# A Blockchain based Autonomous Data Storage System-Revolutionizing Data Storage and Comparative Analysis with Centralized System

Nguyen Dinh Hoang Son
UG Scholar
Computing Dept, FPT Greenwich University,
Ho Chi Minh City, Vietnam

Sharmila Mathivanan
IT Lecturer
Computing Dept, FPT Greenwich University,
Ho Chi Minh City, Vietnam

Ho Nguyen Phu Bao
Academic Head & Lecturer
Computing Dept, FPT Greenwich University,
Ho Chi Minh City, Vietnam

## ABSTRACT
Cloud storage systems have revolutionized how data is stored, accessed, and managed in today's digital landscape. With the rapid growth of digital information and the increasing reliance on remote access and collaboration, traditional local storage solutions still need to be improved in meeting the demands of modern businesses and individuals. In traditional cloud storage systems, attribute-based encryption (ABE) has emerged as a significant technology for addressing data privacy challenges and fine-grained access control. However, attribute-based encryption (ABE) schemes, while effective in data privacy and access control in cloud storage systems, suffer from inherent limitations. Specifically, in all ABE schemes, the private key generator (PKG) processes the authority to encrypt all data stored in the cloud server. This centralized decryption capability introduces potential risks, including the abuse of encryption keys and the potential leakage of sensitive data. Moreover, the traditional cloud storage model operates in a centralized storage manner, making it susceptible to single points of failure that can result in system-wide disruptions or collapses. These issues highlight the need for alternative approaches that mitigate the risks associated with centralized decryption and single points of failure, ensuring enhanced security and reliability in cloud storage systems. Blockchain technology has brought decentralized storage models to the forefront of public attention. Decentralized storage systems offer a viable solution to address the inherent vulnerabilities of single points of failure found in traditional cloud storage architectures. Moreover, decentralized storage exhibits numerous advantages compared to centralized storage, including cost-effectiveness and enhanced data throughput capabilities. By leveraging the principles of blockchain technology, decentralized storage systems present a promising alternative that can potentially revolutionize how data is stored, accessed, and managed. This research paper studies the paradigm of blockchain-based decentralized data storage systems and compares them with traditional cloud centralized storage systems. The study begins with an exploration of the background and significance of data storage, highlighting the need for advanced storage solutions. The research objectives include examining the principles, architecture, and challenges of blockchain-decentralized storage systems, along with their advantages over cloud centralized storage. This paper mainly contributes to the understanding of blockchain decentralized storage systems and their potential impact on data storage practices.

## Keywords
Attribute-based encryption, Centralized decryption, Blockchain technology, Data throughput, Smart Contract, Deep Root Analytics.

## 1. INTRODUCTION
The landscape of data storage and sharing has undergone remarkable transformations in recent years. From the early days of physical storage devices to the revolutionary impact of the internet, the manner in which we store and oversee data has undergone significant transformations. The swift advancement of internet technology has given rise to the prominence of cloud storage as a pivotal operational model in everyday existence, offering convenient and flexible data storage and sharing solutions for individuals and enterprises alike.

In the year 2021, global data generation reached an impressive volume of about 79 zettabytes. However, an insightful analysis presented by [1] sheds light on the fact that major tech giants such as Google, Amazon, and Facebook collectively managed to store a mere 1.2 million terabytes of data. Even with a significant fivefold increase in data storage during 2022, this accumulation represents a mere fraction, specifically 0.015%, of the total data generated.

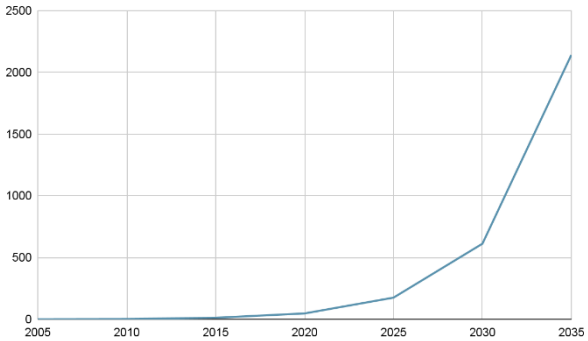


**Chart 1: Global Yearly Data Generation Growth in zettabytes (2005-2035)**

This underscores the relatively modest adoption of decentralized storage solutions, highlighting their current status as a relatively small segment within the larger data storage landscape. In 2006, Clive Humby famously stated that "data is the new oil," highlighting the growing significance of data in the modern world. This statement remains highly relevant today, as the data becomes increasingly intertwined with the online activities and digital presence. However, traditional centralized storage solutions, despite their accessibility and convenience, come with inherent drawbacks. Users often surrender control over their data, leading to concerns about

privacy, security, and the vulnerability of sensitive information.

While traditional distributed storage systems have shown limitations in terms of centralization and trustworthiness of third-party institutions, recent years have witnessed the emergence of innovative solutions leveraging blockchain technology. One notable example is Arweave [2], which introduces a decentralized storage protocol called the "Blockweave" to ensure permanent and censorship-resistant data storage. Filecoin [3], another groundbreaking platform, leverages the Inter Planetary File System (IPFS) [4] to establish a fully decentralized distributed storage network, enabling customers and storage miners to participate in a marketplace for storage and retrieval services. Siacoin [5], on the other hand, employs smart contract technology to facilitate secure and reliable data storage through agreements between storage providers and customers.

These advancements in decentralized storage systems have the potential to address the limitations of centralized approaches [6], ensuring enhanced data security [7], availability, and control for users. By decentralizing storage, these systems offer advantages such as lower costs compared to traditional cloud storage, high data throughput, and reduced concerns about single points of failure [8].

In light of these developments, this research paper aims to explore the potential of blockchain-based decentralized storage systems, examining their security, availability, and privacy aspects. We present a framework that allows for precise management of data access within decentralized storage systems and streamlines the process of retrieving data efficiently using relevant keywords. By investigating the benefits and challenges of these systems, we seek to contribute to the broader understanding of how decentralized storage can reshape the landscape of data storage and sharing, ensuring a more secure and user-centric approach to data management.

The subsequent sections of this paper provide a comprehensive review of relevant literature, discuss the methodology employed, present the principles and architecture of blockchain decentralized storage systems, analyze their advantages and challenges, and showcase a case study/project of a decentralized data storage system. Furthermore, this paper presents the results and analysis of the research and summarize the key findings.

# 2. RELATED WORKS

## 2.1 Overview of Decentralized Data Storage System

### *2.1.1 Blockchain*

In essence, a blockchain functions as an expanding digital ledger that meticulously records data in a sequential arrangement of multiple blocks. These blocks are interconnected and safeguarded through the implementation of cryptographic methods [9].

The blockchain is distinguished by its unique data structure, which integrates data blocks in a precise chain, ensuring the preservation of chronological order [10]. This technology boasts decentralization, cryptographic security, and the capacity to establish an immutable and tamper-proof distributed ledger system.

Each block within the chain comprises a specific dataset and a reference to the preceding block, thereby establishing a continuous and transparent log of transactions or information. By employing cryptographic techniques, the integrity and authenticity of the data stored in the blockchain are upheld, rendering any tampering or forgery exceedingly challenging.

The decentralized and transparent nature of blockchain technology renders it an exceptionally dependable and credible platform for a diverse range of applications. Its applicability extends from cryptocurrencies to supply chain management, voting systems, and beyond [11].

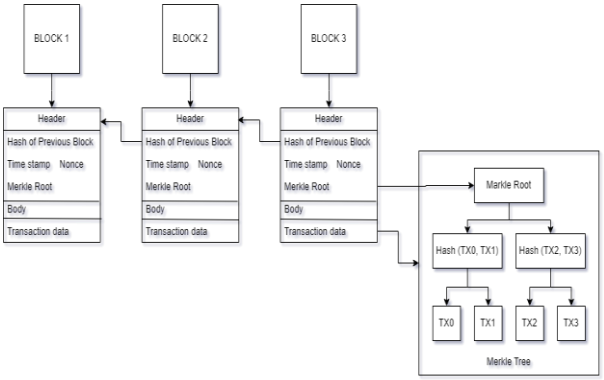


**Fig 1: Blockchain Structure**

Blockchain technology is built upon fundamental principles that distinguish it from traditional centralized mechanisms. Its decentralized structure, based on a distributed system architecture, forms the cornerstone of trust among participating nodes, defining its key characteristic [11].

The underlying data layer of the blockchain is fortified by several techniques, ensuring its security and integrity. Hashing, asymmetric encryption [12], Merkle trees [13], and timestamps play essential roles in this regard. Hashing algorithms transform data into unique fixed-size representations, guaranteeing integrity and facilitating efficient verification. Asymmetric encryption techniques enable secure communication channels, digital signatures, and authentication.

Merkle trees offer an efficient means of verifying data consistency and integrity within the blockchain. By organizing data hierarchically, they facilitate quick identification of tampered or modified data. Timestamps ensure the chronological order of recorded events, ensuring accuracy and accountability.

**Figure 1** illustrates the basic structural model of blockchain technology, showcasing the interplay between its components and emphasizing its decentralized and secure nature.

As a distributed ledger technology (DLT), blockchain is designed to be highly resistant to modification and fraudulent activities like double-spending. The integrity of the Bitcoin blockchain, for example, makes tampering impractical due to the extensive computational power required. This ensures the uniqueness and uncopiable nature of each Bitcoin unit [9].

### *2.1.2 Smart Contract in Decentralized Data Storage System*

A smart contract is a program that operates on the Ethereum blockchain. It consists of code (functions) and data (state) residing at a specific address on the Ethereum blockchain [14].

Smart contracts are considered Ethereum accounts, meaning they have a balance and can receive transactions. However, unlike user-controlled accounts, smart contracts are deployed to the network and function autonomously based on their programmed instructions. Users can interact with smart

contracts by sending transactions that execute predefined functions. These contracts can establish and enforce rules, similar to traditional contracts, using the underlying code. By default, smart contracts cannot be deleted, and any interactions with them are irreversible [14].

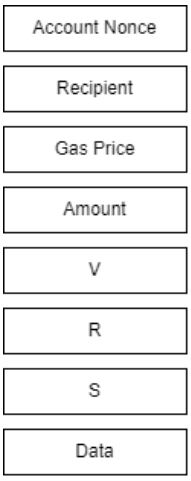


**Fig 2: Ethereum Transaction Data Structure**

Smart contracts are computer protocols that can execute and verify themselves without the need for human intervention [15] [16]. They are essentially autonomous computer programs that perform contract-related operations and provide evidence of their effectiveness. Before deployment, all the necessary logic processes associated with the contract are established.

Within the Ethereum blockchain ecosystem, a smart contract represents a distinctive account paired with specific code. The act of deployment encompasses the compilation of this smart contract into bytecode compatible with the Ethereum Virtual Machine (EVM), followed by its placement onto the Ethereum blockchain. Subsequently, the contract's address and its corresponding Application Binary Interface (ABI) are documented, thereby facilitating engagement with the contract by utilizing these designations. This deployment process is visualized in **Figure 2** as referenced [17].

In a decentralized data storage system, smart contracts are utilized to store encrypted data keywords and execute several functions for data sharing and data users. In a trustless environment, users deposit the service fee directly into the contract, and the smart contract facilitates the retrieval process. The service fee will only be deducted from the contract once the correct result is retrieved. This approach effectively addresses the issue of searchers intentionally withholding or providing incorrect results to conserve resources, a challenge often encountered in traditional cloud storage schemes.

### *2.1.3 Decentralized Data Storage System*

Decentralized storage denotes a storage resolution operating within a decentralized network, often built upon the foundations of blockchain technology. Unlike centralized storage, where data is stored on a single server controlled by a central authority, decentralized storage distributes data across multiple nodes in a network. This approach enhances security, reliability, and user control over data [3].

