Jurnal Teknik Informatika (JUTIF)
Vol. 4, No. 3, June 2023, pp. 639-645

DOI: https://doi.org/10.52436/1.jutif.2023.4.3.1030
p-ISSN: 2723-3863
e-ISSN: 2723-3871

DATA AVAILABILITY IN DECENTRALIZED DATA STORAGE USING FOUR-NODE INTERPLANETARY FILE SYSTEM

Tony Haryanto[*1], Kalamullah Ramli[2], Arga Dhahana Pramudianto[3]

[1,2]Departement of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Indonesia
[3]Department of Computer and Network Engineering, The University of Electro-Communications (UEC), Japan
Email: [1]tony.haryanto@ui.ac.id, [2]kalamullah.ramli@ui.ac.id, [3]arga.pramudianto@uec.ac.jp

(Article received: May 04, 2023; Revision: May 27, 2023; published: June 26, 2023)

Abstract

Centralized storage is a data storage model in which data is stored and managed in a single physical location or centralized system. In this model, all data and information are stored on servers or data centers managed by one entity or organization. This model also has disadvantages such as risk of system failure against distributed denial of service (DDoS) attacks, natural disasters, and hardware failures causing a single point of failure. This threat results in loss of data and a lack of user confidence in the availability of data in centralized storage. This study proposes to evaluate the availability of data in decentralized data storage using a four-node interplanetary file system (IPFS) that is interconnected with a swarm key as the authentication key. Unlike centralized storage which has only one data center, four-node IPFS allows users to upload and download data from four interconnected data centers. This can avoid dependence on the central server and reduce server load. The evaluation results show that decentralized data storage using a four-node IPFS system is three times more resilient than centralized storage against a single point of failure. This system can increase data availability so that organizations can minimize data loss from the threat of system failure.

Keywords: *Availability, Decentralized, IPFS, Storage*.

1. INTRODUCTION

Centralized storage has been the traditional and most commonly used data storage model in various organizations [1]. This model involves the storage of all data and information on a single server or data center that is managed by a single entity or organization[2]. The use of centralized storage has several advantages, such as increased efficiency and effective data management, as it enables the quick and easy storage and retrieval of data [3] and the ability to consolidate all organizational data into one centralized place[4]. However, this model has weaknesses, including vulnerability to security risks such as DDoS attacks, which can result in system failure and create a single point of failure in case of a natural disaster [5]. Given the limitations of the centralized storage model, many studies have been conducted to explore alternative data storage solutions. One of the most promising solutions is the use of a decentralized data storage system such as IPFS [6].

IPFS, which stands for InterPlanetary File System, is a peer-to-peer network protocol that allows users to store and retrieve files from a distributed network without relying on a central server, thus reducing the server load [7]. IPFS uses a decentralized distribution system that allows files to be stored on multiple nodes, providing secure and quick access to files [8]. Each file is identified by its unique hash and can be retrieved from any node on the network. In IPFS, each node is equally important and can store and retrieve files [9]. Every node that wants to connect to an IPFS network must use the same swarm key. By using the same swarm key, nodes can authenticate each other and encrypt their communications[10]. The advantages of using IPFS for data storage include enhanced security, as files are not stored in a central location and cannot be deleted by any party [11]. Additionally, IPFS offers faster file transfer speeds, since files can be retrieved from the node closest to the user. The system also supports different versions of a file, enabling users to access both old and new versions of the same file [12].

IPFS, although still in its infancy, has gained a lot of attention and traction among various organizations due to its potential and benefits in online file storage and distribution [13]. The architecture of IPFS is unique in the way it breaks down data into several encrypted data blocks and assigns each block a unique address based on its hash [14]. This decentralized approach allows for the efficient distribution of data across the network, where each node can contribute storage and bandwidth resources [15]. As a result, data is available for download and access from various points on the network, reducing the load on servers and improving the availability and privacy of data [16]. With this innovative approach to data storage, IPFS offers a more secure and reliable alternative to

traditional centralized storage models, and its increasing popularity is a testament to its potential for widespread adoption.

