool_0$).

One Sharder Pool could choose not to connect with any other Sharder Pools and remains an isolated network, like a private chain, forming a private cloud storage network. Since Sharder Protocol advocates openness, any Sharder Pool that chose to remain closed will have to pay a fee.

Sharder Network: All full nodes that deploy Sharder Protocol form Sharder Network. As a large storage system, Sharder Network devotes itself to establishing a free market for storage to meet the soaring demand for corporate and individual storage. Meanwhile, the network takes advantage of idle or outdated storage devices, reducing e-waste and bring rewards to storage providers.

Sharder Chain: The very first commercial blockchain network that deploys Sharder Protocol, aka Sharder-$Pool_0$. Sharder Chain is the distributed ledger that permanently records information or data objects. Analogous to the backbone node in traditional

telecommunications network, Sharder Chain also works as the intermediate anchoring network in the multi-chain Sharder Protocol.

In the long run, we hope Sharder Chain will make an autonomous and self-consistent public chain with numerous decentralized commercial Dapps.

Sharder Market: The decentralized trade market on Sharder Network. In the past few years, BitShares and EtherDelta have proved that decentralized exchanges could stably operate. We hope Sharder Network could evolve into a free market with pricing based on supply and demand. Sharder Chain provides market-making and trading services in the early stages. Gradually, sellers and buyers bid by themselves and Sharder only acts as the transaction recorder and price analyst that provides reference prices, historic prices, and smart contract service. Any full node with complete block information could provide reference prices and market-making service. We hope services such as testing, certification, and erasure could also join the free market in the future.

4.3 Network Topology

We need to establish a peer-to-peer network with numerous nodes in and out at all times, therefore it is critical to have a good algorithm for routing table maintenance and searching. The Kademlia protocol (hereinafter referred to as Kad) [1] is our priority, and on top of it we built the peer-to-peer network (with Chord algorithm as backup). The Distributed Hash Table is built with Kad and is based on XOR. It measures distance and greatly speeds up the routing search process, which is critical for the Sharder Network with numerous storage nodes. It takes two steps to implement the Kad network: we will first build a peer-to-peer network based on a simple routing table, we then develop the Kad network while opening up the storage node client.

The node list of K bucket in Kad well describes the online state of nodes. In the future, the PoC algorithm will be adopted to rate the credit , which will be used as weight of ranking to help Watchers select the most adjacent nodes.

4.4 Data Object Operation

$$PRC_{Bean} = (Put,Get,Watch)$$

- $\mathrm{Put}(\mathrm{data}) \rightarrow \mathrm{key}$: Client executes Put protocol to store data, with “key” as the unique tag of the data.
- $\mathrm{Get}(\mathrm{key}) \rightarrow \mathrm{data}$: Client executes Get protocol to retrieve data with the unique tag “key”.
- $\mathrm{Watch}()$: Watcher executes Watch protocol to check the stored data, synchronizes the full network state of the data object, and fix abnormalities such as missing data, data error, unavailability according to different security strategies.

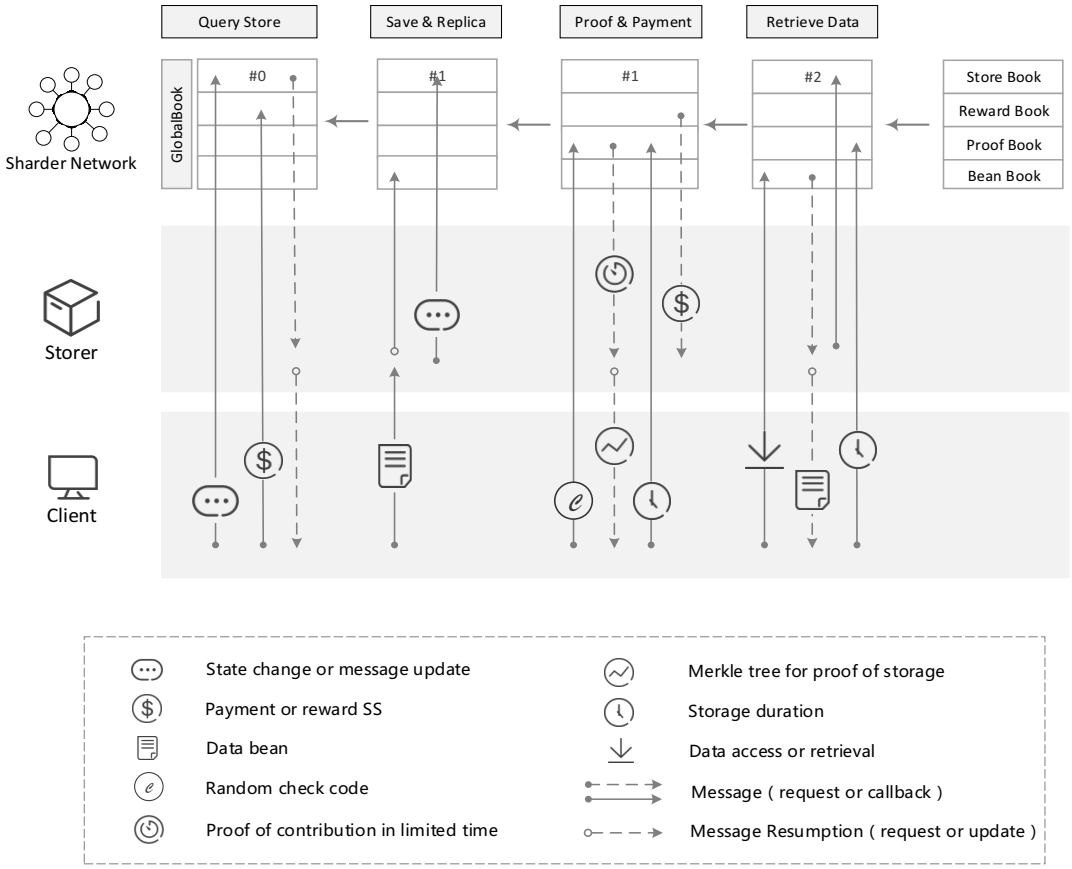

Figure 2 Data Object Operations

4.4.1 Data Storage

a. Client requests data storage on Store-Book.
b. Client pays storage fee and Sharder Protocol responds with the matching Storer.
c. Client uploads files to the Storer.
d. Storer accepts data and updates global state of Store-Book and Bean-Book.
e. Storer broadcasts Replica-Task to the network according to security strategies.
f. Other Storers make the replicas and check if they meet the quantity requirement defined by security strategy. If not, the Storer will keep broadcasting Replica-Task to the network.

4.4.2 Data Retrieval

1. Client requests data retrieval, Sharder Protocol obtains the latest data objects from Bean-Book, gives it to Client and synchronizes the request to the Storers.
2. In active mode, Client connects to the Storer and pulls data from the Storer. While in passive mode, the Storer pushes data to Client.
3. Storer updates Store-Book after data retrieval.
4. Storer updates the global state of Store-Book and Bean-Book after data retrieval.
5. Client updates Proof-Book after data retrieval to prove the Storers have stored the data objects.

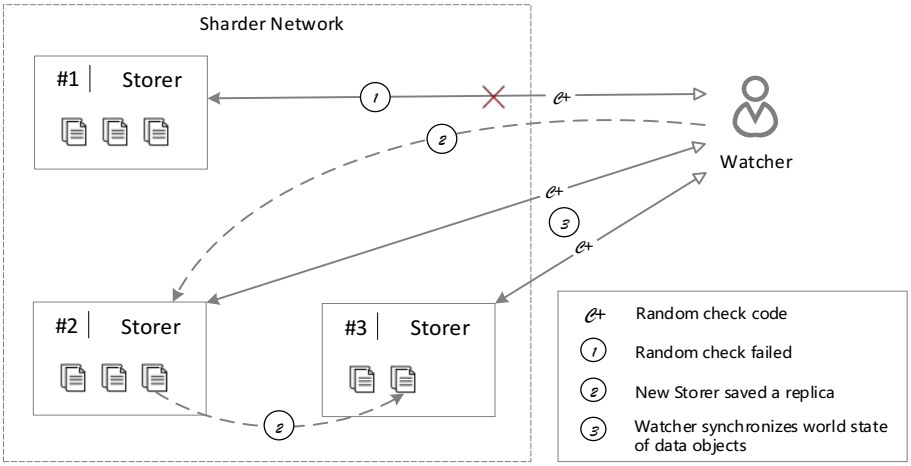

Figure 3 Watcher checks and adjusts data storage

Replica Adjustment: If a Storer is unavailable, the Watcher may require another node to make a replica. Sometimes the original Storer is simply temporarily offline or are experiencing unexpected downtime. The question is, when it comes back online, which node will be eligible to store the data and get rewards? Sharder Protocol will decide by the credit of the nodes. Generally speaking the Storer with the higher credit is allowed to store the data and get rewarded, while the Storer with lower credit has to delete the data.

