Information Processing and Management 58 (2021) 102627

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

A taxonomy for Blockchain based distributed storage technologies

Omer F. Cangir *, Onur Cankur, Adnan Ozsoy

Department of Computer Engineering, Hacettepe University, Beytepe, Ankara, Turkey

ARTICLE INFO

Keywords:
Blockchain technologies
Distributed storage
Taxonomy
Blockchain based storage

ABSTRACT

In recent years, the amount of data stored in computer environments has increased significantly. The growth in volume has made it very difficult to store and process large amounts of data on a single server. Distributed storage technologies are being used as a solution to problems such as scalability and high availability on a single server. Distributed storage technologies evolve the process of handling and serving data on multiple nodes by using related administrative methods. Later, new storage infrastructures were developed by performing distributed storage strategies on the new blockchain technology. However, the literature lacks a methodology for examining solutions that utilize this new blockchain based distributed technology. In this article, we propose, to our best knowledge, the first categorization and taxonomy of blockchain based distributed storage technologies. The state-of-the-art solutions are examined, compared and evaluated using the proposed taxonomy.

1. Introduction

Data storage is one of the most fundamental requirements of the computer era. The storage infrastructure is evolving using new technologies over time. Increase in data volume has led to technological changes. Firstly, studies on retention of the data have emerged. In these studies, it is aimed to keep the data in a way to cover less volume. Many compression methods have been applied in this field (You & Karamanolis, 2004). In spite of these studies, the rapid grow in the data volume revealed alternative solutions.

To address the growth in data volume, network-based storage technologies are being used instead of direct attached storage (Gaonkar, Bojewar, & Das, 2013). Direct Attached Storage(DAS) method is very fast but not enough to handle larger data (Padhy & Patra, 2012). To store larger data, Network Attached Storage(NAS) technologies based on TCP/IP protocol were introduced to enable communication over ethernet ports (Katz, 1992). Because of the TCP/IP overhead, NAS storage infrastructure handles input and output operations slower than DAS does. In addition, the entire infrastructure will be affected in the case of a failure in the network or in any of the storage nodes. Therefore, the NAS network as a whole entity which storage nodes are located can be affected by the single point of failure problem. Therefore, NAS storage infrastructure is not sufficient due to fault tolerance weakness on NAS nodes.

To overcome NAS related problems, faster Storage Area Network (SAN) technology has been introduced using Fiber channel protocol (Clark, 1999). The SAN infrastructure is faster and more fault tolerant (Padhy & Patra, 2012). It allows data to be held on distributed nodes. These specifications are concrete improvements over the weaknesses of NAS infrastructure. Thus, SAN infrastructure is actively used to address large data problems in many commercial environments.

Thanks to the advanced storage technologies, very large volumes of data can be stored easily. However, there are some problems with the use of these advanced technologies. Purchase and maintenance costs of these technologies are very high. In such technologies, high availability is not fully achieved since data is usually located in a single data center. Raid and replication on the

* Corresponding author.
E-mail addresses: omerfarukcangir@gmail.com (O.F. Cangir), onurcankur@hacettepe.edu.tr (O. Cankur), adnan.ozsoy@hacettepe.edu.tr (A. Ozsoy).

https://doi.org/10.1016/j.ipm.2021.102627
Received 27 September 2020; Received in revised form 22 March 2021; Accepted 3 May 2021
Available online 10 May 2021

same storage are useless in disaster situations. In order to manage these problems, disaster recovery mechanisms should be operated. However, the total cost of owning a storage space with disaster recovery mechanisms is excessively expensive (Garcia-Molina & Polyzois, 1990).

Another solution used to store big data is storing it distributedly. Distributed storage systems were introduced to lower the location dependence and cost problems. This type of storage technologies provide higher availability for the whole data (Ford et al., 2010). It is also very powerful in terms of flexibility and performance metrics (Howard et al., 1988). Moreover, many of them are less costly because they can be created using commodity hardware (Wang, Jing, He, Qian, & Zhou, 2010). Because of these advantages, the use of distributed systems is increasing day by day.

Distributed systems can be categorized into two; owned by an authority or open to public joint ownership. Distributed systems controlled by a single entity or central authority raise concerns regarding trust. Google, Amazon, Microsoft and many other big corporate companies have solutions that provide distributed storage. However, your data is managed by these companies and privacy of your data is an important issue. On the other hand, systems which are not owned by a single entity or company can leverage trust issues through the network of nodes which are involved in decision making. One such example is peer-to-peer (P2P) networks. P2P networks have been widely used as a distributed storage system for a long time. While there are many benefits of using these systems compared to non-distributed systems, such systems have some disadvantages as well. One of the most expected features of all storage systems is high accessibility. However, P2P storage systems lack mechanism to strengthen the high availability mechanism, specifically, there is no built-in incentive to keep data holders for longer periods of time. Similarly, there is also no penalty mechanism for the data holders who reduce availability of data. Some studies have focused on this problem (Wang & Vassileva, 2003). In addition to high accessibility problems, P2P network also lack of trust issue between peers (Stodt & Reich, 2020). Any contract between peers running in local environment can be altered. This vulnerability could allow data to be modified by malicious clients. In such infrastructures, technologies that provide solutions to the mentioned problems are needed.

While researchers were looking for solutions to problems on the P2P platform, Blockchain technology emerged as a solution to the stated problems. Blockchain technology overcomes the aforementioned problems of P2P networks, which has become more widely used recently. as it is not managed by a central authority it provides full independence and default incentive mechanism on its own. Besides, it allows smart contract and proof mechanisms to make a more trusted and secure platform available. With recent developments, this technology is used to manage the storage infrastructure. A number of studies have been carried out that have different architectures and allow data to be managed in a distributed manner.

In this study, state-of-the-art blockchain based distributed storage technologies are surveyed comprehensively and compared among each other according to determined categories. We identified projects serving similar purposes and selected whose main purpose is providing storage space. As a novel contribution, a categorization for blockchain based distributed storage technologies is proposed. To make the categorization more accurate, a new taxonomy is presented in accordance with the details learned during this comparison. To our best knowledge, proposed taxonomy is the first categorization specifically for Blockchain based distributed storage technologies. Then each storage technology is evaluated for the proposed taxonomy and each field of the categorization. Additionally, design decision diagrams based on the taxonomy are provided for current technologies.

The rest of the paper is structured as follows. In the Background section, Section 2, the basics of the technologies on which the article is based are given and these concepts are detailed. Related Work section, Section 3, includes studies on different types of distributed storage. In Section 4, the technologies that provide distributed storage services via blockchain are examined with the determined taxonomy items. In the Comparison and Evolution section, Section 5, the selected technologies are compared with their general characteristics and their strengths and weaknesses are indicated. In addition, the general weaknesses of these technologies are mentioned together with the reasons and features that should be developed. Lastly, Section 6 concludes with the discussion of our findings and possible directions for future works.

2. Background

The amount of data around the world is growing tremendously over the years. According to IDC white paper (Reinsel, Gantz, & Rydning, 2017), in 2025, there will be approximately 160 ZB of data created and the increase of the data growth will be more than linear (Fig. 1).

In addition, IBM Marketing Cloud's report (Marketing, 2016) mentions that 2.5 quintillion bytes of data are created every single day. It is also noticed that 90 percent of the created data in the world today has been created in the last two years. Based on these reports, the acceleration of the data growth rate will be even more. All of these statistics show that over the years, the data size keeps growing and there will be a tremendous need for a newer, cheaper, safer, and more efficient way to store the data.

2.1. Distributed storage

One of the ways to store large data is keeping it in a distributed manner. Distributed storage is about storing data on a physically or geographically distanced group of machines (nodes) that can operate concurrently with a shared state and they are independent of each other. It is quite different from traditional storage methods such that instead of storing all data in one machine, distributed storage provides us to distribute data in a replicated manner using more than one machine. This storage method contributes greatly to the storage technology and introduces new features by providing more availability, reliability, and capacity. In this section, how these features contribute to the storage technology and the advantages/disadvantages of using them will be discussed in detail.