The proposed study aims to overcome the problem of data availability in organizations by proposing a decentralized data storage system. The system is based on four interconnected IPFS nodes and acts as a distributed file storage system, which provides higher data availability compared to centralized storage solutions. The system offers increased file availability, which can help organizations reduce the risk of data loss from a variety of external and internal threats, including hardware failures, cyber-attacks, and natural disasters.

This proposed system enables organizations to manage their data effectively safely and efficiently, ensures that they can access their critical data whenever they need it, overcomes data availability issues, avoids dependency on a central server, and reduces server load.

2. RESEARCH METHODS

The research methods used are problem identification, defining objectives, system design, testing, evaluation, and conclusions [17]. The research stages are illustrated in Figure 1 with the following details:

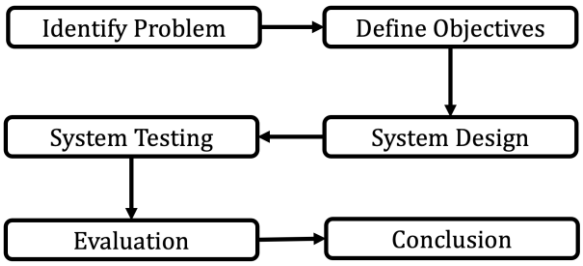

Figure 1. Research Stages

2.1. Identify Problem

In the problem identification stage of this research, it was revealed that centralized data storage is prone to data availability problems and security threats from outside and inside. Failure of the central server or data center can disrupt data access and have a serious impact on the organization's operations. Security threats such as hacking or misuse of data can damage reputation and cause financial losses.

2.2. Define Objectives

This study aims to find a solution by developing a design using a four-node interplanetary file system (IPFS), which uses a decentralized distribution system to overcome the limitations of centralized data storage, reduce the risk of concentrated data availability, and increase security by avoiding a single point of failure.

2.3. System Design

In this phase of the research, a design for a decentralized storage system will be developed using the four-node InterPlanetary File System (IPFS). IPFS utilizes content-based file addresses and distributed hash tables to ensure the availability and accessibility of data. The proposed architecture addresses the limitations of centralized storage systems and provides a dependable solution for secure and efficient data storage and management.

2.4. System Testing

System availability testing involves creating a virtual environment using VMware virtualization software. Four Linux operating systems represent nodes with different specifications. The test scenario includes uploading and downloading files from each node in the IPFS network. Experiments are conducted with disabled nodes to assess file availability. Valuable insights are gained regarding potential issues in an IPFS network resulting in node disablement.

2.5. Evaluation

Furthermore, an evaluation of the availability of data is carried out as a basis for drawing conclusions and recommendations for further research. This evaluation includes data collection and analysis of system test results that implement IPFS as an alternative to decentralized data storage. The evaluation aims to assess the extent to which the system with IPFS can overcome data availability problems encountered in centralized data storage. The results of this evaluation will provide an in-depth understanding of the effectiveness of IPFS in increasing data availability and can provide recommendations for further research development in the field of decentralized data storage.

2.6. Conclusion

The final stage of this research involves drawing conclusions from the results of the research and compiling recommendations and suggestions for further research development. In this stage, the results of data and system evaluation are used to assess the effectiveness of IPFS implementation in addressing data availability and security threats to centralized data storage. The conclusions drawn will provide a clear understanding of the potential of IPFS as an effective alternative, while recommendations will help improve the system, address identified deficiencies, and encourage further exploration in the field of decentralized data storage.

3. RESULTS AND DISCUSSIONS

After identifying the problem and defining objectives as in the stages of the research method, the next research stage will be presented, namely system

design and testing. each stage will be explained in detail along with related tables and figures.

3.1. System Design

The systems design research approach is widely recognized for its ability to provide a comprehensive and systematic analysis of complex systems such as data storage systems. By utilizing this approach, researchers can identify and evaluate the key elements of a system, its structure, and its behavior, and provide recommendations for improvement. In this study, a system design approach is used to develop and evaluate a decentralized storage system that uses a four-node interplanetary file system (IPFS) as its backbone.