In decentralized storage, data is divided into small pieces and stored on multiple nodes within a peer-to-peer network like BitTorrent or the Inter Planetary File System (IPFS). When retrieving data, the network gathers the fragmented components from different nodes and reconstructs the file for downloading as illustrated in **Fig 3** [4].

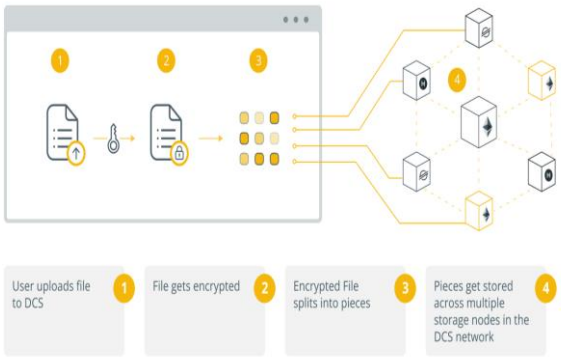


**Fig 3: Working of a Decentralized Storage System [3]**

To ensure security, the data stored in a decentralized system is automatically encrypted using cryptographic hash mechanisms. Only users with private keys can access their data, preventing unauthorized access [4].

While decentralized storage offers notable advantages, it is essential to consider these limitations when evaluating its suitability for specific use cases.

## 2.2 Existing Research and Studies Methodology

### *2.2.1 Blockchain Technology*

Over the past few years, there has been a surge in the popularity of decentralized cryptocurrencies like Bitcoin [7], Ethereum [15], and others. This rise in popularity has led to increased recognition and focus on blockchain technology, which serves as the fundamental technology supporting these cryptocurrencies. Currently, the blockchain has emerged as a significant player in the financial sector [18]. Moreover, its potential extends beyond finance, finding utility in various non-financial domains. Examples include decentralized supply chain management [19], identity-based public key infrastructure (PKI), decentralized proof of document existence [20], decentralized Internet of Things (IoT) applications [20], decentralized storage solutions, and more.

### *2.2.2 Decentralized Data Storage System*

In recent years, decentralized storage systems have gained significant attention due to their potential to address the limitation of centralized storage solutions. Numerous studies and research papers have explored various aspects of decentralized data storage systems, including their architectures [1], security considerations [21], performance evaluation [22], and real-world applications.

## 2.3 The Problems with Centralized Data Storage

### *2.3.1 Censorship*

The current centralized model of the Internet makes it vulnerable to censorship. However, decentralization can address this issue effectively. For instance, even if certain countries block access to Wikipedia, it remains attainable via decentralized storage platforms. Likewise, within an oppressive regime, protesters can upload information onto decentralized storage platforms, minimizing the susceptibility to censorship.

### *2.3.2 Relinquishing Control of Data*

One major drawback of third-party cloud storage services is

that users surrender control of their data to these providers. Consequently, they also relinquish control over the privacy settings of their data. Since data backups are often done in real-time, there is a possibility of inadvertently sharing data that was not intended to be shared initially.

Moreover, it is important to note that the party entrusted with one's data is primarily motivated by profit. As a result, they may make decisions that align with their interests, potentially undermining the user's business model. For example, changes in the Google algorithm have negatively impacted numerous internet marketing companies.

### *2.3.3 Mismanagement Data*

The Cambridge Analytica scandal involving Facebook exemplifies how third parties can mismanage client data. Aleksandr Kogan, a data scientist affiliated with Cambridge University, designed an application named "This is Your Digital Life" and subsequently shared it with Cambridge Analytica. The app was originally intended for academic research, but due to Facebook's design, it was able to collect not only users' personal information but also information about their connections. Consequently, Cambridge Analytica gained access to the personal data of a staggering 87 million Facebook users, including 70.6 million from the United States [23].

The stolen information from Facebook included users' public profiles, page likes, birthdays, current cities, and in some cases, access to their News Feeds, timelines, and messages. The obtained data was detailed enough to create psychographic profiles, enabling targeted advertising to persuade individuals to specific political events. Politicians paid Cambridge Analytica substantial amounts to leverage this breached data for influencing various political activities.

In a separate infamous instance, the company specializing in media analytics, "Deep Roots Analytics," retained data pertaining to roughly 61% of the United States populace on an inadequately secured Amazon cloud server, a situation that persisted for nearly a fortnight. This dataset encompassed details such as names, email and residential addresses, phone numbers, voter IDs, and additional information, as referenced in [23].

### *2.3.4 Centralization of Middleman Role*

To ensure the system's vitality and enhance user access, this program introduces the concept of a system middleman. However, it is important to address the challenge of centralization associated with intermediaries in a decentralized blockchain system. In this system, the intermediary plays a crucial role in system access, protocol upgrades, and data distribution.

To maintain transparency and accountability, the interactions between users and the system middleman, as well as the intermediary's interactions with user data blocks, are recorded on the blockchain. These records are permanently stored and can be accessed and verified by the entire network after being confirmed by network nodes. The immutability of the blockchain ensures that these interactions are preserved and subject to network-wide supervision.

# 4. METHODOLOGY

## 4.1 Data Collection and Sources

To gather relevant data for this research on decentralized storage systems, a systematic data collection approach will be employed. The following outlines the data collection methods and sources utilized.

**Research Papers and Scholarly Articles:** Conduct a comprehensive literature search in academic databases such as IEEE Xplore, ResearchGate, ACM Digital Library, and Google Scholar. Identify research papers and scholarly articles that focus on blockchain-decentralized storage systems. Relevant keywords include “Blockchain-based decentralized storage”. “Distributed file storage”, “Peer-to-peer storage networks”, and “Decentralized storage system”. Extract key information, including theoretical frameworks, methodologies, findings, and limitations, from the selected papers.

**Industry Reports and White Papers:** Explore reports and white papers published by industry experts, research institutions, and blockchain-related organizations. These sources often provide valuable insight into the latest developments, trends, and practical implementations of decentralized storage systems.

**Technical Documentation and Standards:** Refer to technical documentation and standards related to decentralized storage systems. Examples include the InterPlanetary File System (IPFS) documentation [22], Arweave white paper [23], and Ethereum’s Swarm protocol. These sources provide in-depth technical details, protocols, and design principles of decentralized storage systems.

**Case Studies and Projects:** Analyze case studies and projects that have implemented decentralized storage systems. These can be found in academic journals, conference proceedings, or reports from organizations that have deployed or researched on blockchain-decentralized storage solutions. It Extracts information on the design choices, performance metrics, challenges faced, and outcomes of these case studies.

**Online Communities and Forums:** Participate in digital communities, forums, and discussion platforms that are specifically centered around topics related to blockchain technology and decentralized storage systems. Platforms such as Twitter, Reddit, and specialized blockchain forums provide a wealth of knowledge and insights from community members and experts. It Extracts relevant information, discussions, and opinions that contribute to the understanding of decentralized storage systems.

**Blockchain on-chain data:** Utilize blockchain explorers and data analytics platforms to access and analyze on-chain data related to decentralized storage systems. Blockchain platforms like Ethereum, Bitcoin, and others store transactional data and smart contract interactions, data availability, and usage patterns. It extracts relevant data such as transaction details, storage contract interactions, data storage/retrieval requests, and associated timestamps.

**Data Analysis:** After gathering the data from the above sources, analyze and synthesize the information to address the research question and objectives. Use qualitative and quantitative analysis techniques to identify patterns, trends, and common themes in the data

By utilizing the diverse data collection methods and sources, a comprehensive understanding of blockchain decentralized storage systems can be achieved, supporting the research objectives and contributing to the existing body of knowledge in the field.

## 3.2 Research Design and Approach

The research design and approach for studying blockchain decentralized storage systems involve a systematic and structured process to achieve the research objectives. The following outlines the research design and approach for this

research study.

### *3.2.1 Research Design*

The research design for this study is primarily exploratory and descriptive. It aims to gain a comprehensive understanding of blockchain decentralized storage systems, their principles, advantages, challenges, and comparison with cloud centralized storage systems. Additionally, a case study or project analysis will be conducted to provide practical insights into the implementation and outcomes of decentralized data storage systems.

### *3.2.2 Research Approach*

The research approach for this research paper is primarily based on a combination of literature review, data analysis, and case study analysis. The approach involves the following steps:

**Data Collection:** Gather relevant data from diverse sources, including research papers, industrial reports technical documentation, case studies, surveys, and blockchain on-chain data. Employ a systematic data collection process to ensure the inclusion of comprehensive and relevant information.

**Data Analysis:** Analyze the collected data using qualitative and quantitative analysis techniques. This may involve categorizing and organizing the data, identifying patterns, trends, and themes, and conducting statistical analysis where applicable. The analysis will focus on addressing the research questions and objectives defined earlier.

**Case Study/Project Analysis:** Select a representative case study or project in the field of decentralized data storage systems. Analyze the design choices implementation strategies, performance metrics, challenges faced, and outcomes of the selected case study or project. This analysis will provide practical insight into the application and effectiveness of decentralized storage systems.

**Integration and Synthesis:** Integrate the findings from the literature review, data analysis, and case study analysis to form a coherent narrative and framework for understanding blockchain decentralized storage systems. Synthesize the results and draw connections between the research objectives, theories, empirical evidence, and practical implications.

### *3.2.3 Ethical Considerations*

Adhere to ethical considerations throughout the research process. Ensure proper citation and attribution of sources, maintain data confidentiality and privacy, obtain necessary permission for interviews and surveys, and comply with any legal and ethical guidelines related to the use of blockchain on-chain data.

## 3.3 Data Analysis Techniques

It plays a crucial role in interpreting and deriving meaningful insights from the collected data. Here are some data analysis techniques that has been employed.

### *3.3.1 Qualitative Analysis*

**Thematic Analysis:** Identify and analyze recurring themes, patterns, or concepts in the qualitative data collected through interviews, surveys, or textual sources. Categorize and code the data to derive meaningful themes and sub-themes, allowing for a comprehensive understanding of the phenomena under investigation.

**Content Analysis:** Systematically analyze and categorize textual data, such as literature, reports, and online discussion, to identify relevant content related to the research objectives. This technique helps in identifying key concepts, ideas, and perspectives from a wide range of sources.

### *3.3.2 Quantitative Analysis*

**Descriptive Statistics:** Calculate and present summary statistics, such as means, medians, standard deviations, or frequencies, to describe the characteristics of the collected quantitative data. This technique provides a concise overview of the data and facilitates comparisons and generalizations.

**Data Visualization:** Utilize visual representations, such as charts, graphs, or diagrams, to summarize and present the quantitative data in a concise and easily understandable format. Data visualization techniques enhance the interpretability of the findings and aid in identifying trends or patterns.

### *3.3.3 Comparative Analysis*

Conduct a comparative analysis to examine similarities and differences between different cases, systems, or approaches. This technique helps in identifying factors that contribute to the success or challenges of decentralized storage systems, providing insights into best practices or areas for improvement.

# 4. MAIN DISCUSSIONS

## 4.1 Principles and Architecture of Decentralized Data Storage System

### *4.1.1 IPFS*

IPFS [22], or Inter Planetary File System, represents a peer-to-peer (P2P) hypermedia protocol designed to herald a revolutionary transformation in the realm of the internet. This distributed system empowers users to securely store and retrieve various data types, including files, applications, and websites. On a larger scale, IPFS aspires to establish a global computer network that ensures privacy, security, and immunity against censorship.