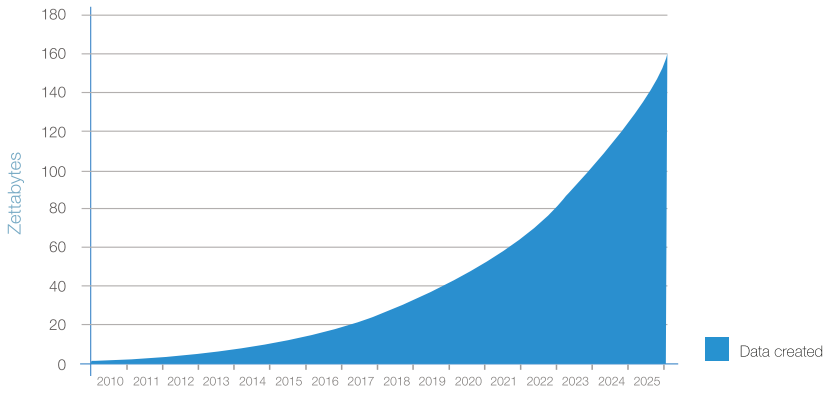

Fig. 1. World Wide Data Volume Growth.
Source: Courtesy of Reinsel et al. (2017).

2.1.1. Distributed storage features

One of the important issues in distributed systems is **availability**. Some nodes might be out of service during a maintenance time, can be powered down because of some kind of unexpected hardware failure, or simply might have received requests more than it can respond. In such scenarios, the system will not be available. However, despite these problems, a distributed system tends to be more available because of consisting many nodes in the system that can satisfy requests. In order to have a **reliability**, a system should function according to the predefined rules/conditions and continue to work for the expected period of time. Additionally, systems must tolerate any fault which is another issue. Faults might be about server and network failures, overloaded servers, or data consistency. A system which can tolerate to these faults is called a system with **fault tolerance**. **Data consistency** is another important issue and a distributed storage system is consistent if every node has the same view of data. Finally, as mentioned earlier, data growth is increasing exponentially over the years and storage of this data requires **scalability** which is about being able to handle increasing capacity and throughput without diminishing performance and data availability of the system. When faced with large number of requests for data, distributed systems can handle single point bottleneck thanks to its flexible structure (Kim, Kwon, & Cho, 2018). Besides, there are some advantages and disadvantages related to these features.

First of all, as an advantage, it makes possible to improve the capacity of the distributed storage system just by adding one more node to the system instead of trying to upgrade the hardware of a single node. That means there will be no limit to increasing the capacity of the system. The other advantage of distributed systems is that it is more fault tolerance. Since there is not only one machine, when one of the machines goes offline or crashes, the system can still work. If a piece of data is distributed to more than one machine, it can be correctly retrieved even if some of the nodes that store the data are not working properly. Therefore, data loss caused by the node failures will be lesser compared to a single-server storage system. That means distributed storage systems are more reliable and have more ability to be available than traditional systems that have a single server.

In decentralized distributed systems, each individual has control over his/her own data. When storing or sharing a piece of data, there is no need for a third party. In order to achieve decentralization of distributed storage systems, there must be incentives for users for keeping the system up. In addition to that, in these decentralized distributed storage systems, it is even more problematic to provide security because there is no central authority to trust. Each individual should care for his/her own data and the system must provide full security for all users. These problems about decentralization can be overcome by using blockchain technology, which enables eliminating the need for a central authority. There are some peer-to-peer distributed file-sharing systems that achieved decentralization like InterPlanetary File System (IPFS) (Benet, 2014), and Metadisk (Wilkinson, Lowry & Boshevski, 2014). They are currently used by other decentralized distributed storage systems which are examined in this paper.

IPFS is one such example that uses blockchain technology to facilitate decentralized distributed storage (Benet, 2014; Daniel & Tschorsch, 2021). The main purpose of IPFS is to make the web distributed. Using IPFS, it is possible to track versions of a file over time and the file can be moved across the network by the users who are on the network, therefore, it is a peer-to-peer, distributed file sharing system and these features make it very different from traditional centralized file systems. IPFS uses content-based addressing instead of location-based addressing and it provides accessibility to data even if the stored location is crashed. FileCoin (Benet & Greco, 2018) which is a distributed storage technology uses IPFS as a file system.

Another example that uses decentralized distributed storage is Metadisk (Wilkinson, Lowry & Boshevski, 2014) which is a prototyping platform for a fully decentralized network and an interface through a web application. In addition to that, they also propose an API for feature extensions and native applications. In Metadisk, there is a cryptocurrency that serves as an incentive for users and a payment mechanism. Moreover, for file metadata, a separate blockchain is used as a data store. It facilitates a P2P network and instead of a central database, this application uses a public blockchain to store information. Metadisk provides a stable testing platform for Storj (Wilkinson, Boshevski, Brandoff & Buterin, 2014) which is a blockchain based distributed storage technology.

2.2. Blockchain

Blockchain, which is the underlying technology for Bitcoin, is introduced in a paper by Nakamoto (2019). Blockchain is a peer-to-peer network where everyone keeps a distributed ledger that contains information about transactions. These transactions are gathered and stored in blocks. These blocks are connected to each other like a chain. It is not possible to change a block after it is added to the network because the change that is made on that block will cause a change in other blocks on the chain as well. Cryptographic hash functions are used to provide this property which makes the chain immutable, append-only, and secure. Since it is immutable, no peer can alter it and the data can be only added to the network because it is append-only. Update request are only granted via consensus protocols among peers, therefore, no central authority is needed. To be able to add a new block to the chain, transactions are verified and built as a block. This is called mining. After finding a valid block, block is broadcasted to the network by the miner. When a miner broadcasts the block, other miners who receive this broadcast is continue to find the next valid block. At the same time, various chain forks can exist in the blockchain. In such case, block information to be permanently added to the chain is determined according to the longest chain rule. In this way, the system continues to function securely in case the malicious attackers do not reach the 51% attack vector. The mining process costs energy and time for the miners and there must be some incentives for miners to keep them mining. For that reason, Bitcoin application provides a price, a token, to the miner for each block added to the network. Tokens can be seen as virtual money that is created by the network and they might have value in real life. In addition to incentives, there is also a penalty such as economic loss for malicious users who are trying to do illegal operations.

Blockchain technology allows building a storage network that is decentralized, distributed, peer-to-peer and cryptographically secure. To facilitate these capabilities, there are several blockchain-based distributed storage projects (Benet & Greco, 2018; Inc, 2019; Vorick & Champine, 2014; Wilkinson, Boshevski et al., 2014) that are examined in this paper and explained in detail in the next chapter.

3. Related work

Distributed storage systems have been developed to store very large data in a much more practical way than central systems. Distributed file systems have been developed to enable distributed storage systems to be implemented. Many distributed file systems have been developed to provide custom solutions for specific areas (Blomer, 2015). For example, Hadoop File System (HDFS) (Shvachko, Kuang, Radia, Chansler, et al., 2010) is designed as a storage layer for the MapReduce framework or Lustre (Pilaud, Halbwachs, & Plaice, 1987) is designed as a storage space for applications on supercomputers. Even if they serve different purposes, all these systems have been developed as a solution to the problem of processing very large data (Depardon, Le Mahec, & Séguin, 2013). As the amount of data increases on a file system, the load of the metadata operations increases. Metadata payload processing is one of the most fundamental challenges on file and storage systems.

The concept of object storage has been introduced to reduce the increased metadata processing load when processing large data. Object storage eliminates the tiered file structure used in file storage, and places everything into a flat address space. File systems that allow object-based storage reduce metadata operations by allowing the client to know where the desired data is located. Many file systems such as Ceph (Weil, Brandt, Miller, Long, & Maltzahn, 2006) and GlusterFs (Boyer, Broomfield, & Perrotti, 2012) support object-based storage.

Advanced distributed file systems can serve users by solving many problems such as addressing metadata load. Distributed storage systems are generally established and operated by a central authority. They are hardly decentralized due to ease of management of the data by a single authority. Centralized systems have inherently some problems. The systems in which the central authority holds all the system management has many risks due to their nature. With the increasing number of users who want to get rid of these disadvantages, the use of P2P systems is becoming more common. With the use of P2P networks, files can be shared directly between systems on the network without the need of a central server. Nowadays, P2P infrastructures such as Bittorrent (Pouwelse, Garbacki, Epema, & Sips, 2005) and Napster (Saroiu, Gummadi, & Gribble, 2003) are widely used.