The proposed design takes into account factors such as data availability, scalability, and security, and presents a solution that addresses the limitations of centralized storage systems. The four-node IPFS system is designed to provide a distributed file storage solution that is reliable, efficient, and secure, with the ability to withstand external and internal threats to data security.

The architectural design of the proposed system is described in detail in this section, including the components and their interactions, as well as the various layers and protocols used to ensure seamless communication and data sharing. Overall, the system design approach provides a sound framework for the development and evaluation of complex systems such as decentralized data storage systems, and the proposed design using a four-node IPFS system offers a promising solution for organizations seeking a more secure and efficient approach to data storage and management.

The design is carried out based on the recommendations obtained in the literature study so that this research can be a solution to the problems that have been defined. The purpose of this system design is to produce decentralized storage so that data can be securely and available when users want to access the data. To support this goal, a system design was carried out using the Interplanetary File System (IPFS). IPFS is a peer-to-peer network protocol for storing data in a distributed system. Unlike file storage on servers, IPFS distributes those files across various nodes in IPFS [18]. While other file systems use location-based addresses to locate files, IPFS uses content-based addresses where file addresses are based on a hash of the file content. IPFS also uses distributed hash tables to know where to find the data or the path that leads closer to the data [19].

In this system architecture, IPFS is used as a decentralized file storage that allows files to be spread across multiple nodes in the IPFS network. This system design will utilize four IPFS nodes as shown in Figure 2.

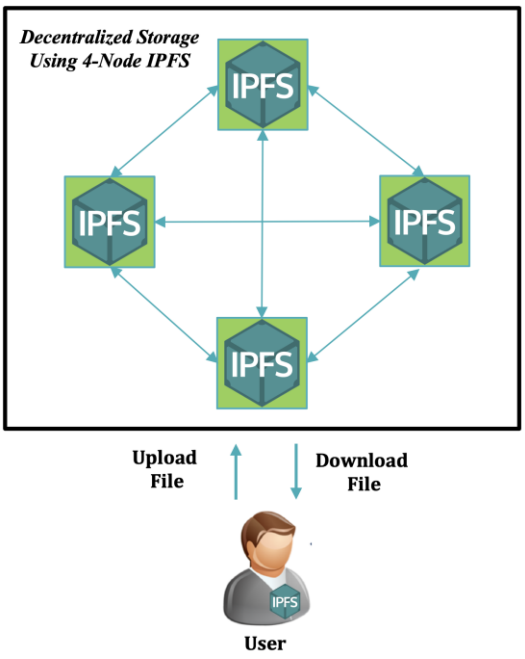

Figure 2. Decentralized Data Storage System Design

The Figure 2 illustrates the architecture of a decentralized storage system using four IPFS nodes. To interact with this system, each user must be connected to the IPFS network. Each user has a special identity, namely peer identity (PeerID). PeerID is the unique identity of a node in a peer-to-peer network. PeerID functions as an identifier between fellow users in a network. In this system architecture, PeerID is used to allow users to interact with other users in an IPFS network. The system has commands for downloading files and uploading the hash value of files to the IPFS network. In describing this design system, an upload and download process algorithm is needed as listed in Table 1 and Table 2.

Table 1. Algorithm for Uploading a File

Algorithm 1
1: function onSubmit
2: upload file
3: convert content of file into buffer
4: call method *upload()*
5: send buffer of file into ipfs
6: return hash
7: end function

Table 1 provides a clear overview of how the system operates in terms of uploading files to the IPFS network. The process is initiated by calling the *onSubmit* function, which triggers the upload process. In the initial step, files are converted into a buffer, which is an array of data that can be processed and stored in the IPFS network. This step is crucial because it ensures that the file is in a format that is compatible with the IPFS network. Once the file is successfully converted into a buffer, it is then uploaded to the IPFS network. Upon completion of the upload process, IPFS returns a hash value that uniquely identifies the uploaded file. This hash value can then be used to retrieve the file from the IPFS

network in the future. The detailed explanation of the upload process provides a clear understanding of how the system functions and the specific steps that need to be followed for the successful storage and retrieval of files in the IPFS network.

