VAULT: A Scalable Blockchain-based Protocol for Secure Data Access and Collaboration

1st Justin S. Gazsi*
School of Computing and Information Sciences
Florida International University
jgazs001@fiu.edu

2nd Sajia Zafreen*
Department of Computer Science
Boise State University
sajiazafreen@u.boisestate.edu

3rd Gaby G. Dagher
Department of Computer Science
Boise State University
gabydagher@boisestate.edu

4th Min Long
Department of Computer Science
Boise State University
minlong@boisestate.edu

***Abstract*—Data sharing is as vital as data storage. Existing centralized data sharing and access systems provide less transparency and traceability as the users have to trust a centralized authority and its decision making for the entire system. There is a need for decentralized distributed data storage and access without a central authority. Blockchain provides promising solutions to such needs. However, the existing decentralized blockchain-based solutions are complex and involve financial incentives, which limits their applications. We propose a secure permissioned blockchain-based decentralized system, VAULT, with a novel quorum-based consensus. We store encrypted files using Interplanetary File System (IPFS) and the references to the files in the blockchain. VAULT is designed for applications involving collaboration from multiple permissioned parties, and users can store, access, and share data as well as manage projects through blockchain. Our experimental results show that our quorum selection is fair, and the VAULT protocol is scalable.**

***Index Terms*—Blockchain, IPFS,Distributed Storage, Data Access and Sharing**

I. Introduction

Data sharing has become a central activity in many applications. There are various data sharing systems available such as Google Drive[1], Google Cloud Life Sciences (for scientific data)[2], Microsoft OneDrive [3], Microsoft Key Vaults (for sensitive data)[4], Dropbox[5], and iCloud[6]. However, some systems are not designed to ensure data integrity, security and transparency [1]. This issue cannot be alleviated by using enterprise level systems like Microsoft Vaults or Google Cloud Life Sciences which are originally designed to support sensitive and scientific data sharing. The researchers must trust these systems, making the data vulnerable to data loss, leakage or malware attacks [2]. A recent example is from Facebook, a social media - another centralized data storage system, where 550 million user data were breached in 2021 [3] .

In contrast to the centralized storage solutions, decentralized storage, such as blockchain-based systems, doesn't depend on a specific entity, and thus can intrinsically avoid a single point of failure. Blockchain can be permissionless (e.g. Bitcoin [4] or Ethereum [5]) where anyone can join the network. However, in a permissioned blockchain (e.g. Hyperledger Project by the Linux Foundation [7]), only known or certified entities are added to the network [6]. Thus, it provides more security by reducing malicious nodes in the network.

There are multiple peer-to-peer decentralized cloud storage systems (e.g. STORJI[8], Sia[9], filecoin[10]) available, which are quite complex and can add additional costs and needs for maintenance for activities like research collaboration. Thus, researchers or developers still need a decentralized, secured file share system where users can openly collaborate and control data sharing through a group consensus. In [7], a permissioned blockchain technology for data storage and access is designed, while the files' metadata are stored in a blockchain. This work primarily focuses on replacing the centralized repository but not on efficient file access and sharing. In [8], to store and access relatively large data and group key management, a centralized storage medium—an IPFS proxy is used to verify and provide end-to-end transparency, leading to the risk of a single entity controlling the decision making.

In this paper, we propose and design a blockchain-based decentralized system for file sharing, access and storage in a secure environment, mainly for collaborations among multiple parties or nodes. For a node to join the system, it should authenticate itself through a certificate authority to reduce malicious activity. The file upload, access and update will take place through transactions on the blockchain. The system is designed to be trustless, where nodes do not have

*These authors contributed equally.
[1]www.drive.google.com
[2]www.cloud.google.com/life-sciences
[3]www.onedrive.live.com
[4]www.azure.microsoft.com/en-us/services/key-vault
[5]www.dropbox.com
[6]www.icloud.com
[7]www.hyperledger.org
[8]www.storji.io
[9]www.sia.tech
[10]www.filecoin.io