Resource Access: Data shards are stored and replicated randomly on the global network. The first step of data retrieval is to know which nodes to retrieve data from, i.e. the data indexing is necessary. The Watcher is an ideal indexing service provider which keeps observing the data state and adjust data distribution to rapidly converge the entire network state. Therefore Watcher could provide users with the latest address of data objects.

4.4.3 Data Check

1. Client or Watcher randomly generates check code C and records the check on Proof-Book.
2. Sharder Protocol requires Storer to generate the proof of storage M in response to the check code C.
3. Storer provides the proof of storage M to the Client or Watcher within limited time.
4. Client or Watcher updates the Proof-Book after the check is completed.
5. After the check, Sharder Protocol generates Reward-Book and unlocks part of the reward for the Storer. Sharder Protocol introduces merkle tree [6] and zh-SNARK [7] for the Storer to prove the storage. The random check on storage is initiated by the data owner or Watcher.
6. Please refer to PoR [3.8.1 Proof of Replica] and [3.8.2 Proof of Storage & Time]. Watchers need to regularly check the entire network data objects according to the security strategy [3.6 Data Availability] and maintain the entire network consistency, and are obligated to fix existing or potential security or availability issues. (such as ① shard missing or unavailable, ② Storer longtime unavailable and beyond threshold.)

4.4.4 State Convergence

The time consumption for data replication, the Watcher's check and adjustment on data distribution could be seen as a problem of the data object convergence proof. The proof is as follows:

Assume there are N nodes in the network, and the time to store a copy of data is S_t. In the extreme situation, after N-1 queries, the last node in the network responses and is recognized as available. The time complexity of replication is $O(N)+S_t$. Since S_t is a constant when the network is stable, the time complexity could be simplified as O(N), i.e. the time consumption to look for the available node. The O(N) of the network with frequently in and out nodes will be intolerable. However, the introduction of k-bucket in Kad network could help reduce the time consumption for available nodes lookup. Assume the target lookup node is t. Since every query could obtain information from the k-bucket closer to t, every recursion operation saves at least half of the distance, and the query could rapidly converge with the rate of O(logN).

4.5 Data Security

Data Encryption: Data files are encrypted(AES-256-CTR)in client before they are stored in Storers, i.e. Storers have no access to the file content. As for confidential data, the owner could hardware encrypt the data before storing it on the Sharder Network.

Data Bean Sharding:

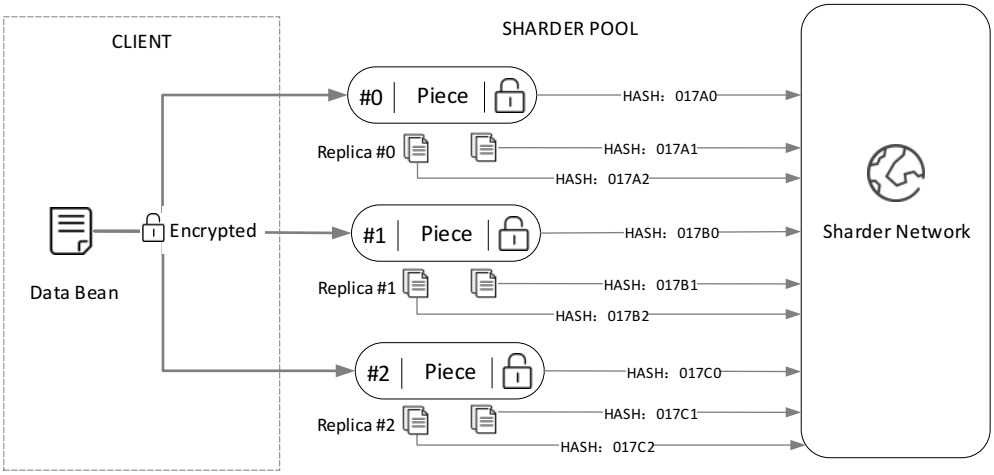

Figure 4 Data Bean Sharding

The strategy of data bean sharding (hereinafter abbreviated as sharding) is highly related to security strategy. If data owner requires high data security, sharding will be of great help. To ensure the availability of sharding, we introduce erasure.

We believe the Storer will not cheat for very small files. Storers won't gain significant incentives by deleting data files and keeping R proof. Most of the time, bandwidth and disk I/O is the performance bottleneck of common Storers, which implies that the sharding files don't fill up the disk space of the Storer. However, too many small files will cause access latency, which could be solved by the high-performance concurrent data processing of the Sharder file system [3.17.1 Sharder File System].

Multi-replica: Assume there are b Storers in the Sharder Network, the data bean is split into p shards and each has n replica. The probability of successful retrieval R_s will be:

$$R_S(\mathrm{b},\mathrm{p},\mathrm{n}) = \frac{\binom{b-p}{n-p}}{\binom{b}{n}}$$

b: Quantity of Storers in the Sharder network p: Quantity of shards n: Quantity of replicas

Code Clips

```
double fac(int p){
    return p == 0 ? 1  : approximation(p * fac(p-1));
}

double choose(int h,int k){
    return fac(h) / fac(k) / fac(h-k);
}

double rs(int b,int p,int r){
    return choose(b-p,r-p) / choose(b,r);
}

double retrieve(int boxerCount, int pieceCount, int replicaCount) {
    return rs(boxerCount,pieceCount,replicaCount);
}
...
```

Retrieval Probability

Storer	Piece	Replica	Retrieve
100	10	10	5.776904234533874E-14
100	10	50	5.934196725858287E-4
200	10	50	3.7276043023296E16
200	50	90	5.7872010853195E44
300	80	90	4.094234910939596E131
500	50	200	3.146459521303754E45
...			

Security Strategy: The basic security strategy is like the normal disaster recovery plan, to create 3 replicas for a copy of data: one at the node or its adjacent node, one at a remote node, and one overseas. However, higher level of security strategy requires more storage space, more complex “watching” and “adjustment”. Sharder Protocol allows data owners to define security strategies according to their own needs. The current configurable parameters are: replica quantity and shard quantity. The security strategy will directly influence how Watchers fix the issue of lost data; and will impact the convergence speed of the entire network’s data objects.

4.6 Data Availability

Data Erasure: Erasure Code (Erasure Coding, EC) [2] is a way to protect data by splitting data into pieces, expanding and numbering the redundant blocks, and storing them at different places such as hard disks, storage nodes, or other physical devices. In order to ensure data availability without taking too much storage space (increasing the utilization of Storers), the Sharder Network adopts data erasure on bean shards.

Reed-Solomon Code (hereinafter abbreviated as RS Code) is a frequently used erasure code, which has two parameters n and m, written as RS (n, m), where n stands for the original block quantity and m for the check block quantity. The comparison between full replication and RS erasure code is as follows: (Please refer to [4] and [5] for the detailed algorithm, we will not go into details here.)

Type	Disk Utilization	Computing Consumption	Network Consumption	Restore Efficiency
Full Replication (3 Replicas)	1/3	Very low	Low	High

RS Erasure Code	n/(n+m)	Fairly high	High	Low

Distribution Adjustment: Watchers continuously adjust the data replication and distribution to ensure the current data file is secure and at least one resource is accessible.

4.7 Consensus and block generation

The PoW consensus in the Bitcoin network demonstrates a concise and explicit economic incentive and consensus mechanism, and also proves that a non-host distributed network could function well. However, we believe it's a waste of resources to utilize expensive hardware and consume huge amounts of electricity and computing power merely to compete for block generation rights. Moreover, the hardware "arms race" also causes a great amount of e-waste. We hope the consensus block generation will make the best out of computing power while protecting the network.