Table 2. Algorithm for Downloading a File

Algorithm 2
1: function onSubmit
2: upload hash
3: call method *download()*
4: send hash of file into ipfs
5: return file
6: end function

In the IPFS network, the process of downloading files is initiated by the system calling the *onSubmit* function, as indicated in Table 2 above. To download a file, users need to send the file hash value to the IPFS network, which then retrieves the file to be shared with the intended consumers. The unique hash value assigned to each file ensures that the correct file is retrieved and shared, thereby enhancing the security and reliability of the decentralized storage system. Moreover, in addition to the files, IPFS also generates a new hash value from the original hash value of the file sent to the IPFS. This feature provides an added layer of security to the system, as it ensures that the file has not been tampered with during the download process.

3.2. System Testing

System availability testing is done by creating a virtual environment that is similar to a real environment, where the system or application can be tested in isolation or without affecting the actual real environment. System simulation is run using VMware 12.2.4 virtualization software, where VMware is built and runs on top of the main operating system. In the system simulation, VMware is made to run four Linux operating systems representing four different nodes with operating system specifications as shown in Table 3, thus enabling more secure and isolated application testing and development. In addition, system simulation can also be used to test information security without disturbing the main operating system or the real environment.

Table 3. Node Specifications

Node	Processor	OS	Memory	Storage
Node1	2 Core	Ubuntu 20.04.5 LTS	4 GB	66 GB
Node2	2 Core	Ubuntu 20.04.5 LTS	4 GB	78 GB
Node3	2 Core	Ubuntu 18.04.06 LTS	4 GB	43 GB
Node4	2 Core	Ubuntu 18.04.06 LTS	4 GB	13 GB

For the test scenario, four nodes on the IPFS network connected with the same swarm key will be used. The four nodes will run in a pre-prepared VMware simulation environment. Even though each IPFS node has a different node name and IP address, all these nodes will still be on the same network. This allows for easy communication between nodes on the IPFS network. On the IPFS network, all nodes will identify each other using their respective PeerID. PeerID is a unique identifier used for each communicating node on an IPFS. In testing file availability, the first attempt will be made by uploading and downloading files from each node on IPFS, this is done to test the functionality of the nodes on the network. The second attempt will be carried out by downloading the file that was uploaded in the first attempt, but with one of the nodes in the network disabled. The third experiment was carried out by downloading the file that was uploaded in the first experiment but with two nodes in the network being disabled. The last experiment was carried out by downloading the file that was uploaded on the first try but with three nodes in the network being disabled. Experiments two, three, and four were conducted to determine the availability of files in a network with four nodes when one, two and three nodes were inactive. Figure 3 illustrates the test scenarios that will be carried out with one, two, and three of the four inactive nodes. Through this scenario, it is hoped that accurate and useful information can be obtained to illustrate various potential problems that may occur in an IPFS network resulting in network nodes being disabled.

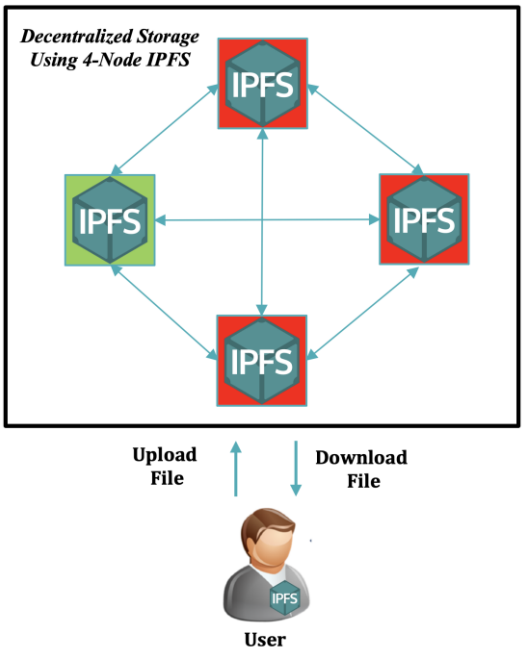

Figure 3. Data Availability Testing Scenario

Before conducting data availability testing, system functionality testing is carried out to ensure that all features and functions in the IPFS implementation run well and as expected. This functionality testing includes upload and download testing of the decentralized data storage system using IPFS. Upload testing aims to verify the system's