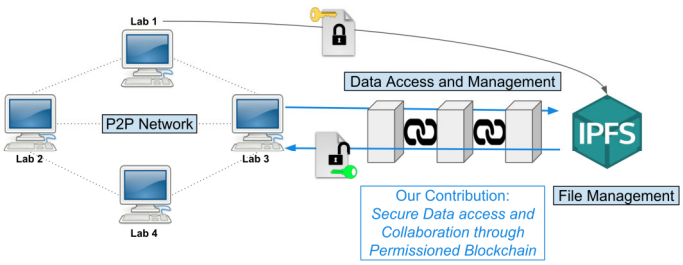

Figure 1: An overview of blockchain-based, file access and management system VAULT.

to trust other nodes to share or access files. The transactions will be validated through consensus, where the validating nodes are selected by seeding the hash of the last valid block to a quorum generation function to ensure randomness in the selection process. The ownership of the files can be traceable as the previous transactions will be recorded in the ledger providing more transparency on how the system works. Since blockchain is not meant to store large data [9], we utilize the IPFS [10] to store files in a peer-to-peer distributed system. Anyone connected to and can deploy the IPFS can view the files stored in IPFS. The files should be encrypted before added to IPFS to ensure data confidentiality [11] [12]. In Figure 1 we present an overview of our protocol representing the collaborating parties as a peer-to-peer network where each node can manage data by storing encrypted files and access files stored in IPFS via blockchain.

The contributions of this paper are as follows:

- We propose a novel permissioned blockchain-based protocol named VAULT, that establishes a collaborative environment requiring no financial incentive where data owners can securely share data and access other shared data through a quorum-based consensus.
- We propose an efficient quorum based consensus protocol for block mining. To ensure transparency and trust regarding data sharing and access, quorum members validate and synchronize their transaction lists, mine the next block, and then announce it to the network.
- We achieve traceability and ownership in VAULT by storing all interactions details of the files as transactions in the blockchain.
- We implement the consensus mechanism of VAULT to simulate the file access and sharing network. Our experimental results demonstrate that our quorum selection protocol is fair for mining loads. They also show that the VAULT protocol is scalable for increasing nodes and transactions in the network.

II. Related work

In contrast to a permissionless blockchain, a permissioned blockchain requires a party to acquire permission or certificate to participate in the system and thus pertains to more control or more centralized comparing to permissionless blockchain [13] [14]. A consensus protocol is required for both permissioned and permissionless blockchain-based data share and access system. In a consensus protocol, there can be financial incentives involved, such as Ethereum smart contracts [15]. Several studies [9] [16] [17] [18] [19] have used the Ethereum smart contract to achieve decentralized data storage using IPFS, . [8] uses both permissionless blockchain and an IPFS proxy to control most functions. [20] uses filecoin as the proof of storage. Although blockchain and distributed P2P storage —IPFS can avoid centralization, a centralized application [18] or a centralized IPFS proxy [8] can cause a centralized system. There are some other studies where blockchain is used only for secure storing purposes. In [7] [21] [22] [23] [24] a distributed blockchain-based data storage system is designed. An Ethereum based smart contract is used in [21] [24]. In [22] a user interface design helps with record-keeping with *Storji* Network. Even though a distributed decentralized storage provides data confidentiality and more transparency than a centralized storage system, additional features like secure data sharing and a distributed storage system are still needed.

Our protocol, VAULT, focuses on collaboration activities commonly seen in research, development and other communities and does not address the need for financial incentive, which is different from the above approaches. Anybody who wants to use our system has to be responsible for mining the blocks in blockchain. We use a quorum-based approach and ensure randomness in quorum selection for fairness. Unlike these approaches except [19], we record interactions between users as blockchain transactions, such as accepting a new party in the network, adding files, accessing and collaborating files in blockchain. Our design does not involve a central authority that controls the users' interactions and can effectively avoid a single point of failure.