Consensus Block Generation: The core factors of multi-chain consensus block generation include Tx-Bundle and Sharder Block, as demonstrated in the figure below. This method allows each Sharder Pool to have its own internal consensus and Tx-Bundle, which is a block that contains many transaction records. Ultimately the Full Node will generate Sharder blocks and broadcast it to the Sharder Network, with each Tx-Bundle recording the data information including: Node-ID，Pool-ID，and Area-ID.

A Full Node only connect to one Sharder Pool. In order to bundle a Sharder block, the node has to connect to the Sharder Chain (Sharder-Pool$_0$). We are trying to allow Sharder Pool to bundle blocks on its own. A feasible scheme is to deploy at least one proxy node (Sharder Agent) that connects to the Sharder Chain in each Sharder Pool.

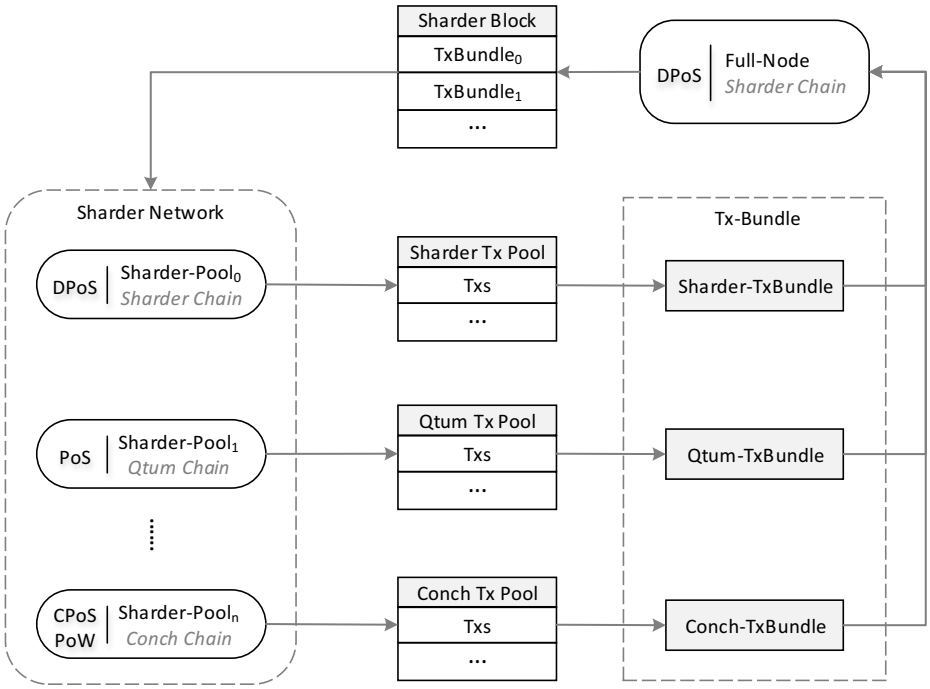

Figure 5 Multi-Chain Consensus

Information On-chain: Not all data and information need to be on-chain. As a matter of fact, most of the data in the Sharder Network is not on-chain. For instance, data file is not stored on-chain. The on-chain URI works as an indicator pointing to the available resource address of the data object.

In addition to the basic block information, other on-chain information includes: accounting transaction, data object, storage transaction, certification transaction, etc. It is notable that one storage transaction corresponds to one data object but probably incurs one or multiple reward transactions (the storage reward based on PoST).

4.8 Contribution Quantification

The Merkle tree [6] and zh-SNARK [7] are introduced to create PoR (Proof-of-Replica) and PoST (Proof-of-Storage & Time) to show proof of contribution of Storers. The credible Storers could provide proof of replica with PoR in short time; while Storers with lower credit will have to adopt PoST to provide proof of storage and time.

4.8.1 Proof-of-Replica

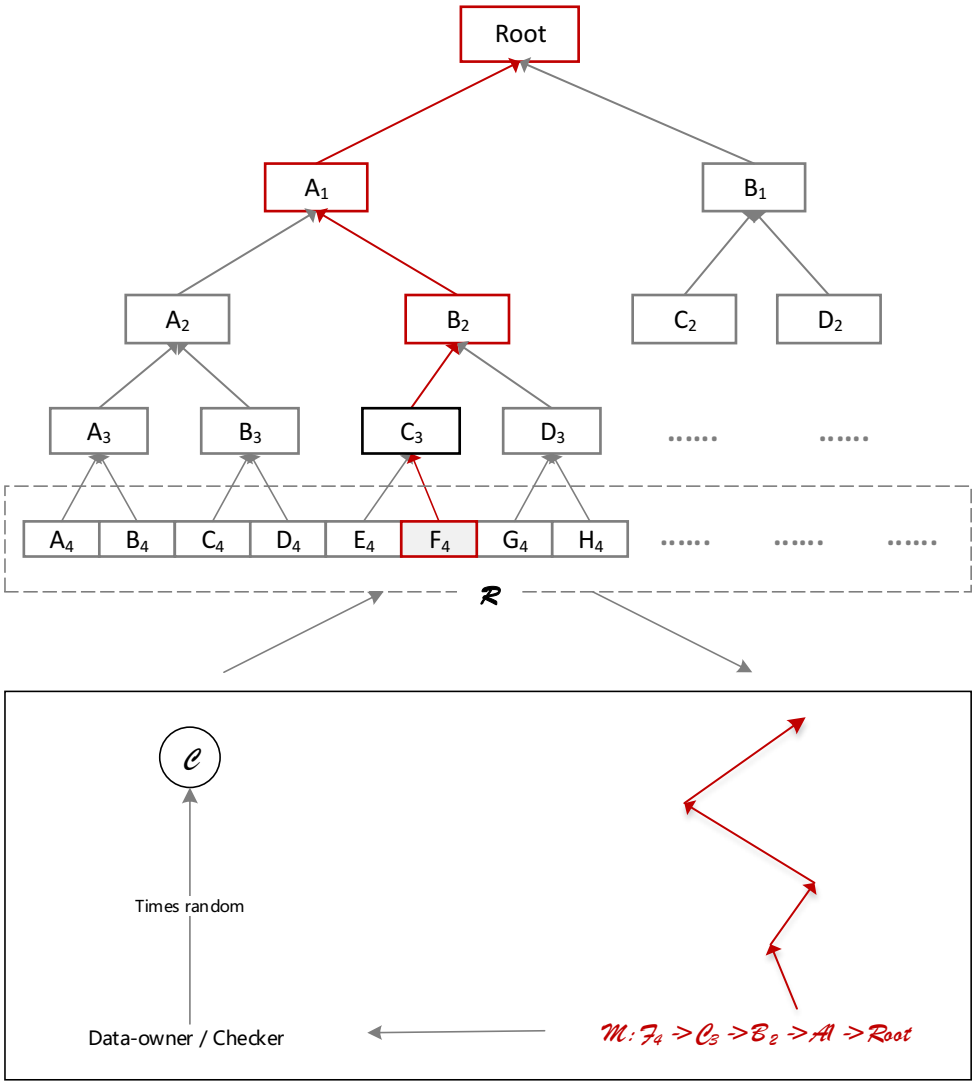

Figure 6 PoR Flowchart

To ensure that Storers will keep the data replicas in Sharder Network, the data owner could request from time to time:

- Data owner creates check code C based on time and send it to Sharder Network.
- Storer finds the data shards according to C and create $\underset{M}{\rightarrow}$ (Merkle tree).

- Once the check is passed, Sharder Network will update Store-Book and Reward-Book so as to issue the reward to the Storer.