The Bitcoin article by Nakamoto brought a new perspective to P2P solutions (Gaonkar et al., 2013). In the proposed paper, blockchain technology (Swan, 2015) is defined as the underlying framework for Bitcoin. It establishes trust in a trustless system, coming to a consensus among distributed peers, and imposing the incentive to be a part of the system which are the features that traditional P2P networks are missing. As a result of the wide acceptance of the Bitcoin as a complimentary financial tool, blockchain technology has been used in many other applications such as health, insurance, voting etc (Berdik, Otoum, Schmidt, Porter, & Jararweh, 2021; Hardin & Kotz, 2021; Jing, Liu, & Sugumaran, 2021; Pilkington, 2016).

The increasing confidence in Blockchain technology has revealed solutions for the problems of a wide spectrum of domains. Blockchain-based studies have been carried out to find solutions against possible security vulnerabilities (Baniata, Anaqreh, & Kertesz, 2021; Oham, Michelin, Jurdak, Kanhere, & Jha, 2021; Yu et al., 2021). Similarly, many studies have been conducted to strengthen integrity and confidentiality features (Putz, Dietz, Empl, & Pernul, 2021; Zhao, Chen, Liu, Baker, & Zhang, 2020). Contributions have also been made on authentication and authorization issues, which are among the most fundamental points in the information industry (Esposito, Ficco, & Gupta, 2021).

The fact that the blockchain infrastructure has become a very inclusive domain in itself has led to various studies to strengthen its internal structures. In this context, studies have been carried out on incentive mechanisms (Khalid, Iftikhar, Almogren, Khalid, Afzal, & Javaid, 2021). General performance characteristics have also become another research topic (Xu et al., 2021). Similarly, research has been done for transaction detection to strengthen the blockchain infrastructure (Hu et al., 2021).

Data storage and sharing has also become one of these applications. Many data storage services have been introduced using the P2P network infrastructure and powered by blockchain technology. With these developed services, data can be stored and shared in a wide spectrum from IOT data to personal private data (Campanile, Iacono, Marulli, & Mastroianni, 2021; Shafagh, Burkhalter, Hithnawi, & Duquennoy, 2017; Zyskind, Nathan, et al., 2015). In addition, studies have been conducted on auditing the data (Li, Wu, Jiang, & Srikanthan, 2020).

There are several projects that provide distributed storage services using blockchain infrastructure on the P2P network. SiaCoin (Vorick & Champine, 2014), Storj (Wilkinson, Boshevski et al., 2014) and FileCoin (Benet & Greco, 2018), PPIO (Inc, 2019) projects allow users to share storage space using smart contracts. In general, projects that serve similar purposes have different approaches in terms of technological approach.

Distributed storage systems can be examined in many aspects. As with all other special systems (Rimal, Choi, & Lumb, 2009a), distributed storage systems must have their own specific taxonomies. In distributed storage and cloud storage, there are several works (Kächele, Spann, Hauck, & Domaschka, 2013; Rimal, Choi, & Lumb, 2009b; Thanh, Mohan, Choi, Kim, & Kim, 2008). In Thanh et al. (2008), a taxonomy on distributed file systems is given. The paper provides a findings section to survey existing distributed file system implementations. Rimal et al. (2009b), the authors provide a taxonomy for cloud computing architectures and then use this taxonomy to identify similarities and differences of the selected systems. Similar to Thanh et al. (2008), only a findings section is provided. In Kächele et al. (2013), authors give a taxonomy for computation, storage and network services of cloud architectures. Different than the aforementioned papers, in this paper decision diagram is used to categorize architectures from the perspective of the proposed taxonomy.

The blockchain based taxonomy work is limited in the literature, but still there are several papers that are of interest of this paper. Xu, Weber, Staples, Zhu, Bosch, Bass, Pautasso, and Rimba (2017) provides blockchain-based systems for architecture design. The taxonomy can be used in general design decisions but not directly addressing distributed storage. Tasca and Tessone (2017) gives a comparison among the widely known blockchain technologies and discusses the differences. There are several survey papers that also give taxonomy (Benisi, Aminian, & Javadi, 2020; Bouraga, 2021; Labazova, Dehling, & Sunyaev, 2019; Mohsin et al., 2019). (Benisi et al., 2020) gives a survey on blockchain based distributed storage networks with a general description on blockchain and storage networks without taxonomy or classification. (Mohsin et al., 2019) provides a survey and taxonomy specifically on the authentication of network applications where (Bouraga, 2021) specializes on blockchain consensus protocols. Although there is not a taxonomy on blockchain-based distributed storage technologies, Placek and Buyya (2006) have proposed a taxonomy that can be used in distributed storage systems. They proposed that distributed storage systems can be categorized in terms of System Function, Storage Architecture, Operating Environment, Usage Patterns, Consistency, Security, Autonomic Management, Federation, and Routing and Network Overlays. Although these substances generally include distributed storage systems, a taxonomy for the categorization of blockchain based distributed storage infrastructures is missing in the literature which this paper aims to fill the gap in the literature.

Current literature lacks a deeper level grouping and ordering instrument for examining and comparing blockchain based distributed storage projects. Thus, a better classification and categorization is required to fully judge and understand the different solutions in this area.

4. Blockchain based distributed storage technologies

In this section, four distributed storage projects using Blockchain technology are examined in detail. As we have mentioned in the literature review, Placek and Buyya (2006) have proposed a taxonomy that can be used in distributed systems, where they proposed that distributed storage systems can be categorized in terms of System Function, Storage Architecture, Operating Environment, Usage Patterns, Consistency, Security, Autonomic Management, Federation, and Routing and Network Overlays. However, these terms come short to specifically address the categorization of blockchain-based distributed storage infrastructures. We have made various classifications on all projects to address their differences and similarities. After we examine the detailed structure of existing projects, we identified six different categories to compare blockchain-based solutions. These categories are maturity, blockchain usage, security, consensus mechanism, redundancy/fault tolerance mechanism, and decentralization level. Regarding blockchain based technologies, maturity is essential since many companies emerge in this decentralized domain and several fake companies cheated people and stole their money. How blockchain is used is another key feature of the involvement of peers in decision making. Security, consensus mechanism, and fault tolerance give key insight into the system and help us to categorize the blockchain based systems. Finally, decentralization level from the chain and data storage perspective is a distinctive feature change from system to system.

Maturity has been revealed as a metric, which is used to determine how close the project to achieve its objectives. The maturity metric examines whether the project has sufficient documentation such as existence of technical and white papers, whether it is an open source project, works at production level and supports object storage. The purpose of using Blockchain technology and the technical details used in projects are evaluated under **Blockchain Usage** category. This metric seems useful as different projects try to strengthen different points through blockchain. The **Security** category is introduced to address the details of the project security level. Many different checks can be done under the security metric. Since we could not reach sufficient data, we could not make a detailed comparison on this subject. Security category is also introduced to address the details of the project integrity and confidentiality level. These domains will be referred as security in the rest of the paper. Many different checks can be done under the security metric. Since we could not reach sufficient data, we could not make a detailed comparison on this subject. The **Consensus Mechanism**, which is revealed by different characteristics due to the nature of the Blockchain infrastructure, is also

proposed as a category. This metric makes it possible to make accurate comparisons among the consensus mechanisms used. **The Redundancy/Fault Tolerance** mechanism is used to determine the resistance to failure and addresses the methods used for this purpose. This metric is important as it directly relates to the availability of data. Finally, **Decentralization Level** category is used to address projects in terms of the level of ownership. Under this we examined decentralization based on data and chain.

4.1. SiaCoin

4.1.1. Maturity

SiaCoin is a platform for decentralized storage which enables the storage contracts between peers (Vorick & Champine, 2014). Instead of renting storage space from centralized providers, SiaCoin allows renting storage between peers. It uses an extended version of Bitcoin scripting mechanism. Storage client or host does not need to know anything about the background technology. SiaCoin handles all the technology features and provides users a simple user interface.

SiaCoin project has a whitepaper with the technological infrastructure details. It is also open sourced. In addition, it is currently serving at the production level. With the help of 3rd party software, it is also compatible with the de facto object storage API used by commercial cloud providers. Considering all these parameters SiaCoin has a high level of maturity.

