2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit) | 978-1-6654-1526-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/EUCNC/6GSUMMIT51104.2021.9482449

Scalable Storage Scheme for Blockchain-Enabled IoT Equipped Food Supply Chains

Janitha Rupasena*, Tharaka Hewa[†], Kasun T. Hemachandra[‡], Madhusanka Liyanage[§]
*[‡][§]Department of Electronic and Telecommunication Engineering, University of Moratuwa, Sri Lanka
[†][§]Centre for Wireless Communications, University of Oulu, Finland
[§]School of Computer Science, University College Dublin, Ireland
Email: *janitha.rupasena@gmail.com [†][§][firstname.lastname]@oulu.fi, [‡]kasunh@uom.lk, [§]madhusanka@ucd.ie

***Abstract*—Blockchain is an innovative technology which enabled new applications for solving numerous problems in distributed environments such as the internet of things (IoT) equipped food supply chains (FSCs). In FSCs, the large volume of IoT data such as audio, video, images, and sensor data will be transferred to ensure the traceability of the food to its source. When blockchain technology is used in an FSC, the storage requirements in the nodes will grow with time, since blockchain only allows information adding, without deleting the already stored information. Therefore, it is evident that offchain storage offers more flexibility than onchain storage in IoT equipped supply chains. This paper proposes a scalable storage scheme using offchain storage, where the data from IoT devices in the supply chain will store offchain. To reduce the growth of offchain storage, we exploit the fact that some information regarding a particular food item may not be required after the expiration date. The scalability of the proposed scheme is validated through numerical and experimental results.**

***Index Terms*—blockchain, food supply chain, offchain storage**

I. Introduction

Supply Chains (SCs) play a major role in transferring goods or services from the initial supplier to the end customer[1]. To gain end consumers' trust, SC authorities need to maintain the quality, integrity, and credibility of the entire SC. In a food supply chain, efficient monitoring of the food products in each stage of the SC is crucial to ensure food safety. As consumers and regulatory bodies are becoming increasingly concerned about food quality, the concept of traceability in the SC has become a critical requirement.

The use of internet of things (IoT) devices and technologies in food supply chains (FSCs) has surged recently, leading to substantial innovation and research towards developing auditable, transparent, reliable tracing systems for FSCs. Conventional IoT-based tracking are built on top of centralized infrastructures, leading to limitations such as, data integrity, tampering, and single point of failures [2]. However, in FSCs, to maintain the integrity of the products along the SC, it is essential for the stored records to be tamper-proof. Therefore, it is preferred not to rely on any centralized third-party intermediary. As a potential solution for the issues related to centralized architectures, blockchain technology, which is a peer-to-peer digital ledger that does not rely on centralized services, has been considered in recent works related to SCs [3]. In the case of blockchain, all the records are based on a consensus reached at least by the absolute majority of the peers within the network itself. This decentralized ledger is unchanging by design and provides a source of information which auditable as well as transparent. Therefore, blockchain can be effectively incorporated into SCs to maintain transparency and immutability in a distributed manner. Furthermore, from an IoT perspective, the sensors in a blockchain-based FSC would only require stable connection to their closely located peer, eliminating the need for connection with a centralized cloud. Thus, it is evident that blockchain can be effectively used in IoT equipped FSCs to make traceable data available at every step of the SC[4].

When a large number of IoT devices are used in an FSC, the amount of transaction data that will be stored in the blockchain will also be extremely large. This data may include audio, video, sensor data, and images. Therefore, the blockchain ledger storage will grow continuously, requiring large storage volume on each node, leading to higher costs. This may significantly affect the scalability of the SC. To circumvent this, offchain storage models have been proposed to store the incoming transaction data, and only the hash of the corresponding data, which is of significantly smaller size compared to the original data, could be store in the blockchain. Smart contracts as well as distributed hash tables (DHTs) can be used to establish a link between hash found in the chain and the actual physical storage [5]. A distributed file system among the already available nodes is used to maintain the offchain storage [1].

Since IoT data will be stored in an offchain storage, the offchian storage will also grow with time. However, it is well understood that traceable data of a particular food product may not be required after a certain period of time (eg. the expiration date). Therefore, in this paper, we exploit this fact and propose a technique to reduce the growth rate of the offchain storage in a blockchain enabled IoT equipped FSC. To the best of our knowledge, this is the first work providing an end to end solution for IoT equipped FSCs with a recyclable offchain storage mechanism. The proposed solution uses smart contracts, distributed file systems with distributed hash tables, and distributed database to assure an efficient, secure, and trusted environment. We validate our claims using numerical

978-1-6654-1526-2/21/$31.00 ©2021 IEEE

results and evaluate the proposed system with a prototype implementation using Hyperledger blockchain platform.

The remainder of this paper is organized as follows. Related work and the contributions are given in Section II. The proposed methodology is presented in Section III. Numerical and experimental results are provided in Section IV and Section V, respectively, while Section VI concludes the paper.