4.8.2 Proof-of-ST (Storage & Time)

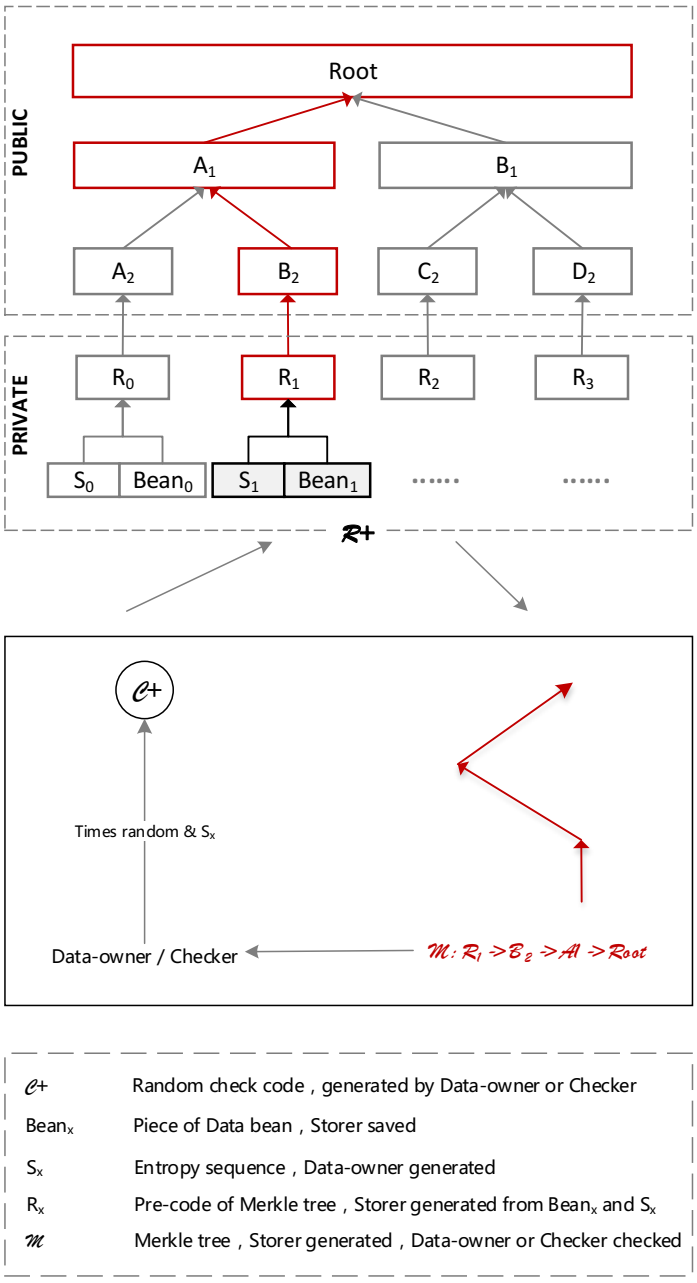

Figure 6 PoST Flowchart

Although PoR ensures that Storers will store data and make replicas, it couldn't prevent malicious Storers from cheating. Think of this:

1. After replicating data and calculating the Merkle check number for all shards and split sequence, Storer deletes sharding files and only keep the Merkle tree.
2. When receiving a request for proof, Storer requests data from other replica nodes and calculates the Merkle tree for C.

Therefore, we introduce the upgraded proof of PoST to ensure that if the Storer deletes sharding files, it will be unable to calculate the Merkle tree and can't pass the check.

- Data owner creates an entropy sequence S after data sharding and then utilizes it along with data shards to generate Hash value R.
- Data owner sends S_x (entropy value based on time, can't be repeated) to Sharder Network from time to time to request PoST. The Storer then calculates R_x based on S_x and data shards, and further generates the Merkle tree based on R_x.

With the entropy sequence S, the Watcher could check the data on behalf of the data owner. The data owner could provide the Watcher with part of the entropy sequence and the latter will excecute PoST. This kind of proxy could also be realized with smart contracts.

4.8.3 Proof-of-Credit

In Sharder Protocol, the PoC is attached to accounts. Deriving from CPOS (Conch-Chain Proof of Stake), the Sharder PoC formula's parameters vary for different roles.

- **Storer:** Total storage volume, storage duration, online duration, amount of penalties.
- **Full Node:** maximum transaction processing, block generation speed, fork convergence speed, online duration.
- **Watcher:** indexing service performance, online duration.
- **Data Owner:** data storage volume, transaction volume.
- **Prover:** proof volume

4.9 Incentives

4.9.1 System Reward

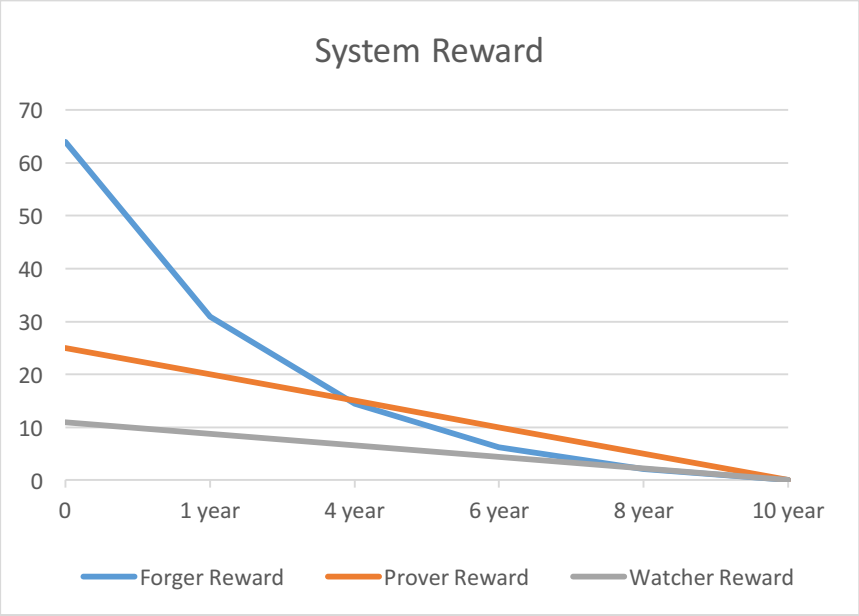

Figure 8 Reward Distribution

In order to encourage more nodes to join the network and build a more secure and robust system, Sharder rewards the nodes that contribute to the network.

Block Generator's Reward: The block generation time is set at 5 minutes (could be as fast as 10 second per block), so 288 blocks per day, 105,120 blocks per year. The

generator gets rewarded 218 SS per block. The reward will be halved for every 210,240 blocks until 1,208,160 blocks are generated, the reward will then remain at 14 SS per block. It is estimated that 64,000,000 SS will be rewarded within 10 years. After all rewards are given, the storage transaction service fees and the technology service fees will be used as economic incentives for network maintenance.

Watcher's Reward: During the early stages of Sharder Network, the official nodes work as Watchers. When the entire network achieves a stable state, full nodes will compete for Watchers. The system reserves 25,000,000 SS as Watcher's rewards, which will be calculated based on the Watchers' PoC, and automatically distributed to Watchers by smart contracts.

Prover's Reward: Sharder Network reserves 11,000,000 SS as the Prover's reward to encourage organizations or institutes with public credibility to work as Provers and provide PoA (Proof of Asset) service. Sharder also allows data owners and Provers to freely set the service pricing.

4.9.2 System Penalties

Sharder Network penalizes the following detriments to the network by confiscating Sharder tokens and lower the credit of the node:

Losing Data: The nodes that lose data will be disqualified from any rewards for future data storage and their credit will be lowered. If the credit falls below the threshold, the node will be blacklisted and banned from the Sharder Network permanently.

Malicious Attack: Behaviors such as maliciously refusing to generate blocks, cheating for SS, attacking the Sharder network incur extremely severe penalties. The nodes and involved users will be blacklisted and permanently banned from the Sharder Network and all SS on the nodes' accounts will be confiscated.

Fraud: Frauds are generally detected after fraudulent behavior has taken place. Therefore, there's no current effective way to recover the loss. The current penalty is to lower the user's credit. Once the credit falls below the threshold, the user will be blacklisted and permanently banned from the Sharder Network. The best way to avoid fraud is to increase cost for fraudulent behavior. We may request a deposit from nodes in some circumstances.

4.9.3 Transaction Reward

Rewards for contributing to the Sharder Free Market.

Storage Reward: Storers that provide storage space get rewarded from data owners; pricing is decided by market.

Service Reward: In the future, when the Sharder free market is active enough, nodes could earn rewards by providing customized services (such as independent data indexing service, customized light wallet client, etc.).

4.10 Sharder Tokens (SS)

Sharder (SS) is the cryptocurrency embedded in Sharder Protocol. It acts as the incentive in the Sharder ecosystem to reward nodes that contribute to the network, penalize malicious nodes, and to avoid the possible infinite-loop logic bomb. Additionally, SS works as the anchoring token in Sharder's multi-chain architecture.