4.1.2. Blockchain usage

SiaCoin has a mechanism to keep storage contracts on the blockchain. Storing all contracts in blockchain, makes it publicly auditable. Data is transferred between host and client machines after the contract agreement completed. Thanks to the extended scripts of SiaCoin project, it is possible to store all necessary contract data in blockchain. These scripts enable the creation and enforcement of storage contracts. SiaCoin uses three types of scripts to provide these features. These types are contracts, proofs, and contract update scripts. Contracts are used to clarify the agreement between a host and a client to store a file. Storage proofs serve to prove that the server holds data. Contract updates allow contract changes to be made according to the requests.

A contract is created after the storage provider agrees to store a client's data. When a contract is created, the client pays the fee to the hosting provider. This fee is locked by the system until all the agreement conditions are satisfied. All contracts are associated with agreement conditions and these conditions must be met before the hosting coins are unlocked. Locking operation is done by signing the necessary transaction attributes. Agreement conditions include a time lock, a set of public keys and the number of signatures. Storage flow can be completed when these requirements are fulfilled. After contract is created, the host periodically submits the proof of providing storage service until the contract expires.

4.1.3. Security

In distributed storage systems, all files are divided into several pieces and all these pieces are stored on distributed hosts. To provide a security mechanism over file content, each piece is encrypted before uploading. In this way, only the owner can see the content of the related file. This provides a basic level of security support for SiaCoin infrastructure.

The infrastructure used on SiaCoin platform has some potential security threats. Firstly, block withholding attacks, manipulates the random number which used to prove holding the file by the host, can be threatful for the system. Manipulating random number will possibly cause losing block reward for miners. Therefore, financially motivated attackers would not try to do this type of attack. In addition, to increase deterrence, clients can specify a high challenge frequency and large penalties for missing proofs.

The proof of storing data by host is become valid after they are included in the blockchain by miners. Miners could maliciously exclude proof from blocks. Victim host can face to pay a penalty fee . In this situation, malicious miner is able to request a bribe for transaction proof. This type of attack reduces the reliability and security of the system. The strongest defense against such attacks is to use a larger window size. Another defense mechanism is rejecting confusing contracts. Hosts can reject any contracts which they feel vulnerable to this kind of attacks. Host is also allowed to reject any signed contract until any file is uploaded.

4.1.4. Consensus mechanism

SiaCoin uses Proof of Storage as a consensus mechanism. While operating this mechanism, each contracts are periodically verified by the system. After the system request arrives, hosts prove their storage by providing a piece of the file and a list of hashes using the Merkle tree of the requested file. Each proof uses a randomly selected piece which is determined by the system and submitted by host to the blockchain. Moreover hosts provide Proof of Storage over pre-specified time interval. Each contract defines a maximum number of proofs that can be missed. If this pre-specified number exceeded, the contract becomes invalid. The SiaCoin project team also plans to use Proof of Burn as a consensus mechanism in the future.

4.1.5. Redundancy/fault tolerance mechanism

Clients can use erasure codes, such as regenerating codes (Rashmi, Shah, & Kumar, 2011) to safeguard against host inaccessibility. The values of file piece count and distributed host count vary based on the specific erasure code and redundancy factor. This allows high availability of client files. Moreover downloading data concurrently can increase the speed factor. While SiaCoin platform allows clients to choose preferred hosts by price, volume or reputation metrics, redundancy level can be changed client by client. Reputation is a point which defines the level of locked coin counts of a host.

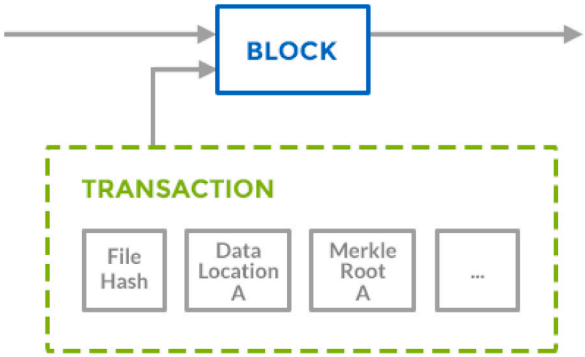

Fig. 2. Metadata information for Storj. *Source:* Courtesy of Wilkinson, Boshevski et al. (2014).

4.1.6. Decentralization level

SiaCoin works on completely decentralized infrastructure, which means that no outside organization or third party can access, control files or affect accessibility of the infrastructure. There is no central services such as indexing service or DNS controlled by an authority.

4.2. Storj

4.2.1. Maturity

Storj is a peer-to-peer cloud storage implementation that was introduced in 2014 (Wilkinson, Boshevski et al., 2014) and the second version was introduced in 2016 (Wilkinson et al., 2016). It allows users to share and transfer their data without the reliance of third party organizations. It uses the basics of the Metadisk project published in 2014 as a separate article (Wilkinson, Lowry & Boshevski, 2014). Metadisk has served as a test environment to the Storj solution. Storj does not use its own chain, instead it uses Ethereum public chain. The information on the chain is written in the Storj contract on this chain.

Similar to SiaCoin, Storj also has a whitepaper with the technological infrastructure details. They also publish their work as an open source project. In addition, it is currently serving at the production level. It has built-in compatible services with the de facto object storage API used by commercial cloud providers. Considering all these parameters, Storj has a high level of maturity.

4.2.2. Blockchain usage

Due to its nature, it is not possible to keep all data on the blockchain. If all data is attempted to be kept, the blockchain bloating will be revealed. In order to avoid this situation, only metadata information is kept on the blockchain. This metadata includes file hash, the network locations of the copies of the shards and Merkle roots . In this way, too many files can be kept in the infrastructure. However, if the number of files increases too much, the size of the blockchain also grows rapidly. Metadata information on Storj has file hash, data locations where file shards are distributed and Merkle root hash as seen in Fig. 2.

4.2.3. Security

The Metadisk model is used to secure the data. In this model, the hash value is used to store the data content securely after the file is encrypted. Thanks to the hash value, it can be guaranteed that the contents of the file have not changed. In addition, only the file owner has the decrypt key in this infrastructure. In this way, the infrastructure is resistant to man-in-the-middle attacks. In addition, to strengthen the security mechanism and strengthen the management of the data, the files are divided into standardized sized shards. This makes it difficult to resolve the file, since it allows the entire file to be kept fragmented on many different miner servers.

The system is also resistant to sybil attacks and bad actor attacks. Using random distribution and uniquely encrypting redundant copies, an attacker cannot execute a Sybil redundancy attack. With the Erasure coding method and group verification nodes, malicious attackers are prevented from abusing the system and these attackers are removed from the system.

4.2.4. Consensus mechanism

To ensure the integrity and availability of a shard which is stored on the network, a miner should prove cryptographically that it has the shard and has not modified it in any way. Storj provides proof of storage mechanism over the Merkle tree, to allow the user to check whether there is a change in the data. Merkle tree hierarchy can be seen in Fig. 3.

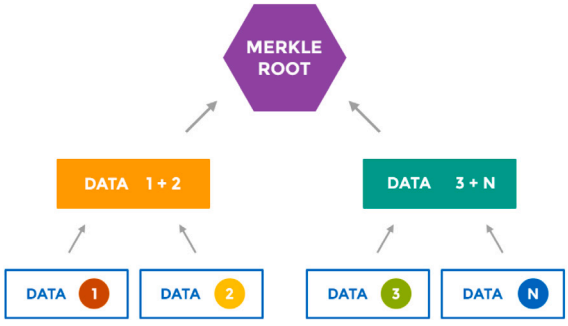

Fig. 3. Merkle tree hierarchy for Storj. *Source:* Courtesy of Wilkinson, Boshevski et al. (2014).