III. Problem formulation

We focus on designing a trustless system that is traceable, and has more transparency in terms of contribution of the network's nodes, since there is a need for a simple decentralized system consisting of a consensus protocol without any financial incentives. The system consists of a network of nodes and each node N_i has to acquire a certificate from a certified authority to enter the network. The node N_i can upload its encrypted file, F_i to IPFS and receives the corresponding CID, C_i. We expect these nodes to store the file CID identifying the owner, access other files, and update files through blockchain. In the process, the nodes will not have to trust each other. A quorum of nodes will ensure the next block has valid transactions. The designed system—VAULT assumes all parties could be malicious but do not collude with each other. VAULT utilizes a certificate authority to monitor each node and prevent activities adversely from them.

IV. Solution: VAULT protocol

We design VAULT, a permissioned blockchain where every operation to access and share data will be recorded in blockchain. Using this protocol, a user can create a project,

accept member nodes to the project, add files, update files, give access to other users to their files through blockchain. The files must be encrypted and obtained a CID when added to IPFS to avoid computational overhead in those frequent operations. Other annotations to references are listed in Table I for discussions in the following sections.

Table I: References of terms used in the VAULT protocol.

Notations	Descriptions
NID	Node ID
FID	File ID
CID	Content ID
PID	Project ID
Sig()	Signature Function
Pk	Private Key
PubK	Public Key
Tx	Transaction

A. Add a member to the network

Figure 2 presents how a new member acquires a certificate from the certificate authority and join the network. After the certificate is obtained, the new node broadcasts the certificate to the network and other existing members in the network can facilitate the propagation. Then the existing members in the network will validate the certificate and broadcast a new transaction of the addition of the new member to the mempool, a mechanism for storing information on unconfirmed transactions. When the transaction is validated by the quorum and added to the blockchain, a confirmation signal is sent to the new member and the addition of a new member to the network is completed.

B. Create a new project

A member node has to initiate a project creation transaction, which acts as the root transaction of the project on the blockchain to create a new project. The transaction includes PID, NID of the node creating the project and a timestamp of when the project is created. The project transaction with all information about the transaction will then be broadcasted through the network. When the broadcast is in the mempool, the quorum collects the transaction, validates the digital signature of the transaction, validates if the PID is unique and ensures a valid member of the network creates the project. If the transaction is valid, it is added to a block which will be added as the next block in the blockchain.

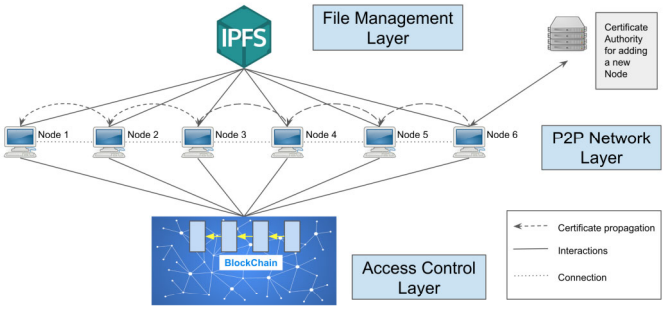

Figure 2: Propagation of certificate for adding a new member through the P2P network.

C. Add a new file

IPFS rather than the blockchain is suitable for storing the actual content of large data files. The owner encrypts the file using the owner's private key and receives a CID after adding the encrypted file to IPFS. To add the file reference or the CID to blockchain, the owner will create a transaction including CID, FID, a timestamp and a digital signature by the owner. After that, the owner will broadcast the transaction through the network. The quorum members pick up the transaction from the mempool and validate the transaction by the owner's digital signature. If the transaction is valid by reaching through quorum consensus, the transaction is added to the next block in the blockchain.

