File Network Whitepaper

Filenet.io

Foreword

Since the dawn of the internet, and concurrently with the integration of the internet with society, the core values of science, technology and engineering have been to ensure freedom and equality in the online world. These fields focused on maintaining a free flow of information through uninterrupted innovation and technological change.

As early as the late 90s, companies like IBM and Microsoft have recognized the drawbacks of address-based file addressing schemes such as DNS, and accordingly developed content-based addressing schemes. In 2014, Juan Benet designed IPFS (Inter-Planetary File System), a network-addressing protocol to save and share files in a more durable and distributed fashion.

Up until 2017, with the development of blockchain technology, decentralized storage projects such as Siacoin, Storj and Filecoin[1] have emerged, focusing on the mechanics of a fully functional distributed storage system.

Filenet is an incentive layer built on IPFS, and initiated by IPFS community developers. Its goal is to form a business ecosystem, creating a low-cost and highly efficient data distribution network which serves as a technological cornerstone for future decentralized projects.

This network wants to allow “free” (as in free beer) and fair use of storage. Users can obtain storage resources without paying any tokens. This ensures that early developers are free to develop a variety of DApps. To test the development of decentralized document distribution, Filenet uses a token to encourage users to create and share content. Through a “burn” token model and user behavior pattern recognition, the network ecosystem is able to achieve an economic equilibrium and long-term vitality.

Contents

1. Filenet Introduction

1.1. What is Filenet?

Filenet is an IPFS incentive layer to reward miners for sharing their storage and networking resources. Filenet is also a token which powers a distributed certification mechanism. It creates a cloud-level system for content-sharing dedicated to storing and distributing valuable content on IPFS.

1.2. Filenet Technological Background

1.2.1. Filenet Consensus Mechanism

A core element of blockchain technology is the consensus mechanism. Currently, the most commonly used mechanisms include PoW (Proof-of-Work), PoS (Proof-of-Stake), DPoS (Delegated-Proof-of-Stake), and PoC (Proof-of-Contribution).

Proof of Work requires miners to solve complex cryptographic math problems and relies on computing power. The advantage of the system is that it is secure and reliable. Disadvantages are its limited capacity and the possibility of "51% attacks".

The Proof of Stake consensus mechanism selects miners according to how many coins he or she has. An immediate advantage is its low resource consumption. However, it opens itself to a range of attacks, such as nothing-at-stake, and also results in centralization since wealth brings more rewards and more decision-making power.

In DPoS, the majority of people holding voting rights authorize a small number of nodes to act for them. The system's merits are its high efficiency, throughput capacity and concurrency. However, the power is then concentrated in the hands of a few nodes, which is not safe.

Proof of Contribution allocates mining and validating rights according to the contributions made by the nodes. The advantage of this system is that it does not waste resources thanks to its concept of selection based on resources provided to network. A disadvantage is that the calculation of contributions depends on specific scenarios.

In the era of Blockchain 3.0, the consensus mechanisms are to advance under the principles of economy of resources, security focus and scalability, throughput capacity and concurrency.

1.2.2. Directed Acrylic Graph

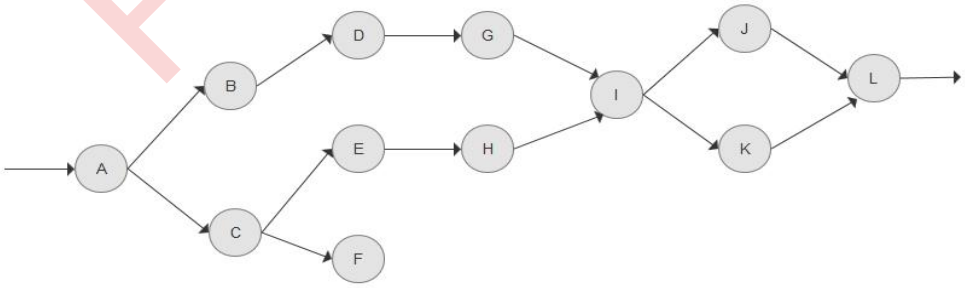

Figure 1: Directed Acrylic Graph

Bitcoin's blockchain structure is a singular-track linked list. This structure was adopted early on in the timeline of blockchain technological progression. It comes with a number of limitations, such as low

block storage capacity, slow transaction speeds, high storage pressure for a single node and a low volume of transactions per second. The linked list structure has a one-to-one sequence, meaning that there is only one valid branch at any given time.