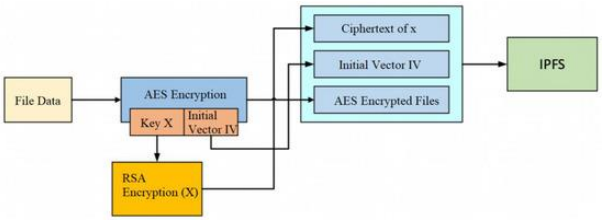


**Fig 4: Encryption Process [6]**

The content hosted on IPFS encompasses a diverse range of types and categories, spanning databases, websites, media files, documents, and applications. To access any content stored within the IPFS network, users simply need to enter a corresponding "link." This process mirrors the familiar experience of accessing a webpage through its URL.

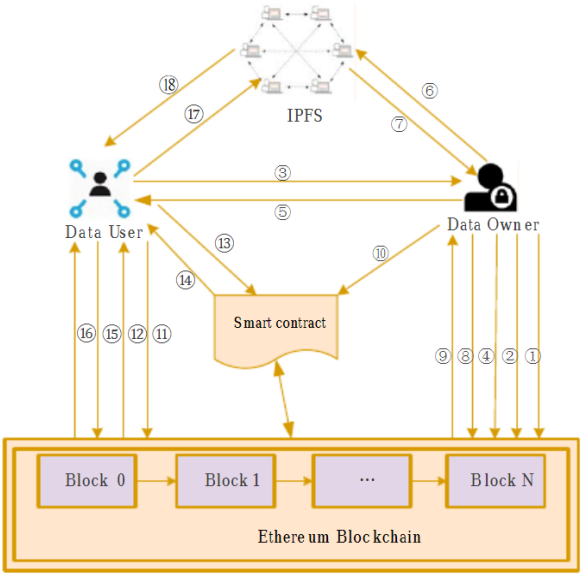


**Fig 5: System Framework**

Much like blockchain networks that rely on nodes to harness computing power for data verification, IPFS leverages hundreds of thousands of nodes, each contributing its storage bandwidth to accommodate the network's data storage requirements.

For those unfamiliar with the concept, nodes represent individual computer systems that collectively constitute the IPFS network. In essence, IPFS is capable of performing all the functions of centralized web2 platforms [24], but without the need for centralized data storage. This decentralized approach ensures greater privacy, security, and censorship resistance.

### *4.1.2 System model*

As outlined in the system model described in citation, decentralized storage is an innovative strategy that deviates from conventional reliance on centralized entities for data storage. Instead, it capitalizes on blockchain-driven decentralized networks to distribute data across numerous nodes. This decentralized paradigm significantly amplifies security, dependability, and robustness by facilitating data dissemination and safeguarding against errors and potential vulnerabilities.

Moreover, the decentralized storage framework empowers users with absolute ownership and authority over their data, thereby obviating the necessity for dependence on third-party intermediaries for data management and preservation. Notable instances of decentralized storage systems encompass the Inter Planetary File System (IPFS) and StorX.

The architecture of a decentralized data storage system comprises two key entities:

1. Data Owner (DO): The DO represents an individual or organization that possesses a collection of files intended for sharing.
2. Data User (DU): The DUs are clients authorized by the DO to access specific files.

While this report focuses primarily on the DO and DU entities, it does not encompass the validators on the Ethereum blockchain or the storage nodes. The system model operates through a series of steps, as illustrated in **Figure 5**.

1. DO initialized the system by encrypting the system masker key and embedding it into an Ethereum transaction.
2. DO employs a smart contract on the Ethereum blockchain.
3. DU initiates a registration request to DO.
4. DO generates a secret key for DU, encrypts it using the shared key, and embeds the encrypted secret key into an Ethereum transaction.
5. DO securely transmits the transaction ID, smart contract address, smart contract ABI, and smart contract source code to DU.
6. DO selects a keyword set from the shared file, encrypts the file using the AES algorithm, and uploads it to IPFS.
7. DO records the returned file location provided by IPFS.
8. DO encrypt the file location using selected AES key K, applies an ABE algorithm to encrypt AES key K, and further encrypts this information using a selected AES key K1 embedded into an Ethereum transaction.
9. DO keeps track of the Ethereum transaction ID and AES key K1.
10. DO generates encrypted keyword indexes and stores them in the smart contract.
11. DU retrieves transaction data associated with the secret key from the Ethereum blockchain.
12. DU decrypts the transaction data using the shared key to obtain the secret key.
13. DU generates a search token and invokes the smart contract.
14. The smart contract performs a search based on the token and returns the relevant results.
15. DU reads relevant transaction data based on the search results returned by the smart contract.
16. DU decrypts the transaction data.
17. DU downloads the encrypted file from IPFS.
18. DU decrypts the encrypted file.

### *4.1.3 Decentralized Storage System Smart Contract Design*

This section is dedicated to introducing the smart contract-related interfaces and algorithmic logic employed in this study, with the code implementation derived from [22]. In the context of the Ethereum network, the development of smart contracts involves the utilization of the solidity programming language [25]. These smart contracts are designed to encompass distinct variables and functions that maintain a continuous presence in the global namespace, serving as pivotal conduits for essential blockchain information.

The following special variables are mainly used for decentralized data storage systems:

**msg.sender** refers to the sender of the current message or transaction being executed. When the smart contract is initially deployed, **msg.sender** represents the address of the contract creator. However, during subsequent function calls to the smart contract, **msg.sender** will reflect the address of the caller of the smart contract.

**msg.value** indicates the amount of wei (the smallest unit of Ether) sent along with the message or transaction. For convenience, **$msg.value** is used to represent the specific amount of wei attached to a message, and **$cost** denotes a fixed value in wei. Notably, 1 ether is equal to 10^18 wei.

On the other hand, **tx.origin** refers to the sender of the entire transaction, encompassing the full call chain. In a situation where an external account, known as an Externally Owned Account (EOA), calls the smart contract and another smart contract is subsequently invoked within the original contract, a call chain is established, with **tx.origin** representing the EOA initiating the transaction.

### *4.1.3.1 dataSharing Contract*

The **dataSharing** contract, deployed by the data owner, will serve as the central smart contract for facilitating data-sharing activities. Through this contract, authorized parties can securely access and interact with the data owner's shared information.

**dataSharing** contract Initialization: During contract creation, several variables are defined as follows.

1. The **dataOwner** variable of address type, which represents the address of the data owner (DO).
2. The **authorizedUsers** variable of mapping type establishes a mapping collection from authorized user addresses to boolean values. The data owner can add, modify, or delete entries within this collection through the relevant function interfaces of the contract.
3. The **Index** variable of mapping type, which defines a mapping collection from encrypted keyword indices to related information. The data owner possesses the authority to add, modify, or delete entries in this collection, while authorized users can access and read the contents through relevant function interfaces of the smart contract.

The **dataSharing** contract primarily offers the following seven function interfaces:

**addUser(newUserAddress)** is limited to execution by the contract's creator (Data Owner). Whenever a user sends a registration request to DO, accompanied by their identity certificate (authenticated through a secure out-of-band channel), the DO authorizes the user's Externally Owned Account (EOA) using this function.

**Algorithm 1:** addUser

```
Input: newUserAddress
Output: bool
1 if msg.sender is not dataOwner then
2     throw;
3 end
4 if newUserAddress exists then
5     return false;
6 else
7     authorizedUsers[newUsersAddress] <= true;
8     return true;
9 end
```

**removeUser(oldUserAddress)** is exclusively accessible to the contract creator (Data Owner). When there is a necessity to remove a user from the authorized set, the DO accomplishes this task by providing the user's EOA as an argument to the function.

**Algorithm 2:** removeUser

```
Input: oldUserAddress
Output: bool
1 if msg.sender is not dataOwner then
2     throw;
3 end
4 if oldUserAddress hasn't existed then
5     return false;
6 else
7     authorizeUser[oldUserAddress] <= false;
8     return true;
```

**addIndex(keywordIndex, txid, key1)** function is only executable by the contract's creator (Data Owner). Whenever DO upload new files to IPFS, a selection of keyword sets is made from each file, leading to the creation of corresponding encrypted keyword indices. These encrypted keyword indices are then stored in the smart contract. The function requires three arguments: encrypted keyword indices (**keywordIndex**), the transaction ID (**txid**), and the encryption key (key1).

**deleteFile(keywordIndex, txid)** function can only be executed by the contract's creator (DO). In situations where the need arises to delete a specific file, the encrypted keyword indices (**keywordIndex**) associated with the file and the corresponding transaction ID (**txid**) are provided as arguments to the function.

**Algorithm 3:** addIndex

```
Input: keywordIndex, txid, key1
Output: bool
1 if msg.sender is not dataOwner then
2     throw;
3 end
4 mapping keywordIndex to (txid, key1), and add it to
5     Index variable collection
6 return true;
```

**deleteKeyword(keywordIndex)** function is restricted to execution by the contract's creator (DO). When there arises a need to delete all files corresponding to a specific keyword, the function takes the encrypted keyword indices (**keywordIndex**) as input.

**search(keywordIndex)** function can be executed solely by the user within the authorized set and the contract's creator (DO). The function accepts the encrypted keyword indices (**keywordIndex**) and returns a set of transaction IDs and keys associated with the **keywordIndex**.

**withdraw()** function is solely executable by the contract's creator (DO). DO can withdraw the search service fee paid by the user using this function.

**Algorithm 4:** deleteFile

```
Input: keywordIndex, txid
Output: null
 1 if msg.sender is not dataOwner then
 2     throw;
 3 end
 4 get Index[keywordIndex] array's length len
 5 if len equals 0 then
 6     return;
 7 else
```

```
    for i <= 0 to len-1 do
        if Index[keywordIndex][i].txid equal txid then
            for j <= i+1 to len-1 do
                Index[keywordIndex][j-1]
                    <= Index[keywordIndex][j]
            end
            delete Index[keywordIndex][len -1]
            break;
        end
    end
end
```

**Algorithm 5:** deleteKeyword

```
Input: keywordIndex
Output: null
if msg.sender is not dataOwner then
    throw;
end
getIndex[keywordIndex] array's length len
if len equals 0 then
    return;
else
    delete Index[keywordIndex]
end
```

**Algorithm 6:** search

```
Input: keywordIndex
Output: searchResult
if tx.origin is not dataOwner and $msg.value < $cost
    then
    throw;
end
get Index[keywordIndex] array's length len;
if tx.origin is not dataOwner then
    if len equals 0 then
        send $msg.value to msg.sender;
        searchResult <= null;
    else
        send $cost to dataSharing contract address;
        send $msg.value - $cost to msg.sender;
        searchResult <= Index[keywordIndex];
    end
else
    searchResult <= Index[keywordIndex];
end
return searchResult;
```

**Algorithm** 7: withdraw

```
Input: null
Output: null
if msg.sender is not dataOwner then
    throw;
end
if the contract's balance > 0 ether then
    send the contract's balance to msg.sender;
end
```

### *4.1.3.2 Data User Contract*

In the Ethereum smart contract, the return value of a non-constant function can only be obtained through log events. Consequently, in the aforementioned dataSharing contract, the search results returned by the search function can only be accessed through events. However, relying solely on events to obtain search results poses security risks, as Ethereum events are publicly viewable, allowing anyone to listen in and obtain some results effortlessly.

To tackle this challenge, another smart contract, deployed by a data user has been devised. The data user invokes the search function of the **dataSharing** contract and saves the search results in this new contract, referred to as the dataUser contract. By adopting this approach, only the data user holds the privilege to view the search results, effectively resolving the security concern associated with event-based retrieval.

During the initialization of the **dataUser** contract, the following variables are defined:

1. A **dataSharing** contract object instance is initialized to enable the invocation of the search function in the **dataSharing** contract.
2. The "owner" variable of address type represents the address of the Data User (DU).
3. The **searchResult** variable of a struct type is used to store the search results.