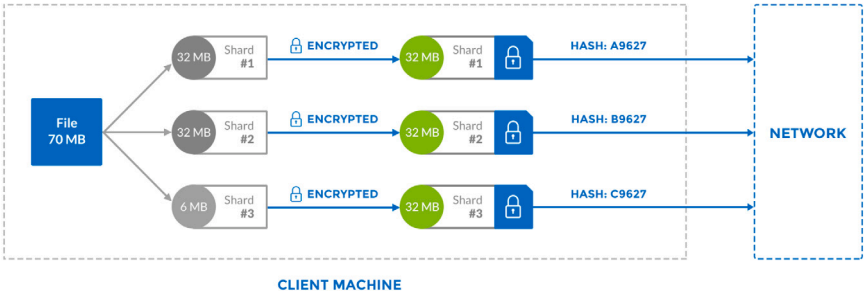

Fig. 4. Merkle tree hierarchy for Storj. *Source:* Courtesy of Wilkinson, Boshevski et al. (2014),

4.2.5. Redundancy/fault tolerance mechanism

All files on Storj are splitted into standardized sized shards. Splitting procedure can be shown in Fig. 4. Storj uses K-of-M erasure encoding scheme to make sure shards are available. In this structure, the node numbers can be selected by the client. In this way, according to the file importance level, redundancy can be adjusted. This naturally increases the speed of reading of the file. In addition, with the simple fault tolerant mechanism used to maintain the data availability, new nodes are recovered instead of the closed nodes.

4.2.6. Decentralization level

Storj works on a decentralized storage infrastructure, but it already uses centralized indexing server. Therefore, it can be said that Storj is partially decentralized. The centralized indexing infrastructure is completely under the management of the organization. From this perspective, if the organization's indexing service is shut down, the entire infrastructure will be affected. In addition, the organization has information about all files in the system.

4.3. FileCoin

4.3.1. Maturity

Instead of using cloud storage on a centralized network, FileCoin (Benet & Greco, 2018) proposed a way of creating a decentralized storage network using blockchain in order to store data. FileCoin works as an incentive layer on top of IPFS, which can provide storage infrastructure for any data (Benet, 2014). In FileCoin project, the protocol token which is also named FileCoin is used and that token can be spent by clients in order to store and retrieve data. It also can be mined by miners who are providing storage(storage miners) or serving data(retrieval miners). Storage miners must submit their pledge which is a commitment to offer storage to be able to accept orders from the Storage Market and they store data(pieces) which are taken from the client in their storage spaces(sectors). In addition to the Storage Market, there is another market called Retrieval Market where clients can retrieve their data. Unlike its counterparts, FileCoin has 2 types of mining methods. Storage miners store data and retrieval miners retrieve data for client. Retrieval mining has been revealed to meet the increasing bandwidth need more effectively. On top of all the features, FileCoin provides a good mechanism to achieve data integrity, retrievability, verifiability, auditability, incentive compatibility and confidentiality.

FileCoin has a whitepaper with the technological infrastructure details. They also publish their projects as open source. Unlike Storj and SiaCoin, it is not currently serving at the production level. Besides these, FileCoin is not compatible with object storage APIs because it positions itself as an alternative to de facto cloud storage infrastructures.

4.3.2. Blockchain usage

FileCoin keeps the metadata on the blockchain in order to manage storage operations. All management operations are handled on the blockchain network. Clients who want to store their data on host nodes, must submit their bid orders showing the price that clients want to pay. Storage miners must submit their pledge using pledge transactions. After their pledge transaction appears in the AllocationTable which is also located inside the blockchain, they can offer their storage space to the Storage Market. AllocationTable keeps track of pieces. In addition it keeps assigned sectors of the pieces. In addition, miners submit their ask orders which contain the price that miners want to take. If the bid orders and ask orders are matched, the client sends the piece to the miner. Furthermore, a deal order is signed by the client and the miner is submitted to the blockchain. To retrieve the data, a similar process is applied, but the order book cannot be run using blockchain since clients and retrieval miners can directly exchange the pieces. In short, for the storage, orders must run on the blockchain and must be checked by the decentralized network, however, retrieving is done without witnessed by the network.

4.3.3. Security

In FileCoin, clients can encrypt their data before submitting them to the network. In this way the data can be stored more securely. To ensure that storage miners are not malicious, they are required to generate their proofs. Since these proofs are stored on the blockchain, they can be verified by any user in the system. If proofs are skipped, then the miner will be penalized and will not be rewarded. In addition, in FileCoin platform, miners are rewarded for the storage space they are providing to the system. To be able to ensure that miners are not rewarded wrongly, there are three types of attacks that are prevented: Sybil attack, Outsourcing attack, and Generation attack.

In sybil attacks, malicious miners create multiple Sybil identities and try to get paid for multiple data storage for the data physically stored only once. In outsourcing attacks, malicious miners undertake to store much more data than the maximum capacity they can store. This commitment is based on the confidence that data can be retrieved quickly from other storage providers. In the generation attack, malicious miners generate storage requests using a small program and act as if they are storing large amounts of data. This increases the rate of winning the storage prize, which increases in proportion to the amount of storage capacity on FileCoin. FileCoin platform keeps the infrastructure more secure with the mechanisms it uses against all these attack types.

4.3.4. Consensus mechanism

To prevent the user from sybil, outsourcing or generation attacks FileCoin uses Proof-of-Replication(PoRep) and Proof-of-Spacetime(PoSt) consensus mechanisms. Both methods can be considered as specialized types of proof of storage mechanisms. Storage miners must prove that they are storing the data, therefore they generate proofs to show that they keep storing it and these proofs are stored in the blockchain to be verified by the network.

Proof-of-Replication (PoRep) is about guaranteeing that the miner stores any replica of the data in physically independent storage (Benet, Dalrymple, & Greco, 2017). Using Proof-of-Spacetime (PoSt), it is able to prove that the miner is storing the data during a period of time. To achieve this, storage providers periodically upload storage evidence to the blockchain network.

4.3.5. Redundancy/fault tolerance mechanism

In the platform, Erasure Coding (EC) and Information Dispersal Algorithm (IDA) are used to store the data. Information Dispersal Algorithm is used for data parsing and Erasure Coding Algorithm is used for achieving redundancy. FileCoin provides retrievability with these mechanisms. In Put request, the client can specify replication factor and (f,m)-tolerant value. The assumption used in the formula is that when there are m storage miners that store the data, the maximum f, the number of failures is tolerated. Higher accessibility can be achieved if the number of storage miners is increased. Client can adjust the cost-access balance according to the criticality level of its data. Since the network assigns the data to the miner and updates the allocation table, the storage allocations are public and if there are some faults, the network tries to repair them.

4.3.6. Decentralization level

Similar to SiaCoin, FileCoin also works on completely decentralized infrastructure. No outside company or third party can access, control or change the accessibility of the infrastructure. There is no service like indexing or DNS controlled by the central authority.

4.4. PPIO

4.4.1. Maturity

PPIO is a programmable decentralized storage and delivery network (Inc, 2019). It is a P2P network that does not rely on a central server. PPIO considers itself as a more robust, more secure, more efficient, and cheaper solution than existing cloud-based storage platforms.

PPIO is the most recent project but it also has a whitepaper with the technological infrastructure details. It is not fully open source project where some sub-components are open sourced but many of the project components are not. Besides these, PPIO is compatible with object storage APIs. PPIO is the least mature project, but it has advanced technological mechanisms.

4.4.2. Blockchain usage

In the PPIO's storage system, there are some different roles such as User Node, Source Node, Miner Node, Indexer Node, Verifier Node, and Coin-Pool Node. There are also incentives and penalties for the nodes according to their behavior on the network. The cryptocurrency of the network is PPIO Coin which can be collected as a reward by the nodes. User nodes are simply the consumers of the system. Source nodes are users that provide download services to other users. Miner nodes provide storage and bandwidth resources. Indexer nodes are needed for indexing and scheduling. Verifier nodes validate the storage proofs. Finally, Coin-Pool nodes provide payment services to the users. All the nodes that are mentioned can attend to the consensus mechanism of the PPIO.

PPIO keeps metadata of the client's file on Blockchain infrastructure. Like in some blockchain technologies, smart contracts are also used in PPIO's storage system. One of the smart contract types is a storage contract that is created when an object is stored on the network by the user. The other is a download contract that is created when an object is downloaded from the network by the user. A storage contract can be updated, renewed, or terminated. Besides, after the storage contract is created, user node pays the gas price to the system to store objects. Eventually miner, verifier, and indexer nodes receive this gas price.

4.4.3. Security

In the whitepaper of the PPIO, it is mentioned that there are five different attack types that are defended against: Sybil Attacks, Outsourcing Attacks, Generation Attacks, Distributed Denial of Service (DDoS) Attacks, and Eclipse Attacks.