Directed Acrylic Graphs (DAG) are a new type of blockchain data structure. In contrast to regular blockchains, DAGs support one-to-many or many-to-many sequences so long as a loop does not appear. The consistency of the two nodes in a DAG can be verified by the foremost node in any given pair. Therefore, the DAG is able to have multiple branches, allowing a higher level of concurrency and data storage. Meanwhile, DAG accounting is an asynchronous process. The data is weakly synchronized and can accept a certain degree of uncertainty, remaining in a waiting state until it is asynchronously confirmed. The verification time is therefore greatly reduced, and the transaction speed can thus be increased.

1.2.3. Merkle Tree

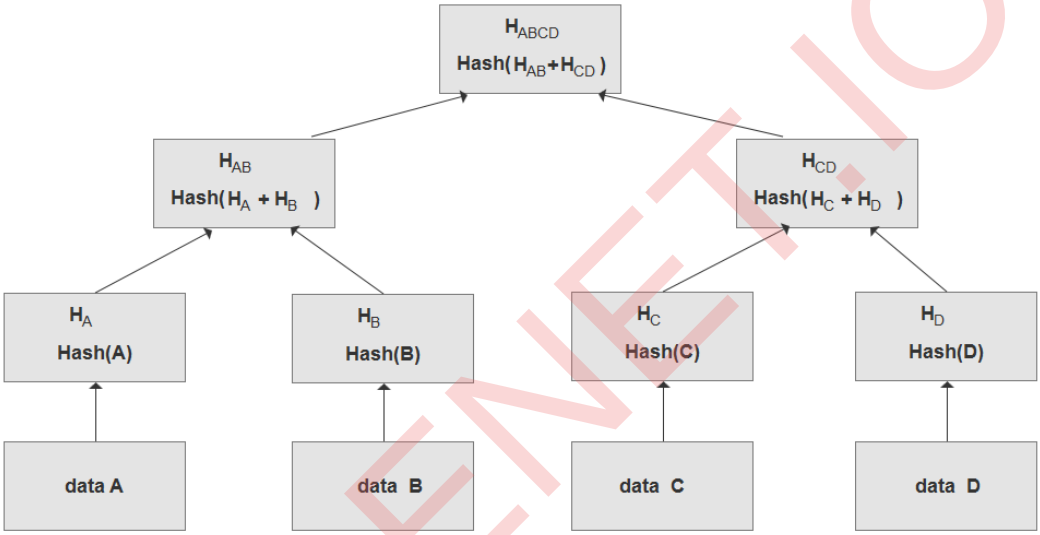

Figure 2: Merkle Tree

The Merkle Tree is a method of validating data. It is used in Bitcoin and is widely used in fields such as peer-to-peer transmissions. In a Merkle Tree, the data is divided into blocks, each computing a hash value. Then, multiple (generally two) data blocks are combined to generate a hash value. Finally, it converges into a root node which creates a tree (usually a binary hash tree). As the data blocks are labelled with hash values, the need to acquire all the data from a big-data or peer-to-peer system for verification is eliminated. Even with a small part of data, we can now verify whether the data is correct or not by addressing its location in the tree and ascertaining the data and hash value in the key location.

1.2.4. Inter-Planetary File System

The Inter-Planetary File System or IPFS is a peer-to-peer distributed file system designed to provide a single file system to all connected devices. In some ways, there are similarities between IPFS and the World Wide Web. It differs from the Web in that the Web is centralized, whereas the IPFS should be viewed as a single BitTorrent swarm using a Git warehouse for distributed repository. In other words, IPFS provides a high throughput, content-addressed block storage model with content-addressed

hyperlinks. This forms a generalized Merkle DAG that aids the building of a versioned file system, blockchain and even permanent websites. IPFS combines a distributed hash table with a block exchange mechanism and a self-certifying namespace. There is no single point of failure in IPFS and nodes do not need to trust each other.