The dataUser contract offers the following three function interfaces:

1. **deposit(value)**: This function allows ether to be deposited into the **dataUser** contract. The contract's balance is used to cover the cost of invoking the search function in the **dataSharing** contract.
2. **dataSearch(keywordIndex)**: Only the contract creator (DU) can execute this function. It takes the encrypted keyword indices (**keywordIndex**) as a function argument.
3. **getResult()**: This function is annotated with the keyword "view," indicating that it solely performs read-only operations and does not alter the blockchain's state. Only the contract creator (DU) can execute this function.

**Algorithm 8**: deposit

```
Input: deposit value
Output: null
if msg.value does not equal the deposit value then
    throw;
end
send $value to dataUser contract address
```

**Algorithm 9**: dataSearch

```
Input: keywordIndex
Output: null
if msg.sender is not the owner then
    throw;
end
call dataSharing contract's search();
save search result to struct searchResult;
```

**Algorithm 10**: getResult

**Input**: null
**Output**: searchResult
1 **if** msg.sender is not the owner **then**
2 throw;
3 **end**
4 **return** searchResult;

This function allows the Data User (DU) to locally retrieve the search results without revealing the process to others. It ensures that the search results are private and only accessible by the DU. Other users or external parties cannot view the details of this retrieval process.

## 4.2 Advantages and Challenges of Decentralized Data Storage Systems

Blockchain-based decentralized data storage systems present a paradigm shift in data management, offering a myriad of advantages that outperform traditional centralized storage solutions. These advantages can be attributed to the fundamental principles and features embedded within blockchain technology, emphasizing its decentralized nature and innovative data storage capabilities. Nevertheless, as with any transformative technology, decentralized data storage systems come with a set of challenges that necessitate strategic solutions for unleashing their full potential.

### *4.2.1 Advantages*

**Cost-Efficient Data Management:** One notable advantage of Decentralized Data Storage Systems lies in their inherent cost-efficiency [26]. By leveraging distributed networks and minimizing reliance on centralized infrastructure, these systems can potentially reduce operational expenses and resource-intensive maintenance. This can translate into optimized resource allocation and competitive pricing models, enhancing the overall cost-effectiveness of data storage.

**Enhanced Data Security:** The decentralized architecture of these storage systems engenders an elevated level of data security. Data is fragmented and distributed across numerous nodes, significantly reducing the risk of a single point of failure or data breach [26]. This fragmentation, combined with robust encryption protocols, contributes to heightened data privacy and protection against unauthorized access.

**Reliability and Fault Tolerance:** Decentralized storage solutions underscore reliability by embracing redundancy and fault tolerance [27]. Data duplication and distribution across diverse nodes ensure that even in the face of hardware failures or network disruptions, data accessibility and integrity remain intact. This resilience is crucial for businesses reliant on consistent and uninterrupted data availability.

**Resistance to Censorship and Control:** A distinctive feature of Decentralized Data Storage Systems is their resilience against censorship and external control. By dispersing data across a distributed network, these systems mitigate the risk of data manipulation or censorship attempts by a single authority [27]. This attribute is particularly valuable in scenarios where data integrity and accessibility need to be safeguarded from external influences.

**Empowerment of Data Freedom:** Decentralized storage empowers users with a newfound sense of data autonomy and freedom. Unlike centralized models, where data may be subject to proprietary constraints, decentralized systems enable seamless movement and access of data without lock-in or restrictions [27]. Users retain greater control over their data, facilitating data portability and enabling more flexible data management strategies.

Incorporating these qualities, Decentralized Data Storage Systems present a paradigm shift in data management, aligning with contemporary IT imperatives for security, reliability, and cost optimization.

### *4.2.2 Challenges*

**Integration and User Experience (UI/UX):** Integrating decentralized data storage systems seamlessly with existing applications and workflows can pose a challenge. The development of intuitive and user-friendly interfaces that facilitate data migration, retrieval, and management across these systems may require substantial effort [27]. Ensuring a smooth user experience while transitioning between traditional and decentralized storage paradigms demands careful consideration of design, functionality, and interoperability.

**Limited Compute Capability:** While decentralized data storage excels in data distribution and retrieval, it often lacks the computational capabilities found in traditional centralized systems [28]. This limitation can hinder the execution of complex data processing tasks directly within the decentralized storage environment. Integrating robust compute capabilities within decentralized storage networks is a technical challenge that requires innovative solutions to expand the range of use cases.

**Performance Assurance and Scalability:** The decentralized nature of storage systems can introduce variability in performance and scalability. As data is distributed across multiple nodes, ensuring consistent and predictable performance levels can be intricate. Balancing the load across the network, optimizing data retrieval speeds, and maintaining responsiveness as the system scales demand ongoing monitoring, optimization, and infrastructure enhancements.

**Self-Service Capabilities:** Decentralized storage systems may lag in offering comprehensive self-service capabilities that are prevalent in centralized counterparts [27]. Implementing user-friendly tools for provisioning, monitoring, and managing storage resources within decentralized networks can be a challenge. Providing users with efficient control over their data, adjusting storage configurations, and ensuring transparent billing mechanisms require meticulous design and development.

Addressing these challenges necessitates a combination of technical innovation, strategic planning, and user-centric design. As the decentralized data storage landscape evolves, solutions that enhance integration, performance, and user accessibility will play a pivotal role in realizing the full potential of this transformative paradigm.

## 4.2 Comparison of Decentralized Data Storage Systems with Centralized Data Storage Systems

In the rapidly evolving landscape of data storage, two contrasting paradigms Decentralized Data Storage System and Centralized Storage System, have emerged as dominant contenders. These systems embody distinct philosophies and architectural approaches, each with its unique set of advantages and challenges. This short comparative analysis from **Table 1** aims to shed light on the fundamental differential differences between these two approaches, offering insights into their implications for data security, control, scalability, and more.

**Table 1: Decentralized Data Storage vs. Centralized Data Storage**

| | Centralized Data Storage | Decentralized Data Storage |
|---|---|---|
| **Strength** | Efficient Operations<br>Seamless Integration<br>Enhanced Gateway<br>User Self-services Options | Cost-effectiveness<br>Robust Security<br>Consistent Reliability<br>Censorship-resilence<br>Flexible Data Mobility |
| **Weakness** | Potential Vendor Dependency<br>Uncertain Expenses<br>Single Point of Failure<br>Privacy and Censorship Concerns | Integration and User Experience Challenges<br>Limited Data Processing Capability<br>Performance Assurance<br>Maturing Self-service Capabilities |

When comparing Centralized Data Storage with Decentralized Data Storage, distinct advantages and disadvantages emerge for each approach. Centralized storage, exemplified by established providers like Amazon AWS, showcases strengths in operational excellence gained through years of refinement in user interfaces and ancillary services. It seamlessly integrates with a plethora of applications such as analytics, data lakes, ERPs, CRMs, and DevOps tools, offering standardized and predictable performance guarantees due to its controlled infrastructure. The centralized model also provides a head start in enabling self-service options while often being more cost-effective compared to traditional cloud providers.

On the other hand, Centralized Data Storage has its share of weaknesses. Vendor lock-in poses a challenge, potentially making data migration cumbersome. The unpredictability of costs, especially concerning bandwidth and API usage, can lead to unexpected expenses. The centralized architecture's single point of failure susceptibility to hacks, attacks, or outages raises concerns about data security. Additionally, the risk of censorship and privacy breaches looms, where governments may demand data access from centralized providers.

In contrast, Decentralized Data Storage presents strengths that address some of these concerns. Its distribution of data across a network of nodes enhances security and reliability by eliminating single points of failure. It ensures data privacy and integrity by being censorship-resistant, and immune to governmental pressures. Moreover, it offers data freedom, enabling easy movement across providers without lock-ins or premium costs, and can provide cost efficiencies, particularly when compared to the potentially higher premiums charged by traditional providers.

Nonetheless, Decentralized Data Storage comes with its set of weaknesses. It currently lacks the capability for efficient querying and computation directly on the stored data, limiting its applicability for certain use cases. Performance variability is a challenge due to the differing capacities and locations of storage hosts (miners), making standardized performance guarantees complex. Self-service capabilities are still under development, impacting user-friendliness.

In summary, the comparison reveals a trade-off between centralized operational excellence and cost-effectiveness, and the decentralized model's enhanced security, data freedom, and potential cost savings. The choice between the two hinges on specific use cases, performance needs, and data sensitivity. As both paradigms evolve, ongoing technological advancements are likely to shape the future of data storage solutions by addressing these strengths and weaknesses.

# 5. RESULT AND ANALYSIS

## 5.1 Data Analysis and Findings

### *5.1.1 Sample Characteristics*

The purpose of this subsection delves into the distinctive attributes of the sample, sourced from existing studies on blockchain decentralized data storage. These sample characteristics are integral to comprehending the contextual framework of the study's outcomes, particularly regarding privacy, cost, speed, user dynamics, and related aspects.

**Sample Size and Scope:** The sample encompasses a selection of participants, data points, or entities drawn from diverse studies on blockchain decentralized data storage. This choice of data scope is instrumental in capturing a multifaceted perspective on privacy, cost-effectiveness, speed, user behavior, and associated dimensions.

**Privacy Focus and Demographics:** Emphasizing the privacy dimension, the sample encompasses studies that shed light on user demographics, privacy concerns, and data protection measures. This lens provides insights into how blockchain-decentralized data storage addresses privacy-related challenges.

**Cost and Economic Implications:** Incorporating studies with a focus on cost considerations, the sample underscores the economic dynamics associated with blockchain-decentralized data storage. This characterization enhances the understanding of the financial implications of adopting such systems.

**Speed and Efficiency Analysis:** The sample comprises studies that examine the speed and efficiency aspects of blockchain decentralized data storage. These insights contribute to assessing the system's responsiveness and its potential advantages in comparison to traditional solutions.

**User Behavior and Interaction Patterns:** By including studies that delve into user behavior and interaction patterns within blockchain-decentralized data storage systems, the sample captures valuable insights into user experiences, preferences, and engagement trends.

**Geographic and Sectoral Representation:** The sample features studies from diverse geographic regions and sectors, fostering a holistic perspective on the applicability and relevance of blockchain-decentralized data storage solutions across different contexts.

**Temporal Context:** The time frame during which the constituent studies were conducted adds a temporal dimension, enabling the examination of how privacy, cost, speed, user dynamics, and other factors have evolved over time.

**Inclusion Criteria and Research Methodologies:** Explicitly highlighting the inclusion criteria and methodologies employed in the constituent studies ensures transparency and consistency

in the data selection process.

By characterizing the sample based on existing studies on blockchain decentralized data storage, this section provides a comprehensive framework for interpreting the subsequent data analysis and findings. The aggregated insights from these diverse sources enable a robust exploration of the key themes of privacy, cost, speed, user behavior, and more within the context of decentralized data storage systems.

### *5.1.2 Descriptive Analysis*

Descriptive analysis, often referred to as descriptive analytics or descriptive statistics, involves utilizing statistical methods to portray or condense a dataset. In the area of data analysis, descriptive analysis stands as a significant approach, valued for its capacity to distill comprehensible perspectives from data that might otherwise remain uninterpreted.

In contrast to alternative modes of data analysis, descriptive analysis refrains from forecasting future outcomes. Its focus rests solely on gleaning insights from historical data, a process that involves transforming the data to reveal its inherent significance.

This section offers insights into the key characteristics, trends, and patterns observed in the purview of blockchain-based decentralized data storage systems. Through a meticulous examination of trendlines, comparative tables, and insightful charts, this section aims to unveil a detailed and nuanced understanding of the data landscape.