D. Update a new file

We design our protocol so that every user is the owner of files who has uploaded and a co-owner of files who has made collaborative contributions to the file. Both owner and co-owners are editors who can change a file. After an editor, edits a file and adds the encrypted file to IPFS, receives a new CID for the corresponding file. The editor then signals the previous owner or editors of the file and waits for the approval from them. When the majority or $51\,\%$ of the previous owner or editors approve the file, the editor becomes a new co-owner and signs the file. After that, the new editor follows the protocol of adding a file to the blockchain. In such cases, the new editor first broadcasts the new file as a new transaction with the CID, FID, and timestamp. The latest CID serves as the pointer to the latest canonical version of the file. When the quorum validates the transaction, it is added to the next block in the blockchain. The FID keeps track of the number of CIDs or the versions of the file and remains unchanged, while a new CID is created in every new edit.

E. Give access to a file

A VAULT user can request access to a file. The user needs to add the FID with the user's digital signature and broadcast the transaction to request access. The transaction is validated by the quorum and added to the next block in the blockchain. Then the owners of the requested file get notified, and any of the owners can grant access to the file by providing a key-wrap using the Sig() function with the owner's private key (Pk) and a CID. When this grant access transaction is broadcasted to the network, the quorum validates the owner's digital signature and adds the transaction to the next block of the blockchain. Then the applicant obtains the key wrap and uses the owner's public key to unwrap the CID and access the file.

F. The quorum consensus protocol

To make the validation process more efficient and fair, we use a quorum based approach involving randomness in the quorum selection process. In Figure 3 we show that the

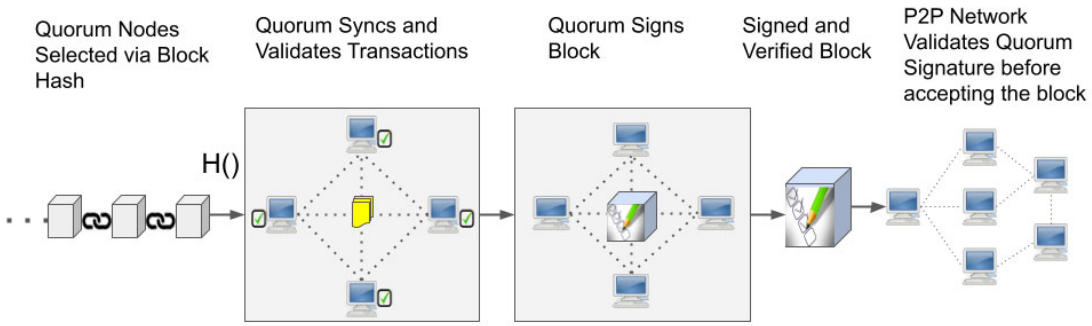

Figure 3: Transaction propagation and block creation through quorum consensus.

hash of the previous block—the *H()* is used as the seed of the random quorum group generator. The number of quorum members can be dynamic or static depending on the user needs. After selecting the group number of quorum members, the numbers will be generated, and each quorum member will be notified. The quorum member will then access the mempool to validate the transaction. When a threshold number of quorum members agree on a list of transactions, a consensus is reached. After that, the quorum collectively synchronizes the validated transactions. The last quorum member who gets the list of validated transactions creates the block, signs it and broadcasts it for other quorum members to sign. When the quorum signs the block, it is broadcasted to the peer-to-peer network, and the nodes outside of the quorum validate if all the members of the quorum signed the block. If the block is valid, every node updates its version of the ledger, the block is added to the blockchain.

G. Block and transaction

When the threshold number of quorum members reach a consensus, the quorum members collectively create a new block and sign the block. This block contains a block header, transactions in the block body and a footer. The block header has the block number, node ID of the quorum member who created the block, and the list of quorum member signatures. The block footer contains the timestamp, previous block's hash and the current block's hash. The body of the block contains the list of transactions validated by the threshold number of quorum members. The overall block structure is shown in Table II. A general view of the transaction is shown in *Transaction A*. Each transaction contains a type code that ensures the type of the transaction, such as creating a new project, adding a member, adding a file, updating a file, and giving access to the file. The transaction also contains the file name, the FID, owners' list, CID or the IPFS hash and a timestamp of when the transaction is created.

Table II: Block structure.