In the future, IPFS will play a huge role in the following three areas: The Web, file storage and blockchain.

Firstly, IPFS will replace the traditional HTTP protocol in the Internet. IPFS has an advantage over HTTP as it can save storage and bandwidth resources. This substantially saves costs and improves the transmission's efficiency. The design of IPFS makes distributed websites more secure, stable and open.

Secondly, IPFS is going to build a global distributed file system that seeks to build a comprehensive repository of documents. This has the potential to include all of humanity's files. With its versioning system, it will preserve every historical version of those files with ease.

Finally, IPFS will be closely integrated with blockchain technology, bringing blockchain to a whole new era. The underlying data of IPFS is perfectly compatible with blockchain data structures. Common blockchains can be expressed as a unified IPFS underlying data structure. Various heterogeneous blockchains can choose IPFS as a storage medium, thus solving the biggest difficulty of blockchain technology – data storage. On this basis, a variety of blockchain products will emerge with enhanced performance and convenience.

1.3. Advantages

Building a decentralized file storage and sharing network is a very attractive and exciting vision [2]. However, there is no built-in incentive mechanism, which begs the questions: Who will provide storage? How do we prevent trash files from consuming system resources? How can these documents be distributed in the most efficient way? If these questions are left unanswered, the network will not be able to exploit all of its potential value. The solution to these problems is a responsible incentive layer.

Today, multiple incentive layers are being proposed. Many of them charge their users for service. Payment is one of several possible solutions to such problems, but is it really necessary to pay? After all, most internet users are not prepared to pay when they are using such services. The internet has already educated people to use most of its basic services for free. Thus, it is impossible to expect users to be willing to pay for browsing a regular web page.

Filenet proposes an innovative incentive layer to settle these problems. The gist of it is as follows:

First, users can upload files to the network freely, but the default files stored cannot be rewarded and distributed. Even if they are being distributed, they will neither be made permanent nor occupy network resources for a long time. In this way, the problem of garbage files is solved.

Second, when a file is retrieved, it will be distributed and saved to more nodes. These nodes will be rewarded in order to tackle the question of who is going to offer storage and distribution resources.

Third, the system is not only free for file uploaders, but will also reward them when the file is looked for frequently enough. Thereby, the cost problem is settled. A payment-free service and a reasonable incentive mechanism are the strong advantages of Filenet. We firmly believe, the advantages of Filenet mentioned above will bring it great practical value and broad space for imagination.

2. Filenet Components and Operating Mechanism

2.1. Architectural Design

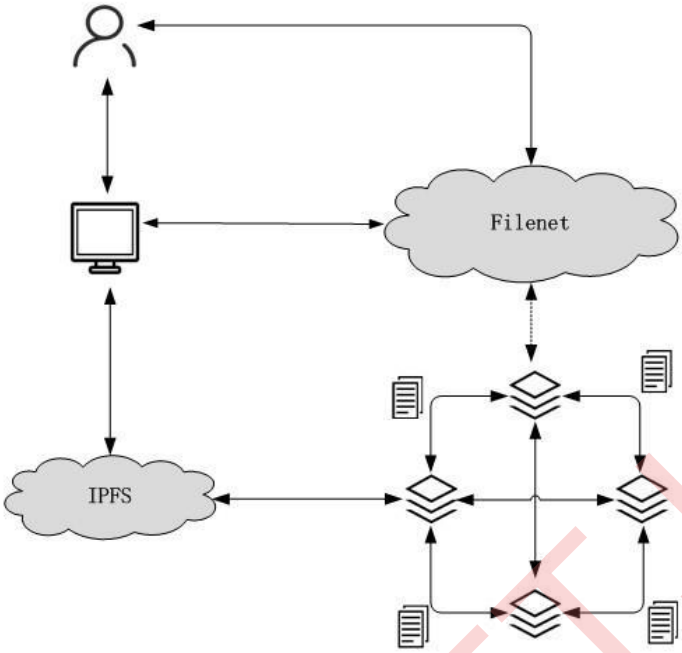

Figure 3: Architectural Schema

Filenet and IPFS combined create a complete product. The underlying functions of distributed data storage, such as P2P Transmission and distributed computing, are accomplished by the IPFS protocol and the “mining” of physical devices.