#### *5.1.2.1 Features and Technical*

Amid the scope of data storage systems, a diverse array of providers has emerged, each offering a unique blend of features and technical capabilities. This section delves into a comprehensive analysis of the key attributes that distinguish these providers, providing a detailed exploration of their market cap, consensus algorithm, data replication and retrieval mechanisms, encryption protocols, smart contract execution, and minimum hosting requirements.

In the realm of decentralized storage, comprehending the technical foundations of various providers is crucial. This comparison highlights key features that illuminate operational methods, security, and performance. By analyzing these aspects, readers can gain insight into the diverse approaches taken by different providers. The subsequent sections delve deeper into specific features and technical details, offering a comprehensive understanding of each provider's offerings.

**Market Cap**

The market capitalization (market cap) is a key financial metric that represents the total value of a cryptocurrency or token in circulation. It is calculated by multiplying the current price per unit by the total number of units in circulation. In the context of the decentralized storage providers listed in the table, the market cap reflects the overall valuation of each project, indicating its perceived value within the market. A higher market cap generally signifies a larger user base, investor confidence, and resources available for development and expansion.

Each entry in the "Current Market Cap" column denotes the approximate total valuation of the respective decentralized storage provider. This value provides an insight into the popularity and perceived significance of each project in the realm of decentralized data storage. It is important to note that market cap can fluctuate significantly over time due to factors such as market sentiment, adoption rates, technological advancements, and regulatory developments.

Comparing the market caps of different providers can help assess the level of interest and investment in each project. Additionally, it can serve as an indicator of how well a particular provider may be positioned to drive innovation, withstand market fluctuations, and attract users and stakeholders.

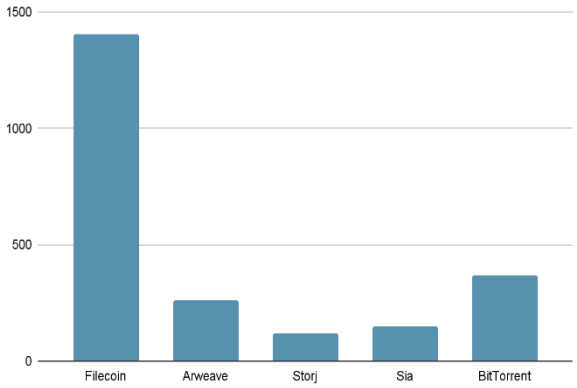


**Chart 2: Current Market Capitalization of Decentralized Storage Projects (in Million USD)**

**Consensus Algorithm**

The consensus algorithm is a fundamental component of blockchain and decentralized networks that determines how transactions are validated, added to the blockchain, and agreed upon by network participants. Different consensus algorithms are designed to ensure the security, integrity, and decentralization of the network while addressing challenges like double-spending and achieving agreement among nodes.

1. Filecoin: Filecoin employs a unique combination of Proof of Spacetime (PoSt) and Proof of Replication (PoR). PoSt ensures that storage providers are dedicating actual physical space to storing data, while PoR verifies the replication of stored data across the network.
2. Arweave: Arweave utilizes a consensus algorithm known as Succinct Proof of Random Access (SPoRA). SPoRA focuses on providing secure and scalable access to stored data while maintaining a high level of decentralization.
3. Filecoin: Filecoin employs a unique combination of Proof of Spacetime (PoSt) and Proof of Replication (PoR). PoSt ensures that storage providers are dedicating actual physical space to storing data, while PoR verifies the replication of stored data across the network.

**Table 2: Features and Technical Comparison of Decentralized and Centralized Data Storage Providers**

| | Current Market Cap | Consensus Algorithm | Data Replication & Retrieval | Encryption | Smart Contract Execution | Minimum Hosting Requirements |
|---|---|---|---|---|---|---|
| Filecoin | ~$1.52B | Proof of Spacetime (PoSt) & Proof of Replication (PoR) | Users determine number of replicated copies | Users choose encryption of stored data | Utilizes Filecoin Virtual Machine (FVM) | CPU: 8 cores RAM: 137GB Hard Drive: 1.1TB |
| Arweave | ~$546M | Succinct Proof of Random Access (SPoRA) | Data stored by miners, replicated over 16 times | Users choose encryption of stored data | 'Lazy' SmartWeave contracts executed and validated by users, not the network | CPU: 6 cores RAM: 8.6GB Hard Drive: 4TB |
| Storj | ~$81M | Proof of Availability (PoA) | Data split into 80 pieces, 29 needed for retrieval | Automatically encrypted with AES-256 algorithm | No smart contract capability | CPU: 1 cores RAM: 2GB Hard Drive: 550GB |
| Sia | ~$214M | Proof of Work (PoW) | Data split into 30 pieces, 10 needed for retrieval | Automatically encrypted with Threefish algorithm | File contracts enforce agreements | CPU: 4 cores RAM: 8G Hard Drive: 64GB |
| BitTorrent | ~$851M | Proof of Stake (PoS) | Data split into 30 pieces, 10 needed for retrieval | Users choose encryption of stored data | Utilized BitTorrent-Chain Virtual Machine (BTTCVM) | CPU: 1 cores RAM: 1GB Hard Drive: 32GB |
| Amazon S3 | ~$1.5T | N/A | Users select files to replicate within regions | Users can enable server-side encryption using AES-256 algorithm | N/A | N/A |

4. Filecoin: Filecoin employs a unique combination of Proof of Spacetime (PoSt) and Proof of Replication (PoR). PoSt ensures that storage providers are dedicating actual physical space to storing data, while PoR verifies the replication of stored data across the network.
5. Arweave: Arweave utilizes a consensus algorithm known as Succinct Proof of Random Access (SPoRA). SPoRA focuses on providing secure and scalable access to stored data while maintaining a high level of decentralization.
6. Storj: Storj uses Proof of Availability (PoA) to ensure that storage providers maintain a high level of uptime and accessibility for users' stored data.
7. Sia: Sia relies on Proof of Work (PoW), similar to the algorithm used by Bitcoin. PoW requires participants, known as miners, to solve complex mathematical problems to validate transactions and secure the network.
8. BitTorrent: BitTorrent employs a Proof of Stake (PoS) consensus algorithm. PoS validates and creates new blocks based on the amount of cryptocurrency held by participants, rather than computational work.
9. Amazon S3: Amazon S3 is a centralized cloud storage service and does not rely on a blockchain-based consensus algorithm.

The choice of consensus algorithm significantly impacts the security, efficiency, and scalability of decentralized storage solutions. Projects like Filecoin and Arweave introduce innovative algorithms tailored to the specific needs of data storage and retrieval, aiming to optimize space usage and access. On the other hand, projects like Sia and BitTorrent leverage established algorithms like PoW and PoS, emphasizing security and energy efficiency.

The consensus algorithm also influences factors such as transaction speed, network decentralization, and resource requirements. While PoW and PoS are well-established and widely used, newer algorithms like SPoRA and PoSt offer promising alternatives with potential advantages.

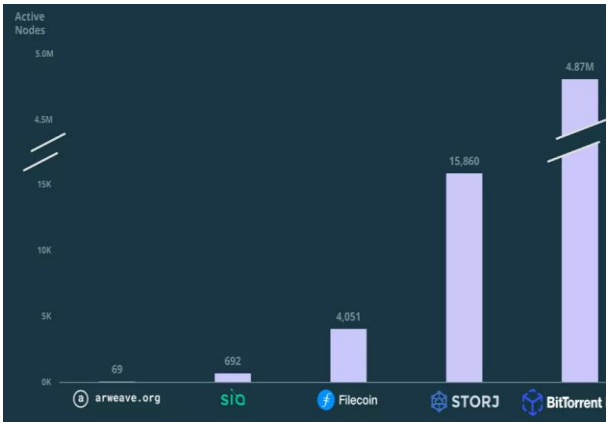


**Chart 3: Active Nodes**

Nonetheless, as indicated by [28], not all decentralized storage services exhibit true decentralization. Certain nodes may necessitate significantly elevated technical or hardware prerequisites, exemplified by Filecoin and Arweave. Distinct systems will entail diverse trade-offs. Ultimately, the extent to which users' data is fragmented and extensively distributed underpins the censorship-resistant nature of decentralized storage services.

In summary, understanding the consensus algorithms used by different decentralized storage projects is crucial for assessing their reliability, security, and overall performance in delivering efficient and secure data storage solutions. Each algorithm has its strengths and weaknesses, catering to different use cases and priorities within the realm of decentralized data storage.

**Data Replication & Retrieval:**

The process of data replication and retrieval plays a crucial role in decentralized data storage systems, determining how data is stored, duplicated, and accessed across the network. This aspect directly impacts data availability, durability, and efficiency.

1. Filecoin: Users have the authority to determine the number of replicated copies their data will have on the network. This approach allows users to balance redundancy and storage costs based on their individual needs.
2. Arweave: Miners on the Arweave network store data, and each piece of data is replicated over 16 times across the blockweave. This extensive replication ensures high data availability and durability.
3. Storj: Data on Storj is divided into 80 pieces using Reed-Solomon erasure coding. For data retrieval, only 29 out of the 80 pieces are needed. This approach optimizes retrieval efficiency and data integrity.
4. Sia: Sia divides data into 30 pieces, with only 10 pieces required for retrieval. This strategy balances data availability with network efficiency.
5. BitTorrent: Similar to Sia, BitTorrent splits data into 30 pieces, and only 10 pieces are needed for retrieval. This redundancy ensures data availability and efficient retrieval.
6. Amazon S3: Users of Amazon S3 have the option to select specific files for replication within different geographic regions. This approach allows users to tailor data replication to their geographical requirements.

The data replication and retrieval strategies employed by different projects reflect their emphasis on data redundancy, accessibility, and retrieval efficiency. Projects like Arweave and Storj implement robust replication mechanisms to ensure data availability even in the face of node failures. On the other hand, projects like Sia and BitTorrent strike a balance between redundancy and retrieval efficiency by dividing data into smaller pieces.

The choice of data replication strategy directly influences factors such as data durability, retrieval speed, and network performance. Some projects allow users to customize replication levels, enabling them to optimize data redundancy based on their specific needs. Others rely on innovative erasure coding techniques to achieve data integrity and efficient retrieval.

To sum up, understanding how different decentralized storage projects approach data replication and retrieval is essential for evaluating their ability to maintain data availability, accessibility, and reliability. Each approach offers a unique combination of redundancy and efficiency, catering to diverse user preferences and uses cases within the realm of decentralized data storage.

**Encryption**

The application of encryption techniques is a fundamental component of data security within decentralized data storage systems. Encryption safeguards data by converting it into a secure and unreadable format, ensuring its confidentiality and protection against unauthorized access.

1. Filecoin: Users have the autonomy to choose whether to encrypt their stored data. This approach empowers users to decide the level of security for their data.
2. Arweave: Similar to Filecoin, Arweave allows users to choose whether to encrypt their stored data. This user-driven encryption strategy ensures data privacy.
3. Storj: Data stored on Storj is automatically encrypted using the AES-256 encryption algorithm. This robust encryption ensures the confidentiality and integrity of stored data.
4. Sia: Similarly, Sia employs the Threefish encryption algorithm to automatically encrypt data. This encryption mechanism enhances data security and protection.
5. BitTorrent: Users of BitTorrent have the flexibility to choose whether to encrypt their stored data, providing them with control over data privacy.
6. Amazon S3: Users on Amazon S3 can enable server-side encryption using the AES-256 encryption algorithm. This encryption at the server level adds an extra layer of protection to stored data.

The encryption strategies adopted by different decentralized storage projects demonstrate their commitment to ensuring data security and privacy. While Filecoin, Arweave, and BitTorrent give users the freedom to decide on encryption, Storj and Sia prioritize automatic encryption using robust algorithms like AES-256 and Threefish. Amazon S3's server-side encryption offers an additional layer of protection.