ability to receive files from users and store them effectively in a decentralized network. Meanwhile, download testing is conducted to ensure that users can retrieve files successfully from various nodes distributed in the IPFS network. By conducting these functionality testing, it can be ensured that the system can operate properly in terms of file upload and download before further evaluation is carried out regarding data availability which is the main focus of this research. The results of the functionality testing illustrated in Table 4 shows that the upload and download functions of the proposed data storage system can function properly, which indicates that the tested system has good performance and can be used to meet the needs of users in various situations.

Table 4. System Functionality Testing Results

Node	Status	Testing	Result
Node1	Active	Upload	Success
		Download	
Node2	Active	Upload	Success
		Download	
Node3	Active	Upload	Success
		Download	
Node4	Active	Upload	Success
		Download	

The availability testing of data was conducted in a VMware simulation environment using four nodes on IPFS sharing the same swarm key. In this experiment, each IPFS node had a different IP address but remained within the same network, allowing them to connect and communicate effectively with one another. A unique identifier for each node, known as a PeerID, was used collectively by all the nodes to communicate within the IPFS. To examine data availability, experiments were designed to test the system's capability in handling unexpected situations, specifically when one, two, or three out of the four nodes were inactive or disabled. In the data availability testing with one disabled node, the author attempted to upload data from Node1 to IPFS. Subsequently, the author tried to download the data from all the nodes, including Node1. However, in this particular experiment, Node4 was intentionally disabled while the other nodes remained active. The results of the data availability testing with one of the four nodes disabled are presented in Table 5, providing insights into the system's ability to maintain data availability in less-than-ideal scenarios.

Table 5. Result of One Node Non-active

Node	Status	Testing	Result
Node1	Active	Download	Success
Node2	Active	Download	Success
Node3	Active	Download	Success
Node4	Non-active	Download	Failed

In the subsequent stage of the research, an investigation was conducted to examine the accessibility and availability of files in a scenario where two of the four nodes were deactivated in the IPFS network used by the system. The main objective of this investigation was to assess the system's ability to maintain file accessibility and availability despite the absence of two nodes. The investigation began with the researcher attempting to upload files from Node1 to IPFS. Subsequently, an attempt was made to download the files from all nodes, including Node1. To simulate a system failure or potential cyber attack, the experiment was specifically designed to deactivate two nodes, namely Node3 and Node4, while keeping the other nodes operational. This setup aimed to evaluate the system's resilience in unfavorable situations. The results of the investigation on the file availability with two of the four disabled nodes are presented in tablTableThese results provide valuable insights into the system's capability to ensure access to desired files even when facing unexpected circumstances, such as the non-operational status of two nodes. The findings highlight the reliability and effectiveness of the system in maintaining data availability, underscoring its robustness in challenging scenarios.

Table 6. Result of Two Node Non-active

Node	Status	Testing	Result
Node1	Active	Download	Success
Node2	Active	Download	Success
Node3	Non-active	Download	Failed
Node4	Non-active	Download	Failed

After thoroughly testing the availability of files with two of the four disabled nodes and analyzing the results, the subsequent step in the experiment focuses on assessing the system's capability when three of the four nodes are disabled. This critical test aims to evaluate the system's resilience and determine how it performs in the face of multiple node failures, emphasizing its ability to maintain access to the desired data. In this particular experiment, the author initiates the process by attempting to upload files from Node1 to the IPFS network. Subsequently, the author proceeds to download the files from all nodes, including Node1, despite three of the four nodes being inactive, specifically Node2, Node3, and Node4. By deliberately simulating these conditions, the experiment effectively replicates a more severe scenario resembling a potential cyber attack or system failure. The results obtained from testing the availability of files with three of the four disabled nodes are meticulously recorded and presented in Table 7.