In IPFS, there are two types of nodes: normal nodes and dedicated nodes. All of them are interconnected through a P2P network. Dedicated nodes are generally private user nodes. When uploading data, the data is first stored in any dedicated node on the IPFS network. This is a data promotion system. In our design, the inflow of data has no token reward, while the outflow of data has one. The outflow destination can either be a DApp or other nodes in IPFS.

Filenet, as the incentive layer, needs to set up block management, a consensus mechanism and smart contracts. With DApps running in IPFS and Filenet, users can upload data directly through Filenet and DApps.

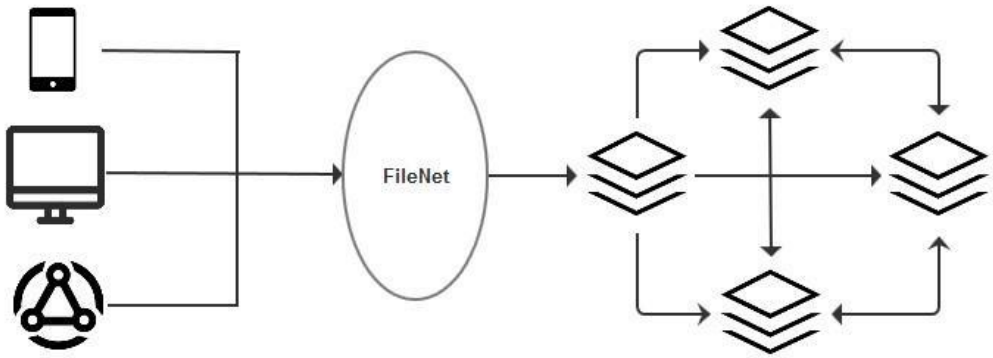

Figure 4: Filenet Storage Management

The Filenet Decentralized File Network (DFN) provides a data promotion system. When a user uploads data through a device, the data is first stored locally. When the data is retrieved multiple times, it gradually enters the open network, reaching a "hot storage" status so it can participate in mining.

2.2. Data Structure

Filenet blocks will include all metadata, resulting in a relatively large volume. Traditional linked lists overwhelm nodes because of redundancy. Conversely, Filenet adopts a Directed Acyclic Graph (DAG) structure, along with Merkle trees. In comparison to traditional linked lists, DAG is more flexible, faster, and greatly improves the efficiency of block building, thus enhances the performance of the Filenet network. Moreover, the Merkle tree only requires some information from key Merkle nodes, instead of requiring the complete block information to verify the blockchain data. Thus, nodes can become lighter, sparing management energy and resources that can be redirected towards scaling the network. At the same time, the Merkle tree can simplify the verification process to further improve network performance. We will detail the specific block structure in the Yellow Paper (forthcoming).

2.2.1. Ledger

The Ledger records users' data, miners' statuses and trust tables. It is a data chain that can be accessed at any time and will keep expanding over time.

2.2.2. Tables

Data Retrieval Table

The data relationships in a retrieval table are very complex. Types of data, data access dates and traffic volume are constantly changing. By saving this data in a DAG, those changing parameters remain segregated.

Miner Operation Status Table

As for data storage and access, the operating status of miners, idle space size and other network environment information will be stored in the current block, and this table will be publicly available.

2.3. Roles

2.3.1. Users

Users upload files through the client and do not necessarily run a node. They can use the Filenet client (DApp) to directly store and retrieve data in a payment-free process.

2.3.2. Servers

The service providers are the miners, who provide resource and data repositories, as well as data retrieval and distribution services to users. They earn revenue through search volume.