The choice of encryption method directly influences the confidentiality and integrity of stored data. Automatic

encryption adds a layer of security by ensuring that data is consistently protected, regardless of user choices. On the other hand, user-driven encryption empowers individuals to customize their data security according to their preferences.

In conclusion, understanding the encryption mechanisms employed by different decentralized storage projects is crucial for assessing their dedication to data security and user privacy. Each project adopts an encryption approach that aligns with its values and priorities, offering users a spectrum of options to protect their data within the realm of decentralized data storage.

**Smart Contract Execution**

Smart contracts play a pivotal role in enabling automation and self-executing agreements within decentralized data storage systems. These contracts are programmable scripts that facilitate interactions and transactions between participants on the network.

1. Filecoin: Filecoin employs the Filecoin Virtual Machine (FVM) to execute smart contracts. This virtual machine enables the execution of predefined code, facilitating various automated functions and interactions within the network.
2. Arweave: Arweave's approach to smart contracts is characterized by 'Lazy' SmartWeave contracts. In this model, contracts are executed and validated by users rather than the network itself. This approach emphasizes user involvement and validation.
3. Storj: Storj does not possess smart contract capabilities, which limits its ability to automate and facilitate self-executing agreements within the system.
4. Sia: Smart contracts on Sia are implemented through file contracts. These contracts enforce agreements between renters and storage providers, ensuring that agreed-upon terms are upheld.
5. BitTorrent: BitTorrent utilizes the BitTorrent-Chain Virtual Machine (BTTCVM) for smart contract execution. This virtual machine enables the execution of code to automate various processes and interactions on the BitTorrent network.
6. Amazon S3: Amazon S3 does not offer smart contract capabilities, which restricts its ability to facilitate automated and programmable agreements.

The differences in smart contract execution methods across decentralized storage projects reflect their varying approaches to automation and self-executing agreements. Filecoin's utilization of the FVM enables a comprehensive smart contract environment, while Arweave's 'Lazy' SmartWeave contracts emphasize user validation. Sia's file contracts and BitTorrent's BTTCVM offer distinct approaches to contract enforcement, enhancing the integrity of agreements.

In contrast, projects like Storj and Amazon S3 lack smart contract capabilities, limiting their potential for automation and programmability within their ecosystems.

The availability and sophistication of smart contract execution mechanisms influence the versatility and utility of decentralized data storage systems. The adoption of different smart contract models reflects the priorities and design philosophies of each project, ultimately shaping the scope of automation and interactions within their respective networks.

**Minimum Hosting Requirements**

The technical specifications and minimum hosting requirements for decentralized storage projects play a vital role in determining the hardware needed to participate in their networks. These requirements define the computing resources necessary for storage providers to contribute to the system's operations.

1. Filecoin: To participate in the Filecoin network, storage providers need a robust infrastructure. The minimum hosting requirements include a CPU with 8 cores, a substantial RAM of 137GB, and a hard drive capacity of 1.1TB. These requirements highlight the need for considerable computational power and storage capacity.
2. Arweave: Arweave's minimum hosting requirements are more moderate, necessitating a CPU with 6 cores, 8.6GB of RAM, and a larger hard drive capacity of 4TB. These specifications reflect a balance between computational power and storage space.
3. Storj: Storj's hosting requirements are relatively lower, with a minimum CPU configuration of 1 core, 2GB of RAM, and a hard drive capacity of 550 GB. These requirements indicate a more accessible entry point for storage providers.
4. Sia: Sia's minimum hosting requirements include a CPU with 4 cores, 8GB of RAM, and a hard drive capacity of 64 GB. These specifications emphasize a balance between computational capabilities and storage capacity.
5. BitTorrent: BitTorrent's hosting prerequisites are relatively modest, necessitating a CPU with 1 core, 1GB of RAM, and a hard drive capacity of 32 GB. These minimal requirements enable a broader range of participants to engage in the network.
6. Amazon S3: Amazon S3 does not specify minimum hosting requirements, as it is a centralized cloud storage service that abstracts hardware considerations from users.

The variance in minimum hosting requirements across decentralized storage projects reflects their diverse technical architectures and priorities. While Filecoin demands substantial computational resources and storage capacity, projects like Storj and BitTorrent prioritize accessibility by imposing more lenient requirements.

Arweave's specifications strike a balance between computational power and storage capacity, accommodating a middle ground. Sia's requirements align with a balance between computational capabilities and storage capacity, emphasizing a more balanced approach.

These hosting prerequisites are crucial in shaping the participation landscape of each project, influencing the types of storage providers that can effectively engage in their respective ecosystems. The differences in minimum hosting requirements underscore the varying technical considerations and resource allocation strategies adopted by decentralized storage platforms.

### *5.1.2.2 Capacity and Usage*

With the advent of the NFT craze in 2021, the demand for decentralized storage experienced a remarkable upswing, triggering a substantial expansion in the realm of accessible

storage options. By the conclusion of 2021, the cumulative storage capacity had surged beyond 16.7 million terabytes, showcasing an impressive growth of over fourfold compared to the preceding year.

During this period, Filecoin emerged as the clear frontrunner among various decentralized storage solutions, boasting an unparalleled network storage power that exceeded 21 million terabytes. This capability represents a remarkable accomplishment, overshadowing the storage capacity of BitTorrent's BTFS network, which holds the position of the second-largest provider of decentralized storage. In direct comparison, Filecoin's storage capacity is more than 40 times greater than that of BitTorrent's BTFS network, underscoring its dominance in the decentralized storage landscape. This remarkable contrast in storage power highlights the significant strides Filecoin has taken in bolstering its infrastructure and solidifying its position as a leader in the field of decentralized data storage.

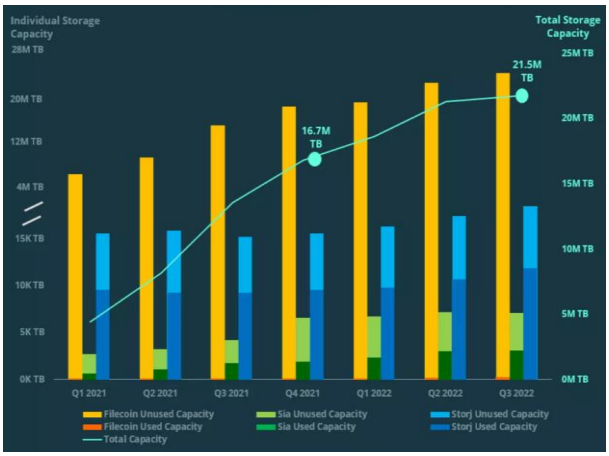


**Chart 4: Enhancement of Decentralized Storage Capacity**

Nevertheless, a substantial portion of this expanded storage capacity is currently lying dormant. As of the third quarter of 2022, a mere 1% of Filecoin's overall capacity is actively engaged. In stark contrast, the utilization rate of Storj's total capacity stands at a notably higher figure, with a utilization rate of -64%, exemplifying a scenario where demand outstrips available resources. This discrepant utilization between the two platforms highlights the evolving landscape of decentralized storage adoption and underscores the need for further exploration into the factors influencing such utilization rates.

### *5.1.2.3 Cost Efficiency*

**Table 2** provides a comparative overview of the monthly prices per terabyte (TB) for decentralized and centralized data storage providers. The data aims to assist users in evaluating the cost implications of utilizing different storage solutions based on their individual preferences and needs.

| Decentralized Providers | Monthly Price per TB | Centralized Providers | Monthly Price per TB |
|---|---|---|---|
| Filecoin | $0.0002 | iCloud | $6.00* |
| Arweave | $1.09 | Google Drive | $5.00 |
| Storj | $4.00 | OneDrive | $7.00 |
| Sia | $0.94 | Dropbox | $5.00 |
| BitTorrent | $3.01 | Amazon Drive | $7.00* |

**Table 2: Comparison of Monthly Price per TB:**

**Decentralized vs. Centralized Data Storage Providers**

Among the decentralized providers, Filecoin stands out with the lowest monthly price per TB at $0.0002. Arweave follows with a slightly higher price of $1.09, while Sia offers storage at $0.94 per TB. Storj and BitTorrent come next with monthly prices of $4.00 and $3.01 per TB, respectively.

On the centralized storage front, iCloud presents a notably higher cost at $6.00 per TB, potentially reflecting the convenience and integration features it offers within the Apple ecosystem. Google Drive and Dropbox offer storage at $5.00 per TB, and OneDrive comes in at $7.00 per TB. Amazon Drive, similar to iCloud, presents a higher potential cost of $7.00 per TB.

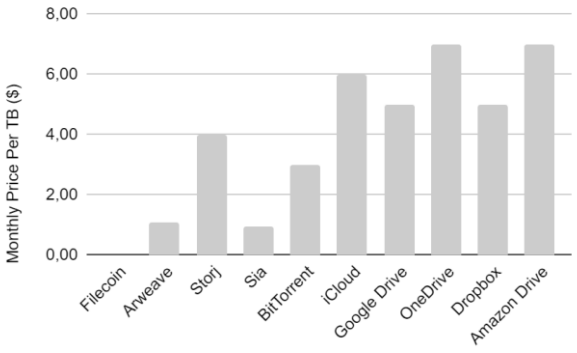


**Chart 5: Cost of Decentralized Storage vs. Centralized Storage**

Currently, Filecoin stands out as the most cost-effective storage option, boasting monthly expenses of less than a single cent. Their recent incentive initiative, Filecoin Plus, offers heightened rewards to verified storage providers engaging in legitimate transactions. Often, these transactions are subsidized by Filecoin in a bid to expand the network. Consequently, storage providers readily offer nearly negligible or zero charges in the competitive pursuit of block rewards, ultimately benefiting the network's user base. It is worth noting, however, that this approach might not accurately reflect the actual amount users are willing to spend for storing data on the Filecoin platform.

Despite the apparent cost advantages associated with decentralized storage solutions, the economic intricacies at play are far more nuanced than meets the eye. The landscape of decentralized storage encompasses diverse pricing structures that warrant a closer examination. Notably, various decentralized storage networks adopt distinct fee frameworks for data uploads (ingress) and retrievals (egress). To illustrate, consider the case of Storj, which imposes a uniform fee of $7 per terabyte (TB) for both uploading and downloading data. In contrast, Sia follows a differentiated approach, charging $0.41 per TB for uploads and $2 per TB for downloads. Contrastingly, Filecoin introduces a dynamic dimension to cost determination by relying on the fluidity of market-driven pricing, as established by its network's storage miners. This approach pegs the cost of storage to prevailing supply and

demand dynamics within the ecosystem. Consequently, the economic outlay for utilizing Filecoin is intrinsically linked to the ebb and flow of market forces. Moreover, an underpinning consideration lies in the stability of incentivized benefits extended to storage miners. Should these inducements lose their allure, there exists the potential for a paradigm shift in fee structures. In such a scenario, storage miners might opt to recalibrate their charges upwards to ensure the preservation of their bottom line. This intricate interplay of cost dynamics within decentralized storage systems underscores the multifaceted nature of financial considerations. Beyond the surface simplicity of affordability, the realm of decentralized storage harbors a labyrinth of economic variables that demand scrutiny. As stakeholders navigate these complexities, they are compelled to factor in not only immediate cost savings but also the underlying mechanisms that govern pricing, thereby embarking on a more comprehensive and informed decision-making journey.

### *5.1.2.4 Security*

The question of data security is one that has rightly gathered pace in recent years. The emergence of the cloud has reshaped the digital landscape, offered organizations greater flexibility and efficiency, and transformed how businesses store and share data on an unprecedented scale, empowering even the smallest of enterprises. But security has become something of a moving target and the risk of data breaches has risen dramatically.