II. Related Works

The utilization of blockchain in SCs has been proposed in several works. The authors in [6] proposed a scheme for governance on the drug supply chain via Gcoin blockchain. The double-spending prevention mechanism in Gcoin blockchain is exploited to identify the counterfeit drugs. Reference [7] proposed a blockchain based aircraft parts supply chain to improve traceability and to establish the authenticity of spare parts. However, the paper does not discuss the implementation aspects.

In relation to IoT equipped FSCs, in [8], an FSC system is established for China, based on radio frequency identification (RFID) and blockchain technology. All aspects of data gathering and information management in each link of the FSC, which enables monitoring, and traceability for the quality and safety of the food ”from farm to fork” was discussed. Authors of [9] proposed a scheme for the use of blockchain in SC management for rice. Reference [10] proposed a traceable scheme based on Hazard Analysis and Critical Control Points (HACCP), blockchain and IoT. To solve scalability issues due to storage growth, BigChainDB is used.

When the number of transactions in a blockchain increases, the growth in storage requirements puts limitations on system scalability. A thorough survey on scalable blockchain solutions is given in [11]. Some popular solutions include Sharding [12], Sidechain [13], and cross-chain [14]. A storage optimization mechanism based on a residual number system, which reduces the storage volume on each node, is proposed in [15]. In addition, the recovery procedure of the new Chinese remainder theorem (CRT-II) is utilized to capture garbled data from devil nodes. This enables the proposed storage mechanism with strong fault tolerance capability. In [16], an innovative Inter Planetary File System (IPFS) based storage model is proposed for blockchain. However, when the volume of transactions grows, the requirement for IPFS storage also grows significantly.

After reviewing the existing work addressing the storage growth of blockchain, it is evident that many solutions rely on offchain storage. However, these studies did not consider the storage growth in the nodes inside onchain and offchain environments. Therefore, in this paper we address this issue by proposing a storage recycling scheme for blockchain enabled IoT equipped FSCs. The proposed solution uses smart contracts to assure an efficient, secure and trusted environment for SC activities. The main contribution of the proposed approach is its ability to maintain the offchain storage growth at a slower rate by deleting the stored data of food products after its expiration date. The performance of the proposed scheme is evaluated numerically using actual blockchain data and its feasibility is verified through an experimental implementation. Table I highlights the differences of the proposed scheme compared to the existing solutions.

TABLE I
Comparison of Proposed Solution with Related Works

Features	[8]	[9]	[10]	[15]	[16]	Proposal
Traceability	✓	✓	✓	✓	✓	✓
Accountability	✓	✓	✓	✓	✓	✓
Credibility	✓	✓	✓	✓	✓	✓
Authenticity	✓	✓	✓	✓	✓	✓
Scalability	✗	✗	✗	✓	✓	✓
Storage Recycling	✗	✗	✗	✗	✗	✓

III. Proposed Architecture and Methodology

In this section, we propose a layered architecture to establish a blockchain enabled IoT equipped FSC, with storage recycling feature, and describe the end-to-end system functionality.

Fig 1 illustrates the system layers and their components. The data layer (DL) extracts the information related to the interactions among the entities in the FSC, such as trading of raw materials, products, and services, along with a proof of an auditable delivery. In each stage of the FSC, IoT devices are placed to acquire the data, including invoices, videos, audio clips, images, and critical sensor readings, to be transferred to upper layers.

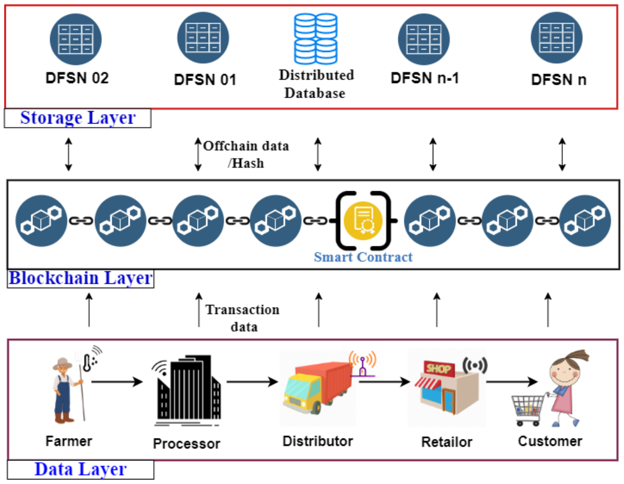

Fig. 1. Blockchain-based end to end solution for food supply chain

The blockchain layer (BL) handles the transaction data of trading goods and processes. To reduce the amount of onchain storage required, BL only stores the hashes of the data, while the actual data are stored on the storage layer. The BL uses strict access control mechanisms to prevent unauthorized access to the data in the storage layer.

The storage layer (SL) is responsible for storing the transaction data of the blockchain in immutable Distributed File System Networks (DFSNs) and immutable Distributed Database System (DDS). The Distributed File System (DFS) is a decentralized storage medium and it leverages the proposed system with low latency, high throughput, and scalability [16]. All the transaction data including invoices, videos, audio clips, images and sensor data will be stored in the relevant DFSN, and a hash value corresponding to the respective content will be returned to the BL.