Miners store users' data, generate time-based proofs of distribution, and submit these to the blockchain network to validate their activity status. Filenet will reward miners according to the distribution volume in its network. If the miner cannot report their Proof-of-Distribution within the specified time, or if the submitted proof is invalid, they will not receive rewards.

When looking to acquire data, miners have to provide information such as their space, bandwidth and operational status. These parameters will be saved in a table called the 'Miners Trust Table'. Filenet will allocate its data based on a trust factor. Smaller but frequently used data will be stored in miner devices with a stable network environment. Big data with low frequency be will saved in devices that have large space. Filenet will reduce the amount of data distributed if it finds miner device errors or if the environment becomes unstable.

2.4. Protocol

We abstract users and miners into two instances, all of which have attributes and statuses. They both follow the common Filenet protocol rules.

2.5. Mining

Filenet provides storage and replication for high-frequency data in order to prevent low-frequency data from occupying the system's storage resources. All users release their original files through the DApp. Over time, these documents will be retrieved gradually by users, then spread to the Filenet network and eventually arrive in mining devices. The DApp, which could be located on a cloud disk, a social software, or a news client, will register the hash value of files on the Filenet network upon upload.

2.5.1. Overview

The data stored by the user can be categorized into two types: low-frequency data and high-frequency data (matching hot and cold storage). One way in which Filenet mines is to share high-frequency data. The Filenet incentive mechanism helps these data to be spread as much as possible in a life cycle, generating earnings in return, and become a shared source of wealth thanks to the data sharing principle.

2.5.2. Qualification Acquisition and Margin

A server must keep at least one Filenet token in its wallet to qualify as a real account.

We use a 'margin' mode, which encourages servers to maintain their nodes carefully and to ensure stable and reliable node functionality. This could be seen as a mortgage mode, except that servers will not be fined due to a power or network outage.

2.5.3. Profits

The success rate of Filenet mining is proportional to the miners' activity, which is determined as data is downloaded and used by other users. With more usage and higher levels of activity, miners will obtain more rewards.

2.5.4. Hardware

Filenet aims at connecting all idle storage space. Theoretically, all the storage space that can be linked to the internet is capable of being involved in mining. This includes cloud services, data service centers, computers, notebooks, mobile phones, and even mobile terminals like car computers and intelligent hand rings. They can share their data, participating in the mining process and in doing so, generating revenue.

3. Consensus Mechanism

3.1. Proof of Retrieval and Distribution

Proof-of-Distribution (PoDt) is a new type of proof. This choice lowers the threshold of technology implementation as it no longer requires a complex mechanism to prevent attacks. The Proof-of-Distribution verifies the frequency and volume of data distribution. Integrating Proof-of-Distribution, in conjunction with mining, is therefore a sound scheme.

Filenet provides two additional types of proof to validate miners' activity and reflect their efforts accurately, thereby checking miners' service ability. These are the Proof-of-Retrieval and Proof-of-State models.

Proof-of-Retrieval, or PORE, is a protocol that sums up miners' frequency of data distribution, which indirectly reflects the CDN consumption and periodically updates it. This method is one-direction and fails to examine a miner's actual state. These protocols rely greatly on Zero-Knowledge Proof, in which a prover (the one being verified) can make a verifier believe that an argument is correct without showing him any useful information.

3.1.1. Pore Protocol

Users send data to the Filenet network. Then the Filenet tracks, accesses and distributes the data to miners who will store the data. The following is how Filenet verifies the retrieval:

1.Send:
INPUTS:
Prover Key pair（M，D）
(The public key generated by the prover's key to the data parameter)
Prover send key Pksend
Parameter P, (the miner's data parameter) Date
D (data provided by the miner)
Relation P&D→N

OUTPUTS:
Replica R (a replicate of data parameter P)

DAG Root rt of R (hash root generated by a replicate using DAG)
Proof πseed (a proof of replicate distribution)

Procedure:

1. compute hash of data HN: =CH(N);
2. seal R：=seal（N,skM）；
3. compute hash roof of R rt：=DAG CH(N);
4. 》X：=（pk HN，rt）；
5. 》W：=（skP，N）；
6. compute proof of distribution of R πsend：=SCIP.prove（pkseed，X，W）；
7. output R rt πseed
8.