Table 7. Result of Three Node Non-active

Node	Status	Testing	Result
Node1	Active	Download	Success
Node2	Non-active	Download	Failed
Node3	Non-active	Download	Failed
Node4	Non-active	Download	Failed

These results provide crucial insights into the system's remarkable ability to facilitate access to the desired data, even when three out of the four nodes within the network are unavailable. The findings

underscore the system's robustness and underscore its effectiveness in ensuring data availability, highlighting its resilience in demanding circumstances.

4. DISCUSSION

The next stage of the research is evaluation, where a series of comprehensive experiments are conducted to test the availability of files in various conditions on the proposed decentralized data storage system. The testing process involves intentionally disabling one, two, and three out of the four IPFS nodes in the network used by the system. The objective of these experiments is to simulate potential system failures or cyber-attacks that may occur in real-world scenarios, thereby evaluating the resilience and effectiveness of the system in adverse situations.

The experimental results reveal that even with the disabling of one, two, or three nodes, the system successfully delivers the desired files through the active nodes. This demonstrates the system's ability to adapt and function optimally, effectively mitigating the impact of unforeseen failures or security breaches. The experimental findings, summarized in Table 5, Table 6, and Table 7, confirm consistent data availability, even in the presence of individual or multiple node failures.

These findings establish that the proposed decentralized data storage system is three times more reliable and robust in handling unexpected circumstances compared to centralized storage systems. Previous research has also highlighted the vulnerability of centralized storage to single points of failure originating from within or outside the organization. However, with the implementation of this decentralized system, organizations can obtain a secure and resilient data storage solution, significantly reducing the risks associated with data loss and system downtime. By leveraging the system's capability to maintain data availability even in the face of node failures, organizations can confidently adopt this decentralized approach, ensuring uninterrupted access to their data and enhancing overall operational continuity. This approach also reduces reliance on centralized servers and alleviates the burden on central server infrastructure.

5. CONCLUSION

The proposed decentralized data storage system is three times more reliable and powerful than centralized storage, reducing the risk of data loss and system downtime. In contrast to centralized storage, which stores data in a single location and is vulnerable to DDoS attacks, natural disasters, and hardware failures, the decentralized system utilizes a four-node IPFS. This enables users to upload and download data from an interconnected hub, thereby decreasing reliance on a central server. The evaluation confirms the robustness of the decentralized system, increasing data availability and minimizing the risk of data loss. By adopting this approach, organizations ensure uninterrupted access and reduce dependency on centralized servers.

ACKNOWLEDGMENT

This research received financial support from the Indonesian Ministry of Communication and Informatics.

REFERENCES

[1] S. Khan, X. Liu, S. A. Ali, and M. Alam, “Storage Solutions for Big Data Systems: A Qualitative Study and Comparison,” pp. 1–35, 2019, [Online]. Available: http://arxiv.org/abs/1904.11498

[2] K. Bhalibar, A. Singh, H. Sharma, A. Uphadyay, and H. Gupta, “Centralize Storage System with Encryption vs Decentralize Storage System Using Blockchain,” *SSRN Electronic Journal*, pp. 1–7, 2022, doi: 10.2139/ssrn.4119952.

[3] O. Lo, W. J. Buchanan, S. Sayeed, P. Papadopoulos, N. Pitropakis, and C. Chrysoulas, “GLASS: A Citizen-Centric Distributed Data-Sharing Model within an e-Governance Architecture,” *Sensors*, vol. 22, no. 6, 2022, doi: 10.3390/s22062291.

[4] S. K. Seo, D. Y. Yun, and C. J. Lee, “Design and optimization of a hydrogen supply chain using a centralized storage model,” *Appl Energy*, vol. 262, no. December 2019, p. 114452, 2020, doi: 10.1016/j.apenergy.2019.114452.

[5] M. Hajizadeh, N. Afraz, M. Ruffini, and T. Bauschert, “Collaborative cyber-attack defense in SDN networks using blockchain technology,” *Proceedings of the 2020 IEEE Conference on Network Softwarization: Bridging the Gap Between AI and Network Softwarization, NetSoft 2020*, no. June, pp. 487–492, 2020, doi: 10.1109/NetSoft48620.2020.9165396.