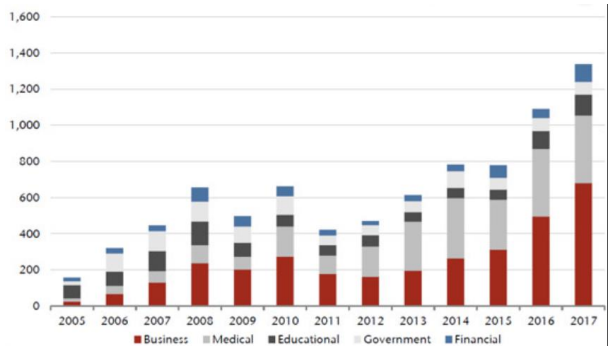


**Chart 6: Increasing number of data breaches (by entity)**

The frequency of data breaches [29] is on a continuous upward trajectory, marked by a notable escalation over the years. Notably, the count of data breaches surged from approximately 200 incidents in 2005 to a staggering figure of nearly 1,400 incidents in 2017. This undeniable trend underscores the escalating challenges associated with data security in the digital age.

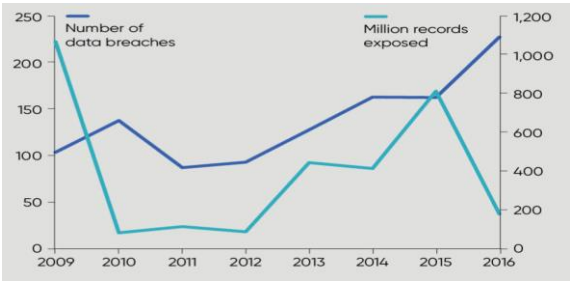


**Chart 7: Number of data breaches and exposed records in the United States annually (2005 to 2016) in millions**

A substantial portion of these breaches has particularly targeted business and government sectors, amplifying the significance of the issue. Business and government entities, entrusted with vast amounts of sensitive information, have found themselves at the forefront of data breaches. The ramifications [30] of such breaches reverberate beyond financial losses, encompassing reputational damage, compromised user privacy, and potential legal consequences.

A significant landscape of data security concerns emerges from recent statistics within the realm of cloud storage practices. Among small and medium-sized businesses (SMBs) that store customer credit card data in the cloud, a staggering 62% confess to not adhering to industry regulations. This unsettling trend highlights a substantial gap between data storage practices and the regulatory frameworks designed to safeguard sensitive financial information.

Furthermore, an alarming 54% of SMBs that store medical data in the cloud acknowledge their non-compliance [31] with industry regulations. The implications of such non-compliance in the healthcare sector are particularly grave, given the sensitive and confidential nature of medical records. The potential for unauthorized access and breaches raises substantial concerns about patient privacy and data integrity.

Another concerning aspect of cloud storage adoption is the lack of internal cloud storage policies. Astonishingly, 56% of surveyed entities do not have well-defined internal policies governing their use of cloud storage. This absence of structured guidelines increases the risk of inconsistent practices, inadequate security measures, and a lack of accountability within organizations.

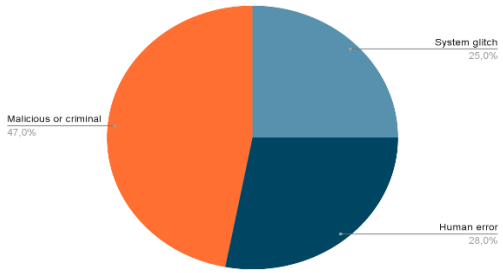


**Chart 8: Main source of data breaches**

The landscape of centralized data storage has long grappled with a myriad of security challenges, with the consequences of data breaches being particularly severe. The alarming data points underscore the extent of these challenges, shedding light on the main sources of data breaches within centralized data storage systems.

One of the primary issues plaguing centralized data storage [32] is the vulnerability to malicious or criminal attacks, which accounts for a staggering 47% of reported breaches. This susceptibility to intentional cyberattacks highlights the attractiveness of centralized repositories as high-value targets for hackers seeking to exploit sensitive information. Moreover, the presence of a single point of failure in centralized systems exposes a significant risk, as any breach could potentially compromise an entire database, leading to devastating consequences.

Another concerning factor is human error, responsible for 28% of data breaches. This emphasizes the inherent fallibility of relying on manual actions for critical data management tasks within centralized systems. From accidental data leaks to unintentional access grants, human errors can lead to inadvertent breaches, underscoring the need for enhanced safeguards and mechanisms to mitigate such risks.

System glitches account for 25% of data breaches, revealing the technical vulnerabilities [33] inherent in centralized data storage architectures. These glitches can stem from various factors, including software bugs, hardware malfunctions, or inadequate system maintenance. Such failures not only jeopardize data integrity but also disrupt business operations, causing financial losses and eroding customer trust.

Decentralized data storage presents a promising solution to these pressing problems. By design, decentralized systems distribute data across a network of nodes, eliminating the single point of failure that plagues centralized architectures. The inherent redundancy and fault tolerance of decentralized systems reduce the impact of system glitches and malicious attacks, ensuring data availability and integrity even in the face of such challenges.

Moreover, decentralized systems can substantially mitigate the impact of human errors. With advanced automation and smart contract-based governance, decentralized storage can minimize the potential for manual errors in data management processes. Additionally, the cryptographic principles underpinning decentralized architectures enhance data security and privacy, making it significantly harder for malicious actors to exploit vulnerabilities.

## 5.2 Interpretation and Discussion of Result

In this section, we embark on a journey of exploration, analysis, and synthesis, aiming to answer key research questions, unravel complexities, and unearth potential implications that may guide future actions and decisions. Through critical analysis and thoughtful interpretation, we endeavor to paint a vivid picture of the implications, significance, and broader relevance of the research findings within the realm of decentralized and centralized data storage systems.

Analyzing **Chart 4** reveals [34] that the substantial demand surge induced by the NFT boom in 2021 played a pivotal role in propelling substantial growth in the realm of decentralized storage alternatives, with Filecoin emerging as the dominant contender. Nevertheless, the significant gap between the remarkable expansion in storage capacity and the present utilization rates, particularly evident when comparing Filecoin and Storj, underscores the intricate array of challenges and possibilities that the future holds for the continuously evolving landscape of decentralized data storage. A comprehensive exploration and comprehension of these

emerging trends and underlying factors will be pivotal in shaping the trajectory of this dynamic field in the forthcoming years.

Looking ahead, we anticipate that the momentum of decentralized storage adoption will continue to gather pace, driven by escalating demands for secure and flexible data management solutions. The lessons gleaned from the observed disparities between storage capacity and utilization rates will likely fuel concerted efforts to optimize resource allocation and improve efficiency within these decentralized systems. Furthermore, as the technology matures and user awareness grows, we can envision increased collaboration between stakeholders to refine protocols, address vulnerabilities, and enhance overall network performance. The dynamic interplay between demand, capacity, and utilization will shape the evolution of decentralized storage systems, ushering in an era of more balanced and efficient data management solutions.

Proceeding further, an insightful examination of **Table 2** unmistakably reveals the stark cost differential between Decentralized Storage Providers and their Centralized counterparts. Evidently, Decentralized Storage Providers [34] offer notably lower monthly prices per terabyte (TB) compared to their Centralized counterparts. This initial observation underscores the considerable cost-effectiveness that decentralized storage solutions can potentially bring to the forefront.

However, the seemingly straightforward cost advantage of decentralized storage solutions belies a tapestry of intricate economic dynamics. The evolving landscape of decentralized storage, intricately intertwined with the swift-paced blockchain industry, is replete with nuanced factors that transcend surface-level impressions. While the pricing disparities are apparent, an in-depth exploration reveals that the matter at hand is significantly more complex and multifaceted.

In this context, it is imperative to recognize the dynamic nature of the blockchain domain. The blockchain industry [35] is characterized by rapid advancements, evolving technologies, and shifts in market dynamics. As these elements interplay, it is not unreasonable to envision a future where the cost dynamics of decentralized storage undergo transformation.

Considering the historical trajectory of blockchain technologies and their propensity to catalyze efficiencies and cost optimizations, a compelling hypothesis emerges. It is conceivable that the current cost differentials witnessed between decentralized and centralized storage solutions may evolve over time. The ongoing innovation in the blockchain sector, coupled with heightened competition among decentralized storage providers, may potentially drive down costs even further.

This trajectory aligns with the broader trend of technological advancements leading to increased accessibility and affordability. Past instances in the blockchain realm have demonstrated how rapid innovation and iterative development can lead to substantial reductions in operational costs. These reductions, in turn, could translate into even more compelling cost advantages for decentralized storage solutions.

In light of these dynamics, it is not unfounded to speculate that decentralized storage costs may trend toward greater affordability in the coming years. As the blockchain industry continues its forward march, propelled by technological breakthroughs and market forces, the cost-effectiveness of decentralized storage could potentially become even more pronounced.

Conclusively, the issue of security emerges as a paramount concern within the realm of centralized data storage. A meticulous examination of **Chart 6** and **Chart 7** casts a spotlight on the escalating trend in data breaches associated with cloud data storage. The discernible rise in such breaches underscores the urgency of addressing inherent vulnerabilities within centralized data storage systems. Moreover, the analysis conveyed by **Chart 8** underscores a pivotal insight - a mere 28% of data breaches are attributed to human error, thereby accentuating the preponderant role of systemic and structural inadequacies.

In this context, the transition towards decentralized data storage solutions stands as a potential avenue for mitigating a substantial portion of these security concerns. By extrapolating the insight that a significant majority of data breaches stem from non-human error factors, a compelling inference emerges: the adoption of decentralized solutions could potentially address up to 70% of the prevailing data breach challenges.

It is important to note that the transition to decentralized data storage systems does not present a panacea for all security-related predicaments. However, the juxtaposition of the security implications associated with centralized systems, as demonstrated through **Chart 6** and **Chart 7**, with the inherent attributes of decentralized architectures, evokes a proposition of substantive merit.

The distributed nature of decentralized storage, characterized by its dispersal of data across a network of nodes, inherently mitigates the vulnerabilities associated with single points of failure. Consequently, malicious attacks, system glitches, or outages that feature prominently within centralized storage systems, are rendered considerably less potent within the decentralized paradigm.

Furthermore, the cryptographic underpinnings of blockchain technology, upon which many decentralized data storage solutions are founded, fortify data integrity and confidentiality. The immutable and tamper-resistant nature of blockchain transactions lends an intrinsic safeguard against unauthorized modifications, thereby augmenting security.

It is, however, essential to acknowledge that while decentralized storage solutions offer a promising trajectory toward enhanced security, the multifaceted nature of cybersecurity necessitates a comprehensive and multifarious approach. Collaborative endeavors involving technological innovation, rigorous policy frameworks, and proactive user engagement remain pivotal in establishing a holistic and robust security apparatus.

In conclusion, the comprehensive exploration into the landscape of decentralized and centralized data storage systems has revealed a multifaceted tapestry of challenges, opportunities, and implications. The surge in demand fueled by the NFT boom of 2021 has catalyzed remarkable growth in decentralized storage alternatives, led by Filecoin. Yet, the contrasting gap between expanded capacity and utilization rates underscores the evolving complexities within this realm.

Anticipating the future, we foresee continued momentum in the adoption of decentralized storage solutions, driven by demands for secure and flexible data management. Collaborative efforts are likely to refine protocols and optimize resource allocation, shaping a more balanced and efficient data management landscape.

The cost dynamics revealed in the analysis emphasize the cost-effectiveness of decentralized solutions, yet unveil intricate economic dynamics. While cost disparities are evident, the rapidly evolving blockchain industry could potentially reshape these dynamics, further enhancing affordability.