Prove
INPUTS:
Prove Proof-of-Distribution key pkPoDt (public key of data distribution)
Replica R
Random challenge C (check random numbers, representative of hash children nodes)
OUTPUTS：
πPoDt (proof of distribution)

3.2. Trust Assessment

Filenet network selects miners for tasks such as data retrieval, data distribution and block packaging based on each miner's individual trust level. To make sure that miners of various grades all stand a chance to get selected and that trust can come into play, miners are graded according to trust.

A preliminary election will be hosted in the light of trust grades. The higher a miner's trust grade is, the more likely he is to get elected. After the initial election, mining devices with the same trust grade are waiting to be chosen with equal probability. The formula for computing miner's trust is shown as follows:

$$T=(1 - e^{-cm}) * 100\%$$

T: Trust
m: Number of tokens held
c: Regulatory factor

Figure 5: Trust Curve

The decentralized storage network of Filenet can use alternative consensus protocols, so long as the proof-of-distribution is verified.

3.3. Estimation of Miners' Computing Power using PoDt

A miner, Yn, has to submit a Proof-of-Distribution (PoDt) to the network every time to be included in each block cycle. After being verified by the majority of the network, this Proof-of-Distribution is added to the blockchain. Afterwards, all nodes in Filenet update the miners' spatial information and delete invalid records. In this way, the computing power of miner Yn can be determined by summing up and testing his distribution history. For large nodes, the computing power of miner Yn can be calculated by summing up all distribution records of Yn from the genesis block to current ones.

On the other hand, smaller nodes can fetch information from reliable big nodes. Those distribution records of miner Yn come to be sorted in a hash tree, from the newest block header to the genesis block; the Merkle path is a guarantee of presence and integrity. In this way, a small node can delegate the verification of PoDt to the network.

If the miners were to maliciously falsify their computing power, they would need to forge the Proof-of-Distribution, which requires a data transfer order. While this order is being run, a false proof should have been sent to the blockchain; because of the verification requirement during this process, if this transfer finally succeeds, then Filenet will consider the proof to be true. Note that Filenet depends solely on the PoDt protocol to maintain its authenticity, and note further that the security of PoDt has already been proven in the Proof-of-Distribution scheme.

Filenet reaches a consensus using computing power, combining PoC (Proof of Contribution) and PoS together with trust factor, margin deposit and the elapsed time being turned into stakes. Hence, it should be viewed as an improved PoS. By adopting PoC and PoS, Filenet probabilistically selects several mining devices to create new blocks, in each cycle, and trust points will be awarded to chosen miners.

PoC + PoS offers a pseudo-random, unpredictable selection, whose goal is to create blocks within a block cycle and broadcast to the whole network. Since the data blocks are stored in a DAG, the more blocks created, the more secure main chains.

3.4. Consensus Mechanism Algorithm

T：the settlement time of a block cycle
Random（t）：a random number
P^t_J：the miner's computing power in a block cycle
L：H（Yn）is a hash function. L is the length.
<Yn>msn: signature generated by miner Yn to message msn
<Yn>msn=（（Yn）SIGmsn（H（Yn））） 1
Mine：Yn
Compute
H（<t||random（t）>Yn）*2L≦Pt/∑j Pt ----------2
π=<t.r>i
verifier Node： Verfy(π, t, Yn) 3

1.π is valid signature
2 P^t_J： is Power of Yn at time t
3. check H（π）/2L= Pt/∑j P^t_J

The miner uses the inequalities above to confirm whether he is selected to create new blocks. In the foregoing Formula 1, a random number is obtained by hashing the current block time. Miner Yn signs this hash. By doing so, he acquires a second hash: hash2. Then, hash2 is divided by the Lth power of 2 to get a number between 0~1.

Formula 2 calculates the ratio of the miner's computing power to that of the whole network.
This consensus mechanism is therefore fair.