Block Number
Node ID
Quorum Signature List
Transaction A
Transaction Type Code: Add file \Update file \Access file
File Name\Description
File ID
IPFS Hash
Owners' List
Time Stamp
Transaction B
... ...
Transaction C
... ...
Previous Block hash
Block hash
Time Stamp

V. Experimental evaluation

The experiment of Network and Protocol was programmed in Java, using Intel Core i5-8265U CPU @ 1.60 GHz x 8, 8GB RAM, and 256GB SSD storage on Manjaro Linux 21.1.0. The full source code can be accessed from here[11].

A. Fairness of quorum selection

We achieve fairness of the quorum selection by leveraging randomness. Quorum selection in the protocol is determined by a random construction of individual nodes on the network as follows. The experiment uses the hash value of the latest valid block on the blockchain as a seed and Java's built-in `java.util.random` to generate a list of network nodes of the appropriate quorum size. No central party is required for the creation of quorum as the latest block hash is publicly available allowing nodes to arrive a random, yet identical list of nodes to serve as the quorum. We demonstrate the fairness of this quorum selection process by averaging the amount of times a node is selected to be in a quorum over 1000 selections, which serves as 1 trial and the trial ran for 1000 times. We compare "our quorum selection" with the

[11]VAULT Source Code: https://drive.google.com/drive/folders/1s7IPdrxZsk9BmcG8PabvLLnCqg11gIy2?usp=sharing

second experiment: "random quorum selection", which ran using Java's `java.util.random` function without a seed to select nodes. We find that our protocol performs least as well as random quorum selection as shown in Figure 4. While randomness achieves ideal fairness in the long term, it can display bias in the short term allowing for certain nodes to be selected more often than others to be validators of transactions.

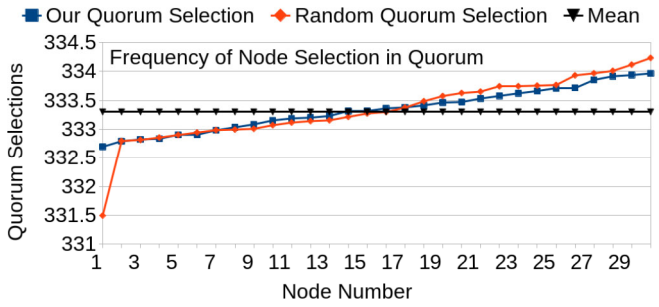

Figure 4: Fairness of quorum selection, average of 1000 quorum selections over 1000 trials.

B. Scalability

Evaluation of the protocol was performed in Java to experiment with various aspects of the network, including number of total nodes in the network, time to broadcast transactions, time for quorum to validate transactions, and for the minted block to be propagated throughout the network. Nodes are connected to a random number of peers with a peer connectivity target of 10. Quorum size was determined to be 1% of the network during the testing phase. We then calculated the run times of 50 transactions per block over an average of 50 blocks as a function of a network size of 1000 to 5000 nodes. Figure 5 displays the scalability of all components of the protocol and we found them to be linear.

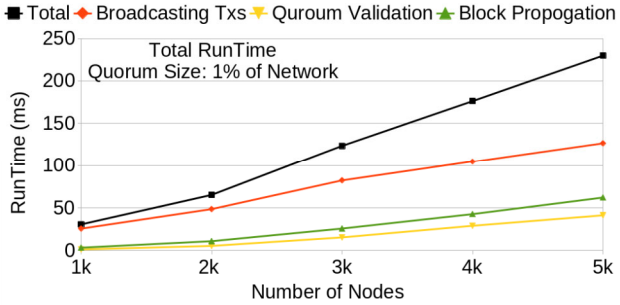

Figure 5: Scalability of protocol, average of 50 blocks with 50 transactions per block.