Sybil attack, outsourcing attack, and generation attack are also explained previously. In addition to them, in DDoS attacks, a large number of nodes attack a target node at the same time to generate huge traffic on the node. Because of this substantial traffic, the node fails to respond. In eclipse attacks, all connections of the attacked node are controlled by malicious nodes and the node became isolated from the network. Since it is isolated from the network, it cannot get the correct information from the rest of the network and it can be totally manipulated by malicious miners. As a result, it may cause economic losses. PPIO is still developing and all these attacks are trying to be prevented using some methods, proofs, and algorithms.

4.4.4. Consensus mechanism

Like the other three blockchain-based distributed technologies that are mentioned in this paper, there is also a consensus mechanism in PPIO's storage system. In any decentralized blockchain network, it is necessary to have a consensus mechanism to add and verify transactions and to provide security.

PPIO project has four different proof methods. Proof-of-Download (PoD), Proof-of-Replication (PoRep), Proof-of-Spacetime (PoSt), and Light-Proof-of-Capacity (LPoC) are used as proof algorithms in PPIO. In addition to PoRep and PoSt that are explained before, to ensure that the data is correctly downloaded from the miner, PoD is used. Furthermore, LPoC is used for verifying the storage capacity of the miner.

PPIO mentioned that using PoSt alone is not enough to measure contributions of each node because even if nodes have not enough storage capacity, they can still contribute with their high bandwidth. Therefore, as well as the storage, PPIO also takes bandwidth as a contribution and this requires a new consensus mechanism.

First of all, based on the power of the miners, a pool of candidate miner nodes is created. As mentioned above, the power of the miners is considered with respect to the storage and bandwidth. Secondly, using Verifiable Random Function (VRF), a random miner is selected to build a new block. After selecting a miner to build a new block, using a consensus which achieves Byzantine Fault Tolerance (BFT), the block will be validated. This unique consensus algorithm is named as PVFT.

4.4.5. Redundancy/fault tolerance mechanism

To be able to have reliable storage, PPIO's storage system uses two different methods to have data redundancy. One of the methods is replicating the data. Using this method, an additional full copy of the data is also stored. The other method is named as redundant coding which is actually erasure encoding. Data is divided into more blocks and even if a certain number of blocks that are holding, some pieces of that data are lost. It is possible to recover the data with the help of erasure encoding.

4.4.6. Decentralization level

PPIO has three phase planning for decentralization level. Strong center, Weak center, and Decenter phase are follows each other. At the first stage, indexer and verifier nodes will be centralized. Other nodes (user, source, miner) will be decentralized. In the second phase indexer and verifier nodes will be managed via consortium. Finally, all nodes are planned to be decentralized.

5. Comparison and evaluation

Although blockchain based storage infrastructures serve similar purposes, they use different technologies and mechanisms. Generally, they use compression, encryption, erasure coding, and replication methods. However, the ways of applying these methods vary according to the technology. Comparison between current blockchain based data storage solutions can be seen in Table 1.

Maturity: Firstly, white papers can be examined to give an idea of the maturity level of the projects. A white paper indicates a level of maturity since the technical details of the solutions are generally only available in these documents. In this regard, all projects published a whitepaper and provide details of their project in these documents. A second parameter for maturity is the current production level. Among the four projects, SiaCoin is fully on the production stage where users can participate in their systems. SiaCoin has been running mainnet for a long time and this network is actively serving all end users. People who want to use the SiaCoin infrastructure can easily download and use the client application. Although Storj has announced that it has begun

Table 1
Comparison between current blockchain based data storage solutions.

Category	Field	SiaCoin	Storj	FileCoin	PPIO
Maturity	White Paper	Yes	Yes	Yes	Yes
	Production	Yes	Partially Yes	No	No
	Open Source	Yes	Yes	Yes	Partially Yes/ Planned
	Cloud Compatibility	Partially Yes (With 3rd party software)	Yes	No	Yes
Blockchain Usage	Focused Elements	Contract, Proof, Contract Update	File Hash, Network Locations of Copies	Storage Market, Retrieval Market	User, Source, Miner, Indexer, Verifier, Coin-Pool Nodes
Security	Client Side Encryption Algorithm	Twofish	AES-256	AES-256	AES-256
Consensus	Mechanism	PoS	PoS	PoRep, PoSt	PoRep, PoSt, PoD, LPoC
Redundancy/ Fault Tolerance	Mechanism	Erasure Code	Erasure Code	Erasure Code	Erasure Code, Full Copy
Decentralization	Data	Fully Decentralized	Centralized Indexing Decentralized Storage	Fully Decentralized	Three phase planning; Centralized, Semi-centralized, Decentralized
	Chain	Single Main	Single Main	Ethereum main	Main with side chains

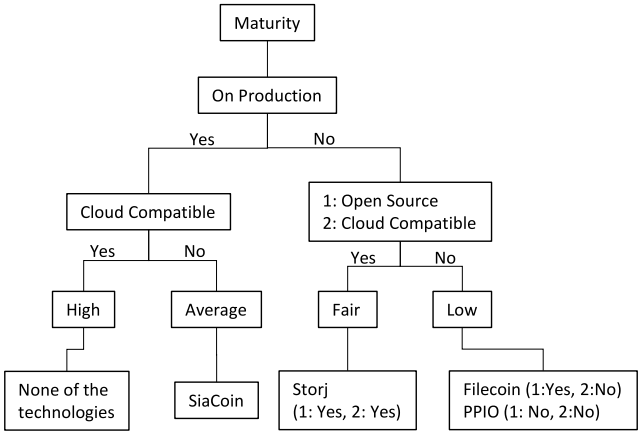

Fig. 5. Maturity categorization of four technologies.

to use the production environment, it does not yet have any accessible environment for end users. The other two solutions, FileCoin and PPIO, have not yet reached the production stage. Since these solutions have not yet published their mainnet, it should be noted that the working logic or proposed architecture of their technology may change. Not being in production may always raise concerns about the future of the projects. In summary, when compared to development stages, SiaCoin works on the production stage fully and more advanced than others.

When compared in terms of source code privacy, all projects have chosen to be open source. Unlike others, PPIO has made demo and SDK applications available for everyone but not yet released their core project as open source. Despite this, all project leaders announced their open source perspective and they are aware of the power of open source environments and community.

Object storage compatibility is one of the most important features for all storage projects. Cloud services are commonly used in many companies. For that reason, object storage compatibility is important for emerging storage technologies. With the ability for compatibility, projects that already use cloud storage services can easily participate in blockchain based infrastructures. Storj and PPIO have object storage support and serve internal Amazon S3 compatible API to everyone. SiaCoin does not provide internal compatible API but it is integrated with S3 compatible Minio which allows easy communication between S3 compatible infrastructure and SiaCoin infrastructure. In contrast, FileCoin does not provide compatible object storage API, thus FileCoin users who want to move their data from cloud storage have to use their own special mechanisms.

As it can be seen from Fig. 5, we categorized the technologies' maturity into four categories: high, average, fair, and low and considered the comparison table that we created. Since there is no technology that meets all criteria such as having a white paper,

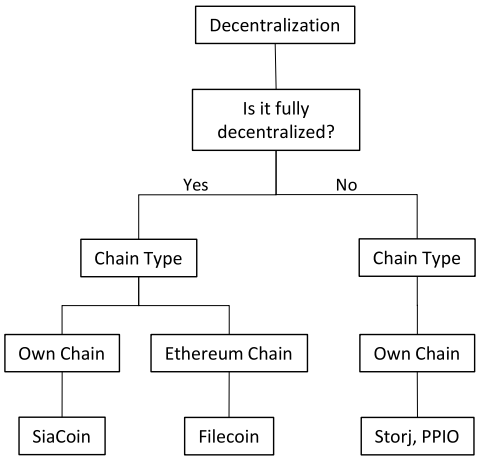

Fig. 6. Decentralization categorization of technologies.

being on production, having open source development, and being cloud compatible, you can see that none of these technologies is categorized as highly mature. SiaCoin is the one with the highest maturity among these technologies (average) since it is the only one in production. Storj follows the SiaCoin having a fair maturity since it is not on production but is open source and cloud compatible. Filecoin and PPIO are categorized as low maturity since they are not on production and cloud compatible. Also, PPIO is not open source unlike Filecoin.