4.11 Smart Contract

Smart Contracts have been proven to be credible and efficient by Ethereum. Smart Contracts of the Sharder Protocol will be realized in two phases. Phase I will be a non-Turing-complete smart contract that adopts Sharder tokens as incentives to support the upper level of the transaction model. Phase II will add advanced features such as virtual machines, Turing-complete, oracle, Hash lock, etc.

Phase I smart contract adopts the classic FILO stack structure. The basic definitions of atomic operation are: OP_INIT, OP_EMPTY, OP_FULL, OP_PUSH, OP_POP, OP_DUP, OP_COUNT, OP_HAS, OP_PROOF, OP_CHECKSIGN, OP_EQUAL. As the apps increase, more atomic operation characters will be utilized to realize more complex operations. A clip of smart contract for reward is as follow:

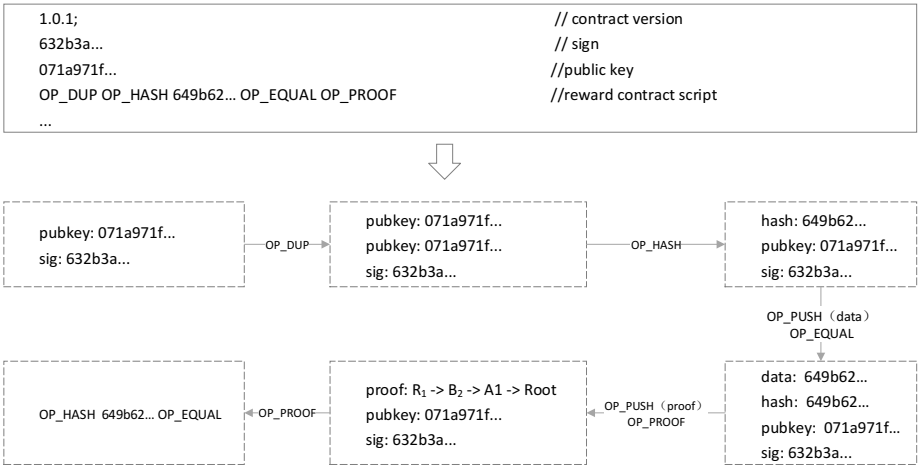

Figure 9 Non-Turing-complete smart contract

The execution is demonstrated above. The storer that correctly provides proof gets the reward. However, the public key (calculated by OP_HASH and get 649b62…) is required. Phase I smart contract is based on UTXO model (simply put based on address) and need the upper level of Sharder accounts to merge the transactions with addresses.

Smart contracts is an important component of the Sharder free market. It safeguards the status, changes, and payment of transaction orders such as Store-Book. Moreover, in the future, various whole network Data-Books will be operated and constrained by smart contracts; with the Watcher as one of the parties. In Phase II we will design and implement the account-based Turing-complete smart contract.

4.12 Client

GUI and CLI clients are available.

Storer Client: Only provides Sharder Network with local storage space. The Storer Client doesn't store all node information but only the data files and contribution checking files. The node deploying the Storer client works only as the Storer in the Sharder Network. In addition to local disk space, popular storage options such as personal web disks and cloud storage will be added when possible.

Full Node Client: Provides Sharder Network with not only storage space, but also all functionality of bundling block generation and Watchers.

Watcher Client: Full Nodes could open the Watcher function and compete for the Watcher role. Watchers need to monitor the state of changes in the Sharder Network and send instructions to fix security and availability issues.

Prover Client: Provers could check and provide the certification service for authorized on-chain data. If the Prover has an independent information system, it will be integrated with API.

We will also provide well-rounded SDK and API. All clients will have complete wallet functionalities. Mobile light clients will also be available, but it needs a connection to the Full Node to function.

4.13 Multi-chain Ecosystem

Sharder Pools that deploy Sharder Protocol together form a multi-chain ecosystem.

Transaction Exchange: Each Tx-Bundle includes the transaction types defined in the Sharder Pool. Eventually the Tx-Bundle will be packed into a Sharder block and broadcast to the Sharder Network.

Value Exchange: Each Sharder Pool can define its own token. SS can be used in the Sharder Pool to complete different types of transactions. We will start with the testing of the multi-chain architecture between Sharder Chain and Conch Chain. Later we hope to implement the cross-chain trading of information and value. We believe the lightning network based on homomorphic channels and smart contracts would be a great option for the connection between the Sharder Network and other blockchain networks.

4.14 One Fair

One Fair is a free peer-to-peer market which comprises of various whole network books, smart contracts, and trade parties. The free market in Sharder Network is not a high-frequency market. The current architecture involves the entire network ledger state update and synchronization, thus ultra-high frequency trading is not feasible. If the Sharder market need to deal with high-frequency or even ultra-high frequency trading in the future, we will adjust the architecture by separating information flows from cash flows and move transactions to downlink and reserve the uplink for cash settlements.

Whole Network Ledger: Currently includes Store-Book, Reward-Book, Proof-Book.

Trade Parties: The trade could be executed in accordance with the smart contract. Once broadcasted to the network, the trade will be carried out even when the nodes are offline.

Smart Contract: Since there's no centralized market maker, the Sharder Network is dependent upon when matchmaking transactions. Currently, the order of a purchaser's transactions are not considered when matchmaking transactions, instead the price and storage demanded determine the matchmaking.

Free Pricing: Ideally the transaction is priced by the trade parties, matched by the market, and executed by the smart contract. However, in the early stages, Sharder Chain will act as the price collector and notifier to provide market reference price and matchmaking to ease the price setting and transaction process of the trade parties.

Commission: Commission is the tokens needed for executing the smart contract. Commission will be priced and charged by the market makers if they make the deal. All commission accounts will be settled in Sharder tokens SS. In the future, commission could be waived, for instance those who help match make the market or confirm the transaction may be exempt from commission.

Sharder Chain: Sharder Chain will always be the seller of storage space and will always ensure there’s a sufficient and stable supply of storage space.

4.15 Authorization Mechanism

Supervision and auditing to some extent is still necessary in various circumstances and should be with the awareness of both parties. On top of this idea, Sharder Protocol provide a fundamental framework to help with the authorization and audit of the accounts. So as to facilitate the individuals, corporations, and regulators to access the data on behalf of the users.

Account Authorization: First of all, the classification and audit should have the data owner's awareness and authorization. We imitate BIP39[9]'s multilevel wallet system to grant credit to all data owners. We will reference BIP44[10]'s path definition mode and the discussion of Ethereum EIP85[11] introduce the path below:

m/purpose'/coin_type'/ account' /change/address_index

Audit: Although block and trade information is public, it's difficult to audit the information as the trade parties are anonymous. The auditor will need the authorization of both transaction parties to unlock and crosscheck the account information and will have to repeat this procedure to audit transaction information. The audit results will be on-chain. Batch authorization could be designed for commercial applications to avoid repeated procedures, especially when C terminal users trust B terminal users.

KYC: Sharder Protocol doesn't identify the users. High level users could determine how to implement user identification by themselves.

Information Classification: Similar to access control, information classification encrypts different data objects with different derivative keys (different HD routes). A derivative key is only able to decipher a specific part of data. It requires a complex logic for key generation and data encryption.

4.16 Malicious Attack

51% Attack: The is an issue that plagues all blockchain systems, including Sharder Protocol, and there's no way to circumvent it. To decrease the probability of a 51% attack, Sharder Protocol adopts PoS and DPoS to generate the blocks and will add PoC to select qualified block generators.

Sybil Attack: Sharder Network requires one transaction to check with at least 3 adjacent nodes. This will greatly reduce the probability of a Sybil attack, unless the attacker could calculate the topology around the target node and disguise itself as one of the adjacent nodes. As more nodes join the network, the probability of a Sybil attack drops. In early stages, to avoid a Sybil attack, the addresses of official nodes are listed publicly and embedded into the client. As long as there is one official node in the 3 check nodes, a Sybil attack could be avoided.

Data Deception: If a Storer acquires data from adjacent Storers and works out the correct Merkle route in a very short time at the random check request, it will pass the check and get rewards even though it doesn't store any data. PoST is adopted to reduce the probability of that happening. The deceiving node will be penalized by lowering the credit or be permanently banned from the Sharder Network.