Each participant has only one electoral opportunity within a block cycle and all of them are randomly selected. The random (t) function is unpredictable, in that computations cannot run in parallel or be manipulated maliciously.

A malicious miner is not able to forge other miners' signatures since they rely on private key-based authentication.

At the same time, it can be publicly validated. If the mining machine calculates that it is the winner, it will be required to submit the verification evidence to the entire network. This allows for the verification of the calculation power of the mining machine and the random number hash value it generated to occur at any time.

4. Smart Contracts

Filenet is a public chain for developers. It features special programming primitives for DApps to easily interact with stored data. These primitives are included at the EVM level. In a contract, it is therefore also possible to access information regarding data location, storage nodes, and miners.

Smart contracts produced by us will allow miners to hold their own tokens. We will use the Ethereum Virtual Machine and Solidity for fast implementation of these contracts.

Filenet has the potential to implement this smart contract mechanism, and we believe that future versions of EVM and WASM will natively integrate Filenet functions to allow other chains to take full advantage of Filenet.

5. Ecosystem Development

Up to now, a great number of blockchain projects are developed based on Ethereum; but these same projects are at a loss when faced with the big data concurrency and large storage requirements of some applications such as decentralized videos, games, live broadcast platforms, and diversified IoT applications. Under these circumstances, Filenet puts forward effective solutions thanks to its technological features.

We are now programming this public chain in the hope that developers can easily develop their own applications and achieve their dream of changing the world. Our chain will keep optimizing its programming code and providing the necessary storage space and network for DApps to run.

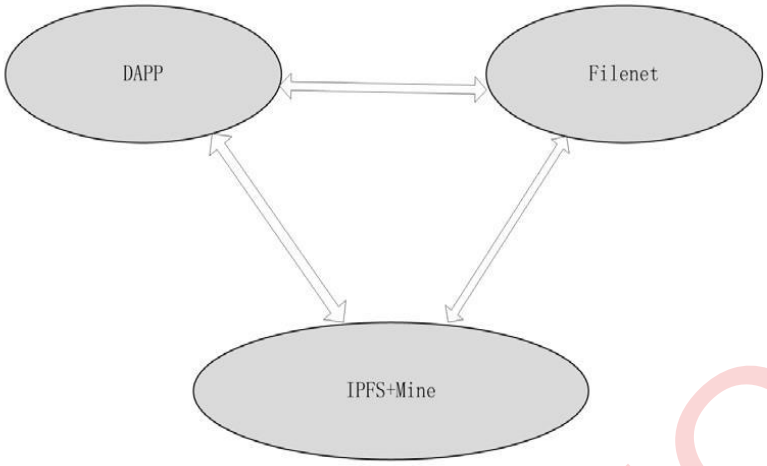

Figure 6: Ecosystem Development

6. Future Developments

This document, the Filenet Whitepaper (First Edition), presents our present strategy and technological plans. The technical Filenet Yellow Paper will provide more details about subsequent stages of development.

The Filenet.io Project will feature a community-centered development approach. Subsequent technical issues will be publicly collected in Filenet.io or on our related communities. Any developer is welcome to participate if he or she is willing to join and dedicate some time. Our github repository will be open and provide a convenient way for anyone to join hands with us.

We convey our gratitude to our friends from the IPFS developer community for their technical advice. Currently, Filenet.io is our only official announcement channel. Please visit our website Filenet.io for more information and follow us on our path.

References

[1] Filecoin: A Decentralized Storage Network, Protocol Labs.
[2] IPFS - Content Addressed, Versioned, P2P File System, Juan Benet.
[3] Disrete Mathematics and Applications Seveth Edition,Chinese Abridgement.
[4] Computer Networking A Top-Down Approach Sixth Edition, James F.Kurose Keith W.Ross.
[5] Peter J. Braam. The Coda Distributed File System, School of Computer Science, Carnegie Mellon University, http://www.coda.cs.cmu.edu.
[6] Lustre File System White Paper http://www.sun.com/software/products/lustre/index.xml.
[7] The Google File System, SOSP'03, Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung.
[8] Network File System, http://www.faqs.org/rfcs/rfcl094.html.