Blockchain Usage: When blockchain based technologies are compared, it is important to compare the Blockchain Usage category. All compared projects store their file metadata on a blockchain. In this respect, all projects are similar, however, when examined in more detail, the main blockchain elements differ from each other. SiaCoin focuses on a scripting mechanism which includes contract, proof and contract update to share local storage between peers. Storj focuses on metadata content which includes file hash, location info and Merkle root. FileCoin focuses on operations on storage and delivery markets used to serve different businesses. PPIO contains many different types of nodes and focuses on the processing of these specialized nodes on the blockchain network.

Security: In order to make a comparison in terms of security, production level services and architectural details of the projects should be shared. However, there is no detail given in any of the projects. On the other hand, the encryption algorithms used on the client side are known and can be compared. SiaCoin uses Twofish encryption management, while other projects use AES encryption algorithm. Both algorithms support 128, 192 and 256 bit key sizes. In summary, the hacking difficulties of these methods are related to the selected key size rather than the chosen method. Since both methods have similar security levels, it can be said that the security levels used on the client side in their projects are close to each other.

Consensus: All blockchain based storage technologies use proof techniques to come to a consensus. SiaCoin and Storj use Proof of Storage (PoS) method. FileCoin has brought a new perspective to the consensus method which is Proof of Spacetime (PoSt) and Proof of Replication (PoRep) method. PPIO also facilitates these methods as well. In addition, PPIO provides Proof of Download (PoD) and Light Proof of Capacity (LPoC) methods.

Redundancy/Fault Tolerance: Redundancy, which is used to increase reliability by duplicating crucial components of a system, is another important technology for storage services. Redundancy mechanisms improve the high availability of data, thus, all the projects support the erasure coding mechanism. In this method, data can be stored in many different locations without overloading the network. Additionally, PPIO uses the Full Copy method in addition to the erasure coding method to provide data distribution more effectively.

Decentralization: Another important metric to compare blockchain-based technologies is the decentralization level. In terms of keeping data on nodes, SiaCoin and FileCoin uses completely decentralized technology infrastructure. Even if the company that creates the technological infrastructure is shut down with all its services, because of the existence of other participants, these projects continue to survive and serve. However, Storj adopted a different way from these projects. The indexing system where all files are stored on metadata is kept centralized. The rest of the infrastructure is decentralized. On the other hand, PPIO preferred the gradual decentralization method. It is planned to develop the system on three stages; centralized, semi-centralized and fully decentralized. No matter how data is distributed in such systems, a single source access bottleneck may be experienced at the blockchain level. Therefore, it is important to establish a distributed structure in the chain. In terms of chain structure, SiaCoin and Storj has its own mainnet and it does not use multiple chains which we refer as “Single Chain” in Table 1. Similarly, FileCoin is implemented on Ethereum main chain. Unlike others, PPIO is implemented on the main chain with side chains in order to provide highly decentralized structure on the chain level as well.

We categorized the technologies considering their decentralization levels and chain types. From Fig. 6, it can be seen that the SiaCoin and FileCoin are fully decentralized. However, SiaCoin has its own chain while FileCoin uses the Ethereum chain. Storj and PPIO are not fully decentralized and both of them have their own chains.

In summary, when we compare all projects, there is no single winner in all categories. However, an assessment can be made for each category and prominent applications can be identified within that category. For the maturity category, it can be said that SiaCoin and Storj are more mature than all other projects examined. Compared to the effective use of Blockchain, we can say that PPIO, which hosts a much higher number of specialized nodes, is superior to others. From a security point of view, no significant difference was found between the projects. Comparing the diversity of consensus mechanisms, we can say that PPIO is richer than others. When a comparison is made in terms of Redundancy/Fault Tolerance mechanism, PPIO stands out because it includes the full copy feature besides erasure coding technology. In terms of data-based decentralization, SiaCoin and FileCoin are superior to other projects because they are fully decentralized. On the other hand, PPIO is the only project providing chain based decentralization.

6. Conclusion

Since the advent of Blockchain technology, the implementation of this new technology has been realized in many areas. Storage infrastructure is one of such areas where Blockchain reshaped the traditional solutions. There are several state-of-the-art solutions for distributed storage based on blockchain technology. However, to compare and contrast these solutions a formal classification and categorization is missing. In this work, as a novel contribution, a categorization for blockchain-based distributed storage technologies is proposed along with a new taxonomy. To our best knowledge, proposed taxonomy is the first in the literature specifically for blockchain based distributed storage technologies. With the proposed taxonomy, current and newly proposed solutions can be compared in a more systematic way.

CRediT authorship contribution statement

Omer F. Cangir: Concept, Design, Analysis, Writing, Revision of the manuscript. **Onur Cankur:** Concept, Design, Analysis, Writing, Revision of the manuscript. **Adnan Ozsoy:** Concept, Design, Analysis, Writing, Revision of the manuscript.

References

Baniata, H., Anaqreh, A., & Kertesz, A. (2021). PF-BTS: A privacy-aware fog-enhanced blockchain-assisted task scheduling. *Information Processing & Management*, *58*(1), Article 102393.

Benet, J. (2014). Ipfs-content addressed, versioned, P2P file system. arXiv preprint arXiv:1407.3561.

Benet, J., Dalrymple, D., & Greco, N. (2017). *Proof of replication (vol. 27)*. Protocol Labs.

Benet, J., & Greco, N. (2018). *Filecoin: A decentralized storage network*. Protoc. labs.

Benisi, N. Z., Aminian, M., & Javadi, B. (2020). Blockchain-based decentralized storage networks: A survey. *Journal of Network and Computer Applications*, Article 102656.

Berdik, D., Otoum, S., Schmidt, N., Porter, D., & Jararweh, Y. (2021). A survey on blockchain for information systems management and security. *Information Processing & Management*, *58*(1), Article 102397.

Blomer, J. (2015). A survey on distributed file system technology. *Journal of Physics: Conference Series*, http://dx.doi.org/10.1088/1742-6596/608/1/012039.

Bouraga, S. (2021). A taxonomy of blockchain consensus protocols: A survey and classification framework. *Expert Systems with Applications*, *168*, Article 114384.

Boyer, E. B., Broomfield, M. C., & Perrotti, T. A. (2012). *Glusterfs one storage server to rule them all*: *Technical report*, Los Alamos, NM (United States): Los Alamos National Lab.(LANL).

Campanile, L., Iacono, M., Marulli, F., & Mastroianni, M. (2021). Designing a GDPR compliant blockchain-based IoV distributed information tracking system. *Information Processing & Management*, *58*(3), Article 102511.

Clark, T. (1999). *Designing storage area networks: a practical reference for implementing Fibre Channel SANs*. Addison-Wesley Professional.

Daniel, E., & Tschorsch, F. (2021). IPFS and friends: A qualitative comparison of next generation peer-to-peer data networks. arXiv preprint arXiv:2102.12737.

Depardon, B., Le Mahec, G., & Séguin, C. (2013). *Analysis of six distributed file systems*: *Research report*, (p. 44).

Esposito, C., Ficco, M., & Gupta, B. B. (2021). Blockchain-based authentication and authorization for smart city applications. *Information Processing & Management*, *58*(2), Article 102468.

Ford, D., Popovici, F. I., Stokely, M., Truong, V.-a., Barroso, L., Grimes, C., et al. (2010). Availability in globally distributed storage systems. *Operations Research*.

Gaonkar, P. E., Bojewar, S., & Das, J. A. (2013). A survey: data storage technologies. *International Journal of Innovative Science Engineering and Technology*, *2*, 547–554.

Garcia-Molina, H., & Polyzois, C. A. (1990). Issues in disaster recovery. In *Digest of papers compcon spring'90. thirty-fifth IEEE computer society international conference on intellectual leverage* (pp. 573–577). IEEE.

Hardin, T., & Kotz, D. (2021). Amanuensis: Information provenance for health-data systems. *Information Processing & Management*, *58*(2), Article 102460.

Howard, J. H., Kazar, M. L., Menees, S. G., Nichols, D. A., Satyanarayanan, M., Sidebotham, R. N., et al. (1988). Scale and performance in a distributed file system. *ACM Transactions on Computer Systems (TOCS)*, *6*(1), 51–81.

Hu, T., Liu, X., Chen, T., Zhang, X., Huang, X., Niu, W., et al. (2021). Transaction-based classification and detection approach for Ethereum smart contract. *Information Processing & Management*, *58*(2), Article 102462.