Security emerges as a paramount concern within centralized storage, with the findings highlighting the potential of decentralized solutions to mitigate a substantial portion of data breach challenges. While not a panacea, the intrinsic attributes of decentralized architectures and blockchain's cryptographic foundation bolster data integrity and security.

In this dynamic intersection of technology and security, these synthesis underscores the transformative potential of decentralized storage. As stakeholders navigate this evolving landscape, they embark on a journey that transcends immediate benefits, delving into the nuanced complexities that will ultimately shape the future of data storage.

# 6. CONCLUSION AND FUTURE WORKS

## 6.1 Conclusion

This research investigation delved deeply into the realm of centralized and decentralized data storage systems, meticulously evaluating their respective strengths, vulnerabilities, and ramifications. The culmination of this inquiry has illuminated a tapestry of discerning observations

**Operational Excellence and Integration**

Centralized data storage, as exemplified by Amazon AWS, has demonstrated a commendable track record in operational proficiency and seamless integration with an array of applications, encompassing analytics, ERP, and CRM functionalities.

In contrast, the evolution of decentralized storage is ongoing, marked by areas necessitating refinement, including user experience, integration capabilities, and overarching performance assurance.

**Performance and Cost Advantages**

Centralized data storage guarantees standardized and foreseeable performance, alongside a head start in cultivating self-service capacities. In the decentralized domain, the potential for economically viable solutions stands out.

Noteworthy is the capacity of decentralized storage providers, typified by Filecoin, to present substantially reduced costs compared to their centralized counterparts. This distinct pricing advantage positions decentralized solutions as formidable contenders.

**Security and Reliability**

Centralized data storage is susceptible to the vulnerabilities associated with single points of failure. In contrast, the decentralized paradigm inherently mitigates this vulnerability, amplifying security and bolstering data reliability.

Emerging technologies, particularly blockchain-based systems within decentralized storage, offer a fortified bulwark of security and data integrity.

**Vendor Lock-in and Data Freedom**

Centralized storage engenders potential unpredictability in terms of costs, stemming from bandwidth and API charges. In parallel, decentralized storage is actively advancing its infrastructure to nurture self-service capabilities.

The landscape of decentralized storage is undergoing continuous refinement in its drive toward comprehensive self-service functionalities.

**Privacy and Censorship Risks**

Centralized storage exposes users to potential privacy breaches and censorship vulnerabilities, as governmental authorities can exert influence to access data. The decentralized paradigm, characterized by data distribution across nodes, counters these risks by providing inherent resistance to censorship and augmenting control over data accessibility.

**User Interface and Compute Capability**

Centralized storage interfaces confront opportunities for enhancement, particularly concerning user experience and integration nuances.

In the decentralized context, the ability to efficiently query and compute stored data remains constrained, thereby dictating the

applicability of this paradigm to specific use cases.

**Unpredictable Costs and Self-Service Capabilities**

Centralized storage engenders potential unpredictability in terms of costs, stemming from bandwidth and API charges. In parallel, decentralized storage is actively advancing its infrastructure to nurture self-service capabilities.

The landscape of decentralized storage is undergoing continuous refinement in its drive toward comprehensive self-service functionalities.

The culmination of this discerning analysis underscores the intricacies that envelop the selection between centralized and decentralized data storage systems. While the realm of centralized storage excels in certain dimensions such as operational finesse and integration, decentralized storage unveils unique virtues in the realms of security, data autonomy, and potentially frugal expenses. These findings illuminate the dynamic nature of both paradigms and foreshadow the potential trajectories that the field of data storage might traverse in the future. As technology evolves, stakeholders must thoughtfully deliberate their specific requirements and preferences to render informed verdicts pertaining to their chosen data storage solutions.

## 6.2 Future Works

This study has provided valuable insights into the realm of decentralized data storage systems, highlighting their strengths, weaknesses, and potential implications. However, the exploration of this dynamic field is far from exhaustive. Future research endeavors could delve deeper into assessing the evolving cost dynamics of decentralized storage, analyzing the impact of emerging technologies on system performance, and exploring innovative mechanisms for enhancing integration, UI/UX, and compute capabilities. Furthermore, investigations into the scalability of decentralized storage systems, especially in the face of increasing data demands, and the development of standardized frameworks for assessing reliability and fault tolerance could contribute to a more comprehensive understanding of this evolving landscape. By addressing these avenues, researchers and readers can continue to enrich their knowledge and shape the trajectory of decentralized data storage systems in the years ahead.

# 7. REFERENCES

[1] Wang, C. 2022. An Honest Report on Web3 Data & Storage. Retrieved June 16, 2023, from https://curiouscat178.substack.com/p/the-full-34-page-version-here-an

[2] Fowler, G. 2021. Taking a Decentralized Approach To Cyber Security, Data Protection and Privacy. Retrieved June 16, 2023, from https://www.forbes.com/sites/forbesbusinessdevelopmentcouncil/2021/04/07/taking-a-decentralized-approach-to-cyber-security-data-protection-and-privacy/?sh=5ea095dc5601

[3] Sheldon, R. 2022. 7 Decentralized Data Storage Networks Compared. Retrieved June 16, 2023, from https://www.techtarget.com/searchstorage/tip/Comparing-4-decentralized-data-storage-offerings

[4] William, S., Diordiiev, V., Berman, L., Raybould, L. and Uemlianin, I. 2020. Arweave: A Protocol for Economically Sustainable Information Permanence. Arweave project yellow paper. Retrieved June 16, 2023, from https://www.arweave.org/yellow-paper.pdf

[5] Protocol Lab. 2017. Filecoin: A Decentralized Storage Network. Filecoin project paper. Retrieved June 16, 2023, from https://filecoin.io/filecoin.pdf

[6] Rosic, A. 2022. Centralized vs Decentralized Storage: Redefining Storage Solutions with Blockchain Tech. Retrieved June 16, 2023, from https://blockgeeks.com/guides/centralized-vs-decentralized-storage-redefining-storage-solutions-with-blockchain-tech/

[7] Nakamoto, S. 2008. Bitcoin: A peer-to-peer electronic cash system. Bitcoin project paper. Retrieved June 16, 2023, from https://bitcoin.org/en/bitcoin-paper

[8] Pataiya, D. 2023. What is decentralized storage, and how does it work? Retrieved June 16, 2023, from https://cointelegraph.com/news/what-is-decentralized-storage-and-how-does-it-work

[9] Binance Academy. 2023. Blockchain. Retrieved June 16, 2023, from https://academy.binance.com/en/blockchain

[10] Liu, A. D., Du, X. H., Wang, N., and Li, S. Z. 2018. Research Progress of Blockchain Technology and its Application in Information Security. Ruan Jian Xue Bao/Journal of Software,2018,6,14:1-24.

[11] Zhu, Yan., Lv, Chunli., Zeng, Z., Wang, J., and Pei, B. 2019. Blockchain-based Decentralized Storage Scheme. Retrieved June 16, 2023, from https://www.researchgate.net/publication/334418265_Blockchain-based_Decentralized_Storage_Scheme

[12] Merkle, R. C. 1980. Protocols for public key cryptosystems [C]//Security and Privacy. 1980 IEEE Symposium on. IEEE, 1980: 122-122.

[13] Fan, J., Yi, L. T., and Shy, J. W. 2013. Research on the technologies of Byzantine system.

[14] Wackerow. 2023. Introduction to Smart Contract. Ethereum Project Smart Contract Document. Retrieved June 16, 2023, from https://ethereum.org/en/developers/docs/smart-contracts/

[15] G, Wood. 2014. Ethereum: A secure decentralized generalized transaction ledger. Ethereum project yellow paper, vol. 151, pp. 1-32, 2014.

[16] Ethereum Team. 2023. Ethereum blockchain app platform. Retrieved June 16, 2023, from https://ethereum.org/en/

[17] Wang, S., Zhang, Y., and Zhang, Y. 2018. A Blockchain-Based Framework for Data Sharing with Fine-grained Access Control in Decentralized Storage Systems. Retrieved June 16, 2023, from https://www.researchgate.net/publication/326075893_A_BlockchainBased_Framework_for_Data_Sharing_With_FineGrained_Access_Control_in_Decentralized_Storage_Systems

[18] IBM. 2023. Blockchain for financial services. Retrieved June 16, 2023, from https://www.ibm.com/blockchain/industries/financial-services

[19] Proof of Existence. 2023. Proof of Existence. Retrieved June 16, 2023, from https://proofofexistence.com/

[20] Iotex. 2023. A decentralized network for internet of things. Retrieved June 16, 2023, from https://iotex.io/

[21] Fowler, G. 2021. Taking a Decentralized Approach To Cyber Security, Data Protection and Privacy. Retrieved June 16, 2023, from https://www.forbes.com/sites/forbesbusinessdevelopmentcouncil/2021/04/07/taking-a-decentralized-approach-to-cyber-security-data-protection-and privacy/?sh=5ea095dc5601

[22] IPFS. IPFS Documentation. IPFS project paper. Retrieved June 16, 2023, from https://docs.ipfs.tech/

[23] Vorick, D., and Champine, L. 2014. Sia: Simple Decentralized Storage. Sia project paper. Retrieved June 16, 2023, from https://sia.tech/sia.pdf

[24] Sivarajah, U., Irani, Z., and Weerakkody, V. 2016. Evaluating the use and impact of Web 2.0 technologies in local government. Retrieved June 16, 2023, from https://www.sciencedirect.com/science/article/pii/S0740624X15000763

[25] Dannen, C. 2017. Introducing Ethereum and Solidity. Retrieved June 16, 2023, from https://www.researchgate.net/publication/315378297_Introducing_Ethereum_and_Solidity

[26] Laposky, I. 2010. Facebook Exposed 87 Million Users to Cambridge Analytica. Retrieved June 16, 2023, from https://www.wired.com/story/facebook-exposed-87-million-users-to-cambridge-analytica/

[27] Wang, C. 2022. An Honest Report on Web3 Data & Storage. Retrieved June 16, 2023, from https://curiouscat178.substack.com/p/the-full-34-page-version-here-an

[28] Win, K. W. 2022. The State of Decentralized Storage. Coingecko Research Paper. Retrieved June 16, 2023, from https://www.coingecko.com/research/publications/the-state-of-decentralized-storage

[29] Jha, A. (2019). 5 Importance of Cloud Computing Security. Retrieved June 16, 2023, from https://www.tricksroad.com/2019/02/importance-of-cloud-computing-security.html

[30] Meng, L., and Sun, B., 2022. Research on Decentralized Storage Based on a Blockchain. Retrieved June 16, 2023, from https://www.mdpi.com/2071-1050/14/20/13060

[31] Mohaisen, D., and Kin, J. 2013. The Sybil Attacks and Defenses: A Survey. Retrieved June 16, 2023, from https://www.researchgate.net/publication/259440924_The_Sybil_Attacks_and_Defenses_A_Survey

[32] Zhou, S., Li, K., Xiao, L., and Cai, J. 2023. A Systematic Review of Consensus Mechanisms in Blockchain. Retrieved June 16, 2023, from https://www.researchgate.net/publication/370695931_A_Systematic_Review_of_Consensus_Mechanisms_in_Blockchain

[33] Banger, R., Mittal, R., Knowal, R., and Mehta, A. 2019. A Study On BlockChain And Cryptography. Retrieved June 16, 2023, from https://www.researchgate.net/publication/342151198_A_Study_On_BlockChain_And_Cryptography

[34] Imperval. What is Fault Tolerance? Imperval Network Security Learning Document. Retrieved June 16, 2023, from https://www.imperva.com/learn/availability/fault-tolerance/.

[35] Nguyen, T. 2018. Discover the secret of encryption. Retrieved June 16, 2023, from https://www.raconteur.net/sponsored/discover-secret-encryption