[6] M. I. Khalid *et al.*, “A Comprehensive Survey on Blockchain-Based Decentralized Storage Networks,” *IEEE Access*, vol. 11, pp. 10995–11015, 2023, doi: 10.1109/ACCESS.2023.3240237.

[7] A. Manoj Athreya *et al.*, “Peer-to-Peer Distributed Storage Using InterPlanetary File System,” *Advances in Intelligent Systems and Computing*, vol. 1133, no. January, pp. 711–721, 2021, doi: 10.1007/978-981-15-3514-7_54.

[8] T. V. Doan, Y. Psaras, J. Ott, and V. Bajpai,

Towards Decentralised Cloud Storage with IPFS: Opportunities, Challenges, and Future Directions, vol. 1, no. 1. Association for Computing Machinery, 2022. [Online]. Available: http://arxiv.org/abs/2202.06315

[9] D. Trautwein *et al.*, "Design and evaluation of ipfs: A storage layer for the decentralizedweb," *SIGCOMM 2022 - Proceedings of the ACM SIGCOMM 2022 Conference*, pp. 739–752, 2022, doi: 10.1145/3544216.3544232.

[10] I. Permatasari, M. Essaid, H. Kim, and H. Ju, "Blockchain implementation to verify archives integrity on cilegon E-archive," *Applied Sciences (Switzerland)*, vol. 10, no. 7, 2020, doi: 10.3390/app10072621.

[11] A. G. Cristea, L. Alboaie, A. Panu, and V. Radulescu, "Offline but still connected with IPFS based communication," *Procedia Comput Sci*, vol. 176, pp. 1606–1612, 2020, doi: 10.1016/j.procs.2020.09.183.

[12] M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State, "Blockchain-Based, Decentralized Access Control for IPFS," *Proceedings - IEEE 2018 International Congress on Cybermatics: 2018 IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, iThings/Gree*, no. July, pp. 1499–1506, 2018, doi: 10.1109/Cybermatics_2018.2018.00253.

[13] S. Y. Lin, L. Zhang, J. Li, L. li Ji, and Y. Sun, *A survey of application research based on blockchain smart contract*, vol. 28, no. 2. 2022. doi: 10.1007/s11276-021-02874-x.

[14] H. Huang, J. Lin, B. Zheng, Z. Zheng, and J. Bian, "When Blockchain Meets Distributed File Systems: An Overview, Challenges, and Open Issues," *IEEE Access*, vol. 8, no. March, pp. 50574–50586, 2020, doi: 10.1109/ACCESS.2020.2979881.

[15] N. Sangeeta and S. Y. Nam, "Blockchain and Interplanetary File System (IPFS)-Based Data Storage System for Vehicular Networks with Keyword Search Capability," *Electronics (Switzerland)*, vol. 12, no. 7, 2023, doi: 10.3390/electronics12071545.

[16] A. Mehbodniya, R. Neware, S. Vyas, M. R. Kumar, P. Ngulube, and S. Ray, "Blockchain and IPFS Integrated Framework in Bilevel Fog-Cloud Network for Security and Privacy of IoMT Devices," *Comput Math Methods Med*, vol. 2021, 2021, doi: 10.1155/2021/7727685.

[17] M. Patel and N. Patel, "Exploring Research Methodology," *International Journal of Research and Review*, vol. 6, no. 3, pp. 48–55, 2019.

[18] E. Politou, E. Alepis, C. Patsakis, F. Casino, and M. Alazab, "Delegated content erasure in IPFS," *Future Generation Computer Systems*, vol. 112, pp. 956–964, 2020, doi: 10.1016/j.future.2020.06.037.

[19] M. M. Arer, P. M. Dhulavvagol, and S. G. Totad, "Efficient Big Data Storage and Retrieval in Distributed Architecture using Blockchain and IPFS," *2022 IEEE 7th International conference for Convergence in Technology, I2CT 2022*, no. April, 2022, doi: 10.1109/I2CT54291.2022.9824566.