Data Hijacking: When the Storer refuses to provide the last data shard to impede the assembly of data bean objects and holds the shard for ransom. In Sharder Protocol, data is split and stored with multiple replicas. The Storer doesn't necessarily know which one is the last shard. Even if it does, the last shard could be retrieved from other Storers, unless all the Storers with the shards are compromised by malicious attackers, which is still unlikely to happen. The probability of a data hijack will be even lower as Sharder Network becomes even more discrete.

Data Erasure: When the Storer erases the data, believing the storage incentive is too low or is simply not willing to continue storing the data. Multi-replicas are adopted to prevent

data loss. The PoST strategy could be adjusted to ensure a significant part of the reward is paid at the end of storage duration. The most effective method is to penalize the Storer by lowering the credit of the node to reduce future rewards.

4.17 Vision

4.17.1 Data Availability

The CAP theorem states that it is impossible for a storage system to simultaneously provide more than two out of the three guarantees: consistency, availability, and partition tolerance. Specifically let's assume N = replica quantity, W = the write replica quantity required by a write operating, R= the read replica quantity required by write operating. The strategy is to assign value to NWR and get a CAP combination. For instance, Amazon adopts N3W2R2, i.e. when two data replicas are unavailable the affected data will be readable only and no longer writable. We will keep researching and refer to the leading cloud storage providers (Amazon, Facebook, Aliyun) and also optimize our data availability.

To reduce the consumption of computing power and network I/O in data erasure, in addition to the classic RS erasure code, we're considering SIMD acceleration and LRC (Locally Repairable Codes) erasure algorithm, like XORing Elephants proposed by FaceBook and University of California.

4.17.2 Digital Assets Management

In many real estate sales centers there are smart devices that help you open bank accounts and freeze some deposits, this acts as your earnest money as well as proof of your assets. This money remain at the buyer's bank account without actually being paid to the property sellers, meanwhile the sellers ensure the intention of the buyers. However, there's no efficient way to prove the non-banking digital assets. For instance, it's difficult to prove your ownership of the tokens in your trade account. Sharder Protocol provides credible PoA (Proof of Assets) in the Sharder Network through Provers and the evidence chain based on the traceable and immutable properties of blockchain. We will come up with a complete solution for digital asset management and certification.

With smart contracts, credible digital assets could be automatically exchanged in low trust or lack of trust circumstances. Various methods for asset management and certification could be created. For example, an automatic transferal of assets to a public address when the time or conditions defined in the smart contract are triggered (analogous to zero-knowledge proof, which proves the ownership of the digital assets address).

4.17.3 Sharder File System

The Sharder File System (SFS) will be upgraded on top of CloudAqua to increase the traffic of single nodes and databases, and to allow concurrent multi-process read-write. It also improves the IO performance of numerous fragmented files. SFS is compatible with common log file systems such as Ext4, HFS+, and NTFS so it is easier to be deployed on nodes in various operating systems or physical environments.

4.17.4 Artificial Intelligence

As with the hardware development of the past few years, AI is greatly improved in supervised learning, antagonism network, etc. Blockchain is a field with innumerable open data and Sharder Protocol constructs a distributed storage network. It is intriguing to research on how to tag, classify the data, and train AI. The continuous AI learning helps Sharder network become smarter, secure, and efficient. In short run, it is expected that AI training could help improve the security strategies and facilitate

Watchers to better “watch” and “adjust” the Sharder Network; and predict and intervene in the potential malicious attack. This helps Sharder become an autonomous and smart network.

5 Sharder Chain

The very first commercial public chain that deploys Sharder Protocol, the number 0 Sharder Pool (Sharder-Pool$_0$), and the cornerstone of the Sharder Network. All functions of the Sharder Protocol will be tested and deployed in the Sharder Chain. Meanwhile, as a commercial public chain, Sharder Chain has some other characteristics such as a user-friendly account model, digital assets, guaranteed transactions, customized API, and operating support system.

5.1 Nodes and Network

The Sharder Chain adopts an upgraded Kad protocol to construct a peer-to-peer network. In order to rapidly build a stable network with enough full nodes, Sharder will release low-power-consumption micro node miners (Sharder Hub) and all-in-one storage-mining miners (Sharder Box).

Sharder Hub not only can conveniently connect idle disk capacities to the Sharder Network, but also ensure more stable online time. With the embedded client, the configuration-free Sharder Hub could start to mine and share storage right out of the box. Sharder Box could get multi-fold rewards for sharing storage space and for generating blocks and while working as Watchers.

5.2 Function Model

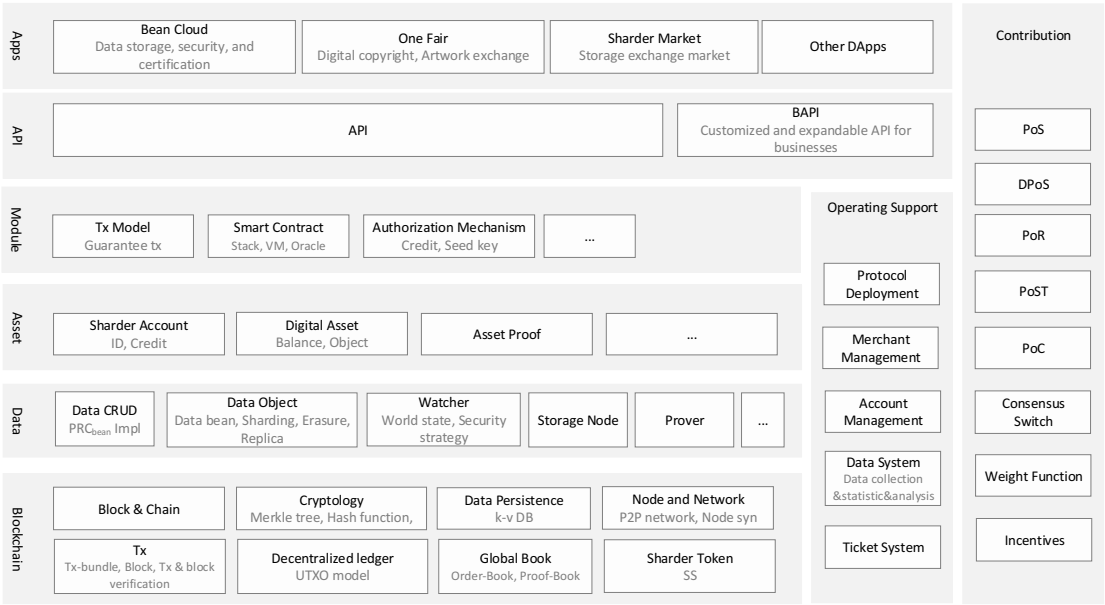

Figure 10 Sharder Chain Function Structure

Blockchain Layer: Comprises of necessary blockchain modules including peer-to-peer network, UTXO model, distributed ledger, global book, protosomatic Sharder tokens.

Data Layer: Implements the data operating, data sharding, replication, and roles of Watchers and Provers as defined in the Sharder Protocol.

Asset Layer: Constructs user-friendly Sharder account models, connects token and data objects with accounts to form digital asset models, and provides digital asset management.

Module Layer: Abstracts and packages basic modules and provides various transaction models based on smart contracts.

Interface Layer: Facilitates the usage of blockchain and distributed storage service.

Contribution Quantification: Quantifies the contribution of various roles to the Sharder Network. Different roles have different quantification formulas. Contributors will be rewarded and malicious behavior will be penalized. Contributions are associated with Sharder accounts.

Operating Support: Facilitates businesses' access to the Sharder Network and provides the businesses with statistics and analytics to improve operating quality.

5.3 Sharder Account

Sharder accounts are no longer discrete addresses (managed with private keys). Instead, mobile phone numbers and email addresses will be used to identify accounts. We will continue to use the traceable and auditable UTXO model of Bitcoin. Account models will be introduced on top of the UTXO model.