Inc, P. (2019). *PPIO: A programmable P2P storage and delivery network*: *Technical report*, PPIO.

Jing, N., Liu, Q., & Sugumaran, V. (2021). A blockchain-based code copyright management system. *Information Processing & Management*, *58*(3), Article 102518.

Kächele, S., Spann, C., Hauck, F. J., & Domaschka, J. (2013). Beyond IaaS and PaaS: An extended cloud taxonomy for computation, storage and networking. In *2013 IEEE/ACM 6th international conference on utility and cloud computing* (pp. 75–82). IEEE.

Katz, R. H. (1992). Network-attached storage systems. In *Proceedings scalable high performance computing conference* (pp. 68–75). IEEE.

Khalid, A., Iftikhar, M. S., Almogren, A., Khalid, R., Afzal, M. K., & Javaid, N. (2021). A blockchain based incentive provisioning scheme for traffic event validation and information storage in VANETs. *Information Processing & Management*, *58*(2), Article 102464.

Kim, S., Kwon, Y., & Cho, S. (2018). A survey of scalability solutions on blockchain. In *2018 International conference on information and communication technology convergence* (pp. 1204–1207). IEEE.

Labazova, O., Dehling, T., & Sunyaev, A. (2019). From hype to reality: A taxonomy of blockchain applications. In *Proceedings of the 52nd hawaii international conference on system sciences*.

Li, J., Wu, J., Jiang, G., & Srikanthan, T. (2020). Blockchain-based public auditing for big data in cloud storage. *Information Processing & Management*, *57*(6), Article 102382.

Marketing, I. (2016). *key marketing trends for 2017*: *Technical report*, IBM, URL: https://www-01.ibm.com/common/ssi/cgi-bin/ssialias.
Mohsin, A., Zaidan, A., Zaidan, B., Albahri, O., Albahri, A., Alsalem, M., et al. (2019). Blockchain authentication of network applications: Taxonomy, classification, capabilities, open challenges, motivations, recommendations and future directions. *Computer Standards & Interfaces*, *64*, 41–60.
Nakamoto, S. (2019). *Bitcoin: A peer-to-peer electronic cash system*: *Technical report*, Manubot.
Oham, C., Michelin, R. A., Jurdak, R., Kanhere, S. S., & Jha, S. (2021). B-FERL: Blockchain based framework for securing smart vehicles. *Information Processing & Management*, *58*(1), Article 102426.
Padhy, R. P., & Patra, M. R. (2012). Moving towards SAN storage: An enterprise perspective. *Journal of Global Research in Computer Science*, *3*(7).
Pilaud, D., Halbwachs, N., & Plaice, J. (1987). LUStre: A declarative language for programming synchronous systems. In *Proceedings of the 14th annual ACM symposium on principles of programming languages (vol. 178)* (p. 188). New York, NY: ACM.
Pilkington, M. (2016). 11 blockchain technology: principles and applications. In *Research handbook on digital transformations (vol. 225)*. Edward Elgar Publishing Cheltenham, UK.
Placek, M., & Buyya, R. (2006). *A taxonomy of distributed storage systems*: *Reporte técnico*, Universidad de Melbourne, Laboratorio de sistemas distribuidos y cómputo grid.
Pouwelse, J., Garbacki, P., Epema, D., & Sips, H. (2005). The bittorrent P2P file-sharing system: Measurements and analysis. In *International workshop on peer-to-peer systems* (pp. 205–216). Springer.
Putz, B., Dietz, M., Empl, P., & Pernul, G. (2021). Ethertwin: Blockchain-based secure digital twin information management. *Information Processing & Management*, *58*(1), Article 102425.
Rashmi, K. V., Shah, N. B., & Kumar, P. V. (2011). Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction. *IEEE Transactions on Information Theory*, *57*(8), 5227–5239.
Reinsel, D., Gantz, J., & Rydning, J. (2017). Data age 2025: The evolution of data to life-critical don't focus on big data. In *Focus on the data that's big sponsored by seagate the evolution of data to life-critical don't focus on big data*.
Rimal, B. P., Choi, E., & Lumb, I. (2009a). A taxonomy and survey of cloud computing systems. In *2009 Fifth international joint conference on INC, IMS and IDC* (pp. 44–51). Ieee.
Rimal, B. P., Choi, E., & Lumb, I. (2009b). A taxonomy and survey of cloud computing systems. In *2009 Fifth international joint conference on INC, IMS and IDC* (pp. 44–51). Ieee.
Saroiu, S., Gummadi, K. P., & Gribble, S. D. (2003). Measuring and analyzing the characteristics of Napster and Gnutella hosts. *Multimedia Systems*, *9*(2), 170–184.
Shafagh, H., Burkhalter, L., Hithnawi, A., & Duquennoy, S. (2017). Towards blockchain-based auditable storage and sharing of IoT data. In *Proceedings of the 2017 on cloud computing security workshop* (pp. 45–50). ACM.
Shvachko, K., Kuang, H., Radia, S., Chansler, R., et al. (2010). The hadoop distributed file system. In *MSST (vol. 10)* (pp. 1–10).
Stodt, J., & Reich, C. (2020). Data confidentiality in P2P communication and smart contracts of blockchain in industry 4.0. arXiv preprint arXiv:2007.14195.
Swan, M. (2015). *Blockchain: Blueprint for a new economy*. O'Reilly Media, Inc..
Tasca, P., & Tessone, C. J. (2017). Taxonomy of blockchain technologies. Principles of identification and classification. arXiv preprint arXiv:1708.04872.
Thanh, T. D., Mohan, S., Choi, E., Kim, S., & Kim, P. (2008). A taxonomy and survey on distributed file systems. *1*, In *2008 Fourth international conference on networked computing and advanced information management* (pp. 144–149). IEEE.
Vorick, D., & Champine, L. (2014). *Sia: Simple decentralized storage*. Nebulous Inc.
Wang, H., Jing, Q., He, B., Qian, Z., & Zhou, L. (2010). Distributed systems meet economics: pricing in the cloud. *Methodology*.
Wang, Y., & Vassileva, J. (2003). Trust and reputation model in peer-to-peer networks. In *Proceedings third international conference on peer-to-peer computing* (pp. 150–157). IEEE.
Weil, S. A., Brandt, S. A., Miller, E. L., Long, D. D., & Maltzahn, C. (2006). Ceph: A scalable, high-performance distributed file system. In *Proceedings of the 7th symposium on operating systems design and implementation* (pp. 307–320). USENIX Association.
Wilkinson, S., Boshevski, T., Brandoff, J., & Buterin, V. (2014). *Storj a peer-to-peer cloud storage network*: *Technical report, storj.io*, Citeseer.
Wilkinson, S., Boshevski, T., Brandoff, J., Prestwich, J., Hall, G., Gerbes, P., et al. (2016). *Storj: A peer-to-peer cloud storage network v2. 0*. Citeseer Press.
Wilkinson, S., Lowry, J., & Boshevski, T. (2014). Metadisk a blockchain-based decentralized file storage application. *Tech. Rep.*.
Xu, X., Sun, G., Luo, L., Cao, H., Yu, H., & Vasilakos, A. V. (2021). Latency performance modeling and analysis for hyperledger fabric blockchain network. *Information Processing & Management*, *58*(1), Article 102436.
Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., et al. (2017). A taxonomy of blockchain-based systems for architecture design. In *2017 IEEE international conference on software architecture* (pp. 243–252). IEEE.
You, L., & Karamanolis, C. T. (2004). Evaluation of efficient archival storage techniques. In *MSST* (pp. 227–232).
Yu, G., Zhang, L., Wang, X., Yu, K., Ni, W., Zhang, J. A., et al. (2021). A novel Dual-Blockchained structure for contract-theoretic LoRa-based information systems. *Information Processing & Management*, *58*(3), Article 102492.
Zhao, Q., Chen, S., Liu, Z., Baker, T., & Zhang, Y. (2020). Blockchain-based privacy-preserving remote data integrity checking scheme for IoT information systems. *Information Processing & Management*, *57*(6), Article 102355.
Zyskind, G., Nathan, O., et al. (2015). Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE security and privacy workshops* (pp. 180–184). IEEE.