C. Component breakdown with constant quorum size

We further broke down the run times of the various components of the protocol with a constant quorum size of 5 to investigate how differing transaction throughput affects network performance. The breakdown consists of measuring the time for block creation (the time it takes for transactions to propagate through the network and the quorum to arrive at a consensus vote) and the propagation of the block back into the network of nodes. In this experiment, each quorum member has a different view of the mempool of transactions and a transaction the other quorum members does not have. Thus, we take into account the time for each node to check all transactions of quorum members and syncronize the transactions with each other to arrive at a common list of transactions to be validated. As network size and transaction throughput increases, the proportion of time required for the broadcasting and validation of transactions increases as well when comparing the increased transaction throughput in Figure 6 (a), (b), and (c). The block creation, that is broadcasting of transactions plus the time for the quorum to validate the transactions, remains linear as can be seen in the total runtimes of Figure 6 (a), (b), and (c).

D. Transaction scalablility

Experiments show that the protocol is efficient. As the number of transactions is increased for a given network size, an overall linear increase is achieved, in terms of run-time from initial broadcasting of transaction to block creation and propagation in the network broadcasted. Figure 7 also shows that for increasing of the network size for any given transaction throughput, efficiency is proportional to the increasing of nodes on the network.

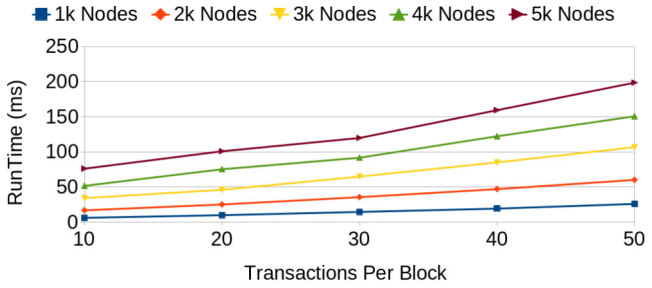

Figure 7: Scalability w.r.t. transactions per block.

VI. Conclusion and Future Work

In this paper, we provide a decentralized blockchain-based protocol VAULT with a novel quorum-based consensus, as a solution for simple, secure data access, share and storage plan. It is designed for applications involving collaboration from multiple permissioned parties, such as project management and collaboration among researchers or developers. Our protocol shows that the quorum selection is fair and scales linearly with increasing network size or transactions per block based on experiments. As a future work, it would be interesting to explore how to improve fairness of the consensus protocol by implementing measures. One example is grey-listing, which means nodes that have participated in a quorum may not take action until an established number of blocks have been mined since their last quorum selection. The probability of a node getting selected in a quorum should increase with the number of blocks mined since its last selection.

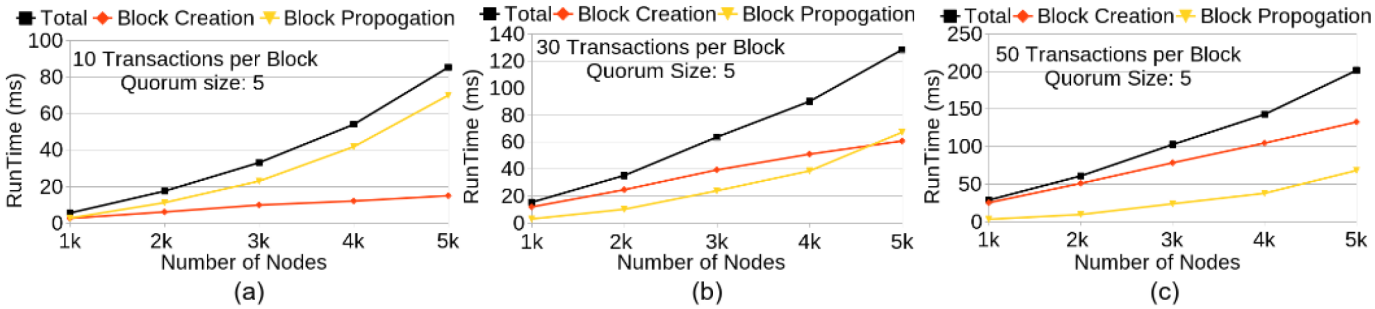

Figure 6: Scalability of protocol with various number of transactions per block.

References

[1] S. Kowalczyk and K. Shankar, "Data sharing in the sciences," *Annual review of information science and technology*, vol. 45, no. 1, pp. 247–294, 2011.