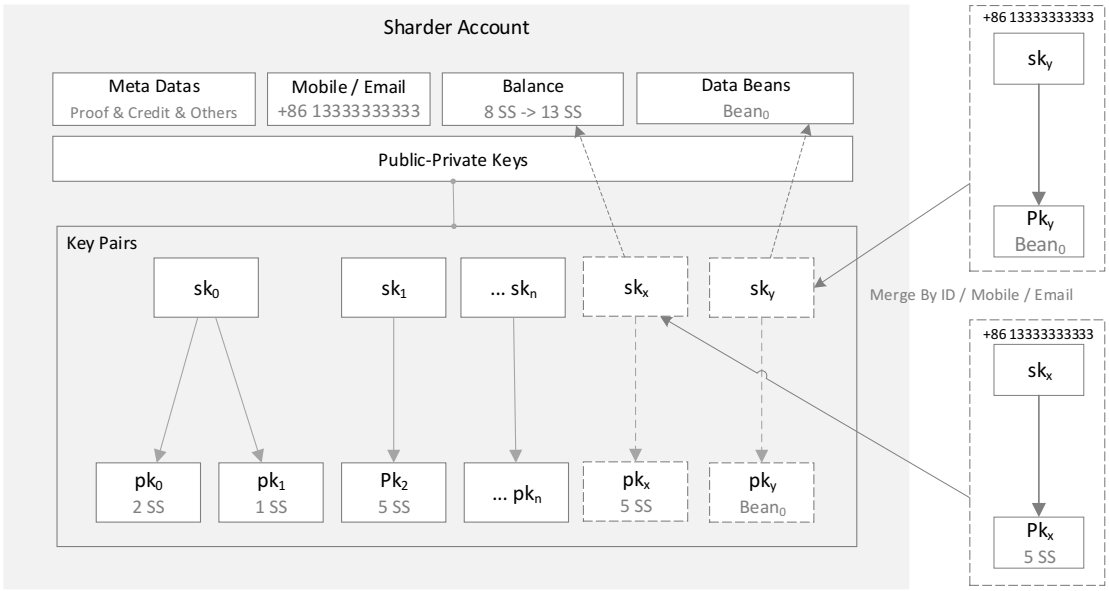

Figure 3 Sharder Accounts

The public and private keys of Sharder accounts are analogous to the HD wallet of Bitcoin. The seed is responsible for the security of Sharder accounts and the private keys of seeds are held by users (a set of signs by BIP44 will be provided to facilitate memorization). The Sharder account model automatically identifies the users and merges the digital assets with the account balance (without the confusing address change of Bitcoin's UTXO model). As long as the seed is securely stored, the account is safe. The key is created by an elliptic curve algorithm [13] and the signature is implemented by EC-KCDSA [14].

This account model requires full nodes with high performance, large capacity storage, and a stable online status to scan all the chains and acquire account information. Watchers could provide efficient data indexing and caching service.

5.4 Digital Assets

Predictably, various physical assets will be digitalized. Sharder Protocol provides the corporations and individuals with distributed data storage and digital asset management service.

The digital assets in Sharder accounts could only be managed and exchanged in the Sharder Network. The exchange of digital assets with the external world still require security exchanges, developing digital asset exchanges, or agents. Sharder Chain will cooperate with the centralized or decentralized exchanges and cross-chain market-making protocols to manage and exchange digital assets.

5.5 Guaranteed Trade

To facilitate the convenient trade of DApps and businesses, the Sharder Chain packages guaranteed trade models on top of smart contracts. It is notable that the “trade” here is not identical to the “transaction” in blockchain systems. Sharder Chain’s guaranteed trade adopts smart contacts as substitutes of intermediate endorsers such as Paypal or Taobao.

Guaranteed trade will automatically create a smart contract, which will freeze trading assets (SS, digital assets, other assets, etc.) on the seller’s address. When buyer pays the agreed amount of SS to the seller’s address, the smart contract will transfer those assets to the buyer’s address. Currently, the auto-trade digital assets is limited to those on Sharder Network. As more Provers connect to the network, the tradable digital assets of guaranteed trade will increase.

Pre-authorized Guaranteed Trade: Both trade parties pay some amount of SS as a deposit. In the case of an auction, all deposit addresses and asset addresses will be frozen by smart contracts, and the statuses will be synchronized to the entire network. Smart contracts hold time-limited private keys (TPK) of both buyers and sellers, similar to the pre-authorization of credit cards.

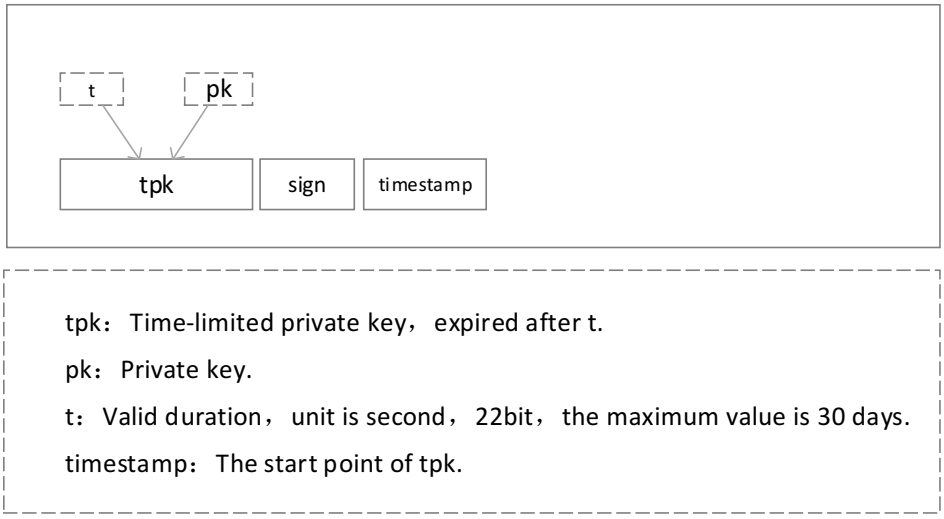

Figure 4 Time-limited Private Keys (TPK)

When the trading is done, smart contracts will automatically utilize the TPKs to transfer the tokens and digital assets ; otherwise the trade will be canceled and the addresses will be unfrozen. In case of malicious bidding, the deposit will be confiscated and the PoC of the account will be lowered.

6 Sharder Community

The Bitcoin community greatly contributed to the operating of Bitcoin since Satoshi Nakamoto published his essay on Bitcoin. We can imply there would be no Bitcoin without the community. Therefore we also hope to continuously improve the Sharder Protocol with the support and wisdom of the community.

Sharder Council: Comprises of Sharder Foundation members, cryptocurrency professionals, and community members; takes charge of the community operations, advocates discussions

and opinions, and manages the community reserves.

Online Platform: Includes: the official website, Telegram group, Sharder community, official QQ group, official WeChat group; where members can freely express their opinions.

Reward Rules: The Sharder Council records, audits, and announces the contribution reward of the community. For the time being, the reward is calculated by formula S=N%×M+50×N, where S is the rewarded SS quantity, N is an integer factor （0<N≦5）, M is the original token balance on the airdropped addresses (M≥100). The Sharder Council will continually revise and improve the reward rules and give a monthly report on their work and usage of reserves in the “Sharder Community White Paper”.

Vision: We believe the development of Sharder Protocol relies on the participation of blockchain enthusiasts in programming, inspection, and testing of Sharder codes. It also needs public chains, corporations, and individuals to test the Sharder Chain. We sincerely hope the community could promote the Sharder protocol, provide feedbacks, and build a free and open Sharder community together with us. Please find detailed rules on operating and rewards in the “Sharder Community White Paper”.

7 Applications

Long-tail customers, high frequency trading, and high frequency usage revolutionized and drove the innovation of China's top IT companies. We firmly believe that long-term development is only possible if more corporations, users, and networks deploy Sharder Protocol.

We believe the decentralized, immutable, traceable, and permanent online status properties of blockchain will greatly benefit the fields of: public welfare, internet of things, supply chains, or the share economy. We have been collaborating with our partners in the research and development of the following blockchain-based business applications:

7.1 Bean Cloud

A data storage, certification, and security platform that services P2P finance, small loans, consumer finance, e-commerce, ERP system, etc. It stores data such as e-contracts, payment documents, and investment records on-chain and provides security certificates and legal evidence for the data based on blockchain’s traceable and immutable features.

7.2 Sharder Matrix

An application that stores personal biological data including genetic information, growth log, medical records, etc. A bold conjecture is that it’s possible to store data like thoughts and memories in the future. As the data accumulates, a personal Sharder Matrix will take shape.

7.3 Sharder Brain

With the development of artificial intelligence, smart devices, internet of things, and the breakthrough of unsupervised learning, we’re convinced that Sharder Brain is able to provide individuals and corporations with smart data services including: data security, data distribution adjustment, data analysis, data search, data alert (data security alert, vital sign alert), etc.

7.4 One Fair

The free market based on Sharder Chain and Sharder Protocol will eventually form a personal data fair – One Fair, where the transparent, open, free and peer-to-peer exchanges take place. Tradable objects include: storage space, digital assets, certified data, valuable information, and etc. For instance, individuals could sell their vital data to

medical research institutes. We believe that idle data will devalue like cash and One Fair will eventually make it possible for your data to circulate efficiently.

8 Development Planning

8.1 Road Map

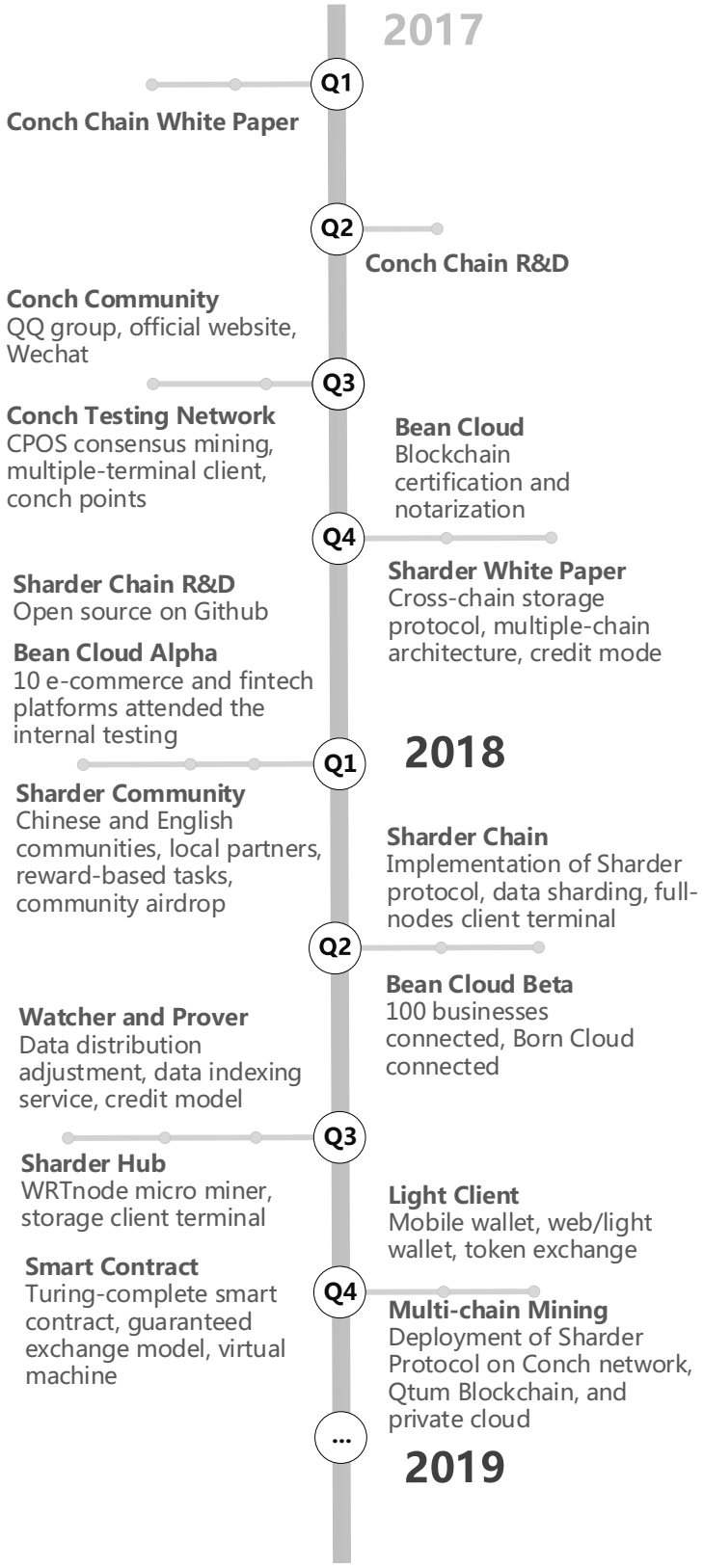

Figure 13 Sharder Road Map

8.2 Profit Model

The Sharder Foundation operates the open-source Sharder Protocol in a nonprofit way . However, in order to continuously invest in iteration, R&D, and operating, we will develop closed-source and premium commercial applications based on Sharder Chain and will profit in the following ways:

Profit Model	Description
Token Appreciation	The improvement of the Sharder ecosystem will reflect the intrinsic value to the appreciation of the tokens.
Storage Trade Commission	When the volume and turnover of storage trading reaches a specific level, we will start charging users Sharder tokens as commission, which will be used to maintain the Sharder Network.
Dapp Service Fee Bean Cloud, Sharder Matrix, Sharder Brain, One Fair	Bean Cloud will charge businesses an annual service fee, which will be a decent source of income as businesses accumulate.
Technology Service Fee	The Dapps charge businesses fees for customized technology services such as smart contracts and trade models.
Advertising Fee	Sharder Chain will start to posting advertisements and charge usage fees when the B terminal and C terminal users accumulate to some extent.

9 Acknowledgement

I am grateful for the revision advice from Xia Zhang, Xinrong Zuo, Aiping during the composition of this paper. I also appreciate the time and effort that Fan Wang put in researching and testing on the Reed-Solomon algorithm. This paper refers to and learns from the design of the distributed Web system IPFS [15] and distributed cloud storage Storj [16] and their codes on Github, for which we're also grateful.

Reference

[1] I. Baumgart, S. Mies. S/kademlia: A practicable approach towards secure key-based routing, (2007). http://www.tm.uka.de/doc/SKademlia 2007.pdf.
[2] Wiki. Erasure Code. https://en.wikipedia.org/wiki/Erasure_code
[3] James S. Plank*. A tutorial on reed-solomon coding for fault-tolerance in raid-like systems, (1996). http://web.eecs.utk.edu/~plank/plank/papers/CS-96-332.pdf.
[4] James S. Plank. Tutorial on Erasure Coding for Storage Applications, (2013)http://web.eecs.utk.edu/~plank/plank/papers/2013-02-11-FAST-Tutorial.pdf
[5] Wiki. Reed–Solomon Error Correction. https://en.wikipedia.org/wiki/Reed–Solomon_error_correction
[6] R.C. Merkle. Protocols for public key cryptosystems, (April 1980). http://www.merkle.com/papers/Protocols.pdf
[7] Zcash Blog. Explaining SNARKs. https://z.cash/blog/snark-explain.html
[8] CPOS. Conch Chain. http://www.conchchain.org/
[9] Bitcoin. bip-0039. https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki
[10] Bitcoin. bip-0044. https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki
[11] Ethereum. Eips. Standardizing HD wallet paths for Ethereum Standard Tokens. https://github.com/ethereum/EIPs/issues/85
[12] University of Southern California & Facebook. XORing Elephants: Novel Erasure Codes for Big Data. https://arxiv.org/pdf/1301.3791.pdf
[13] Yung, M., Dodis, Y., Kiayias, A., Malkin, T., & Bernstein, D. J. (2006). Curve25519: New Diffie-Hellman Speed Records. In , Public Key Cryptography - PKC 2006 (p. 207).
[14] KCDSA Task Force Team. The Korean Certicate-based Digital Signature Algorithm. http://grouper.ieee.org/groups/1363/P1363a/contributions/kcdsa1363.pdf
[15] IPFS. https://ipfs.io/
[16] Storj. https://storj.io

Appendix

Appendix A Definition of Network Operating

1. PING – Online test on nodes
2. STORE – Store KVP at DHT
3. FIND NODE – Return the closest K nodes from the request key-value in the bucket from DHT.
4. FIND VALUE - Return the key-value from DHT

Appendix B Definition of Data Operating

1. PUT – Store data
2. GET – Retrieve data
3. WATCH – Check and adjust data
3.1 SETUP – Set up Initial configuration for check code generation
3.2 PROVE – Generate proof
3.3 VERIFY – Verify proof
3.4 REPAIR – Adjust data distribution

Appendix C Definition of Transaction Operating

1. ADD ORDER – Generate transaction order
2. MATCH ORDER – Match transaction order
3. PROC ORDER – Process transaction order
4. REPAIR ORDER – Repair transaction order
5. DROP ORDER – Drop transaction order