[2] T. Jiang, X. Chen, J. Li, D. S. Wong, J. Ma, and J. K. Liu, "Towards secure and reliable cloud storage against data re-outsourcing," *Future Generation Computer Systems*, vol. 52, pp. 86–94, 2015.

[3] O'FlahertyS, Kate, "Facebook Data Breach: Here's What To Do Now," https://www.forbes.com/sites/kateoflahertyuk/2021/04/06/facebook-data-breach-heres-what-to-do-now/.

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[5] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[6] M. E. Peck, "Blockchain world-do you need a blockchain? this chart will tell you if the technology can solve your problem," *IEEE Spectrum*, vol. 54, no. 10, pp. 38–60, 2017.

[7] S. Ali, G. Wang, B. White, and R. L. Cottrell, "A blockchain-based decentralized data storage and access framework for pinger," in *TrustCom/BigDataSE*. IEEE, 2018, pp. 1303–1308.

[8] H.-S. Huang, T.-S. Chang, and J.-Y. Wu, "A secure file sharing system based on ipfs and blockchain," in *Proceedings of the 2020 2nd International Electronics Communication Conference*, 2020, pp. 96–100.

[9] M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State, "Blockchain-based, decentralized access control for ipfs," in *Congress on Cybermatics*. IEEE, 2018, pp. 1499–1506.

[10] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.

[11] T. Ge and S. Zdonik, "Answering aggregation queries in a secure system model," in *VLDB*, 2007, pp. 519–530.

[12] R. A. Popa, C. M. Redfield, N. Zeldovich, and H. Balakrishnan, "Cryptdb: Protecting confidentiality with encrypted query processing," in *SOSP*, 2011, pp. 85–100.

[13] K. Wüst and A. Gervais, "Do you need a blockchain?" in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018, pp. 45–54.

[14] G. D. Bashar, A. A. Avila, and G. G. Dagher, "Poq: A consensus protocol for private blockchains using intel sgx," in *SecureComm*. Springer, 2020, pp. 141–160.

[15] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, 2014.

[16] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *Ieee Access*, vol. 6, pp. 38 437–38 450, 2018.

[17] M. Naz, F. A. Al-zahrani, R. Khalid, N. Javaid, A. M. Qamar, M. K. Afzal, and M. Shafiq, "A secure data sharing platform using blockchain and interplanetary file system," *Sustainability*, vol. 11, no. 24, p. 7054, 2019.

[18] S. Khatal, J. Rane, D. Patel, P. Patel, and Y. Busnel, "Fileshare: A blockchain and ipfs framework for secure file sharing and data provenance," in *Advances in Machine Learning and Computational Intelligence*. Springer, 2021, pp. 825–833.

[19] L. Sari and M. Sipos, "Filetribe: Blockchain-based secure file sharing on ipfs," in *European Wireless 2019; 25th European Wireless Conference*. VDE, 2019, pp. 1–6.

[20] S. Vimal and S. Srivatsa, "A new cluster p2p file sharing system based on ipfs and blockchain technology," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–7, 2019.

[21] Z. Ning, L. Xiao, W. Liang, W. Shi, and K.-C. Li, "On the exploitation of blockchain for distributed file storage," *Journal of Sensors*, vol. 2020, 2020.

[22] S. Wilkinson, J. Lowry, and T. Boshevski, "Metadisk a blockchain-based decentralized file storage application," *Storj Labs Inc., Technical Report, hal*, pp. 1–11, 2014.

[23] M. J. M. Chowdhury, A. Colman, M. A. Kabir, J. Han, and P. Sarda, "Blockchain as a notarization service for data sharing with personal data store," in *TrustCom/BigDataSE*. IEEE, 2018, pp. 1330–1335.

[24] J. Xue, C. Xu, Y. Zhang, and L. Bai, "Dstore: a distributed cloud storage system based on smart contracts and blockchain," in *ICA3PP*. Springer, 2018, pp. 385–401.