The main functionalities of the proposed system are described in Subsections .

A. Product Data Collection

Similar to a conventional IoT equipped FSC, transaction data related to the traceability of the products will be collected and recorded. In the proposed scheme, we pay special attention to the unique product identifier (PID), product expiry date (PED) and the dispute reason field (DRF). The DRF is used to indicate whether the product faced issues such as legal action, large quantities remained unsold, or any other incident that may compromise the quality of the product.

B. File System Creation

Consider a time span of T as the data retention period (DRP) of the system. The DRP is selected based on the life cycle of the products in the FSC. The DRP can be divided in to n slots of equal duration T/n. The slot duration determines the frequency of storage recycling within DRP. To store the data, $n+1$ DFSNs are pre-configure across all the nodes, and unique identifiers Id0 to Idn will be assigned to them. For the remainder of the paper, we set DRP to 1 year, and perform storage recycling weekly. Therefore, n is set to 52. Each week will be mapped to a separate DFSN and a 'special week id' will be mapped to DFSN Id0. Then DDS consisting two tables, namely, T1 and T2 will be configured across all nodes. T1 is used to keep the mapping of the week identifier (id) (assigned based on the system initiation date) and the corresponding DFSN name while T2 is used for storing every transaction record in the system.

C. Storing Data in DFSN and DDS

When the miners get the transaction data from the DL, they first validate the transactions, check the product expiry date and calculate the current week Id according to the system initiation date by executing smart contracts. Then, the corresponding DFSN name is obtained by querying the DDS T1, and the data will be stored in the DFSN corresponding to the identified week, and the hash will be returned. Thereafter, these data will be saved in a T2 entry, containing timestamp, hash value, product id, expiry week id, DFSN name and dispute reason (if any), by executing smart contracts. When generating the next block, each miner stores the returned DFS hashes of the verified transactions in the new block. Then it calculates the Merkle root and the block hash. The DFSN based storage model for blockchain is shown in Fig 2.

D. Building the blockchain

As the Fig 2, in the case of miner A successfully calculating the block hash that meets the difficulty, that block will be broadcasted to miners B, C and D. Miners B, C and D will need to verify the transactions and the block hash. In fact, during the mining process, miners B, C, D would have received mostly the same transactions as miner A would have. Some differences between the local transaction pools of the miners can be cause by network transmission delays. Hence, most of the transaction DFS hashes in the newly received block are the same as the hashes in the local pool of transactions in miner B, C and D. In the case where the DFS hash of a particular transaction in the block sent by miner A is available in the local transaction pool, miners B, C and D can determine it as a transaction confirmed by them before. Then there will be no need to download it from DFSN. [16]. For the rest of the transactions, the data needs to be requested from the DFSN through their corresponding DFS hashes. Before requesting the data from the DFSN, each miner needs to know in which DFSN the corresponding hash has been saved. For that it needs to query the DDS T2 via the hash and get the DFSN name and then download the data through their corresponding hashes. Then the validity of transactions and the block would be confirmed. Afterwards the new block can be appended to the blockchain ledger.

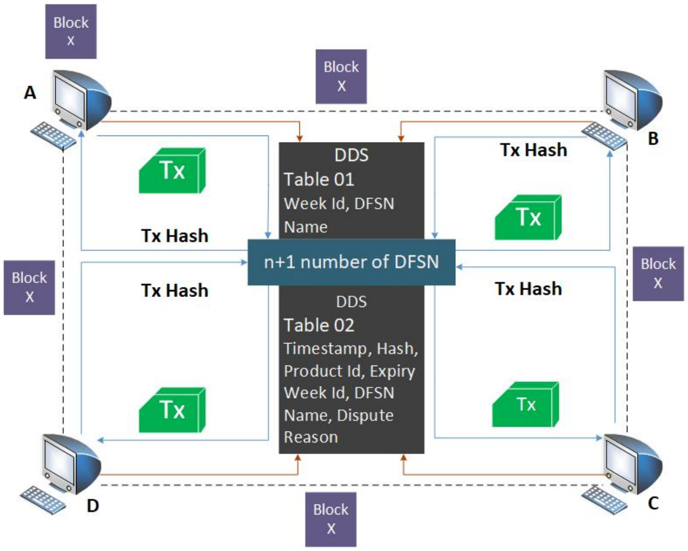

Fig. 2. DFSN based Storage Model

E. Storage Recycling

At the end of each week, a smart contract will be triggered to identify the products whose expiration date falls within that week. A new DFSN will be created for the corresponding week in the subsequent year, and the data deletion process will be initiated. The smart contract obtains the DFSN name corresponding to the current week by querying DDS T1. Furthermore, it checks T2 to find out whether there are any disputes regarding the transactions related to any of the products expire in that particular week. If there are no disputes, the smart contract deletes the relevant DFSN from the nodes. Then the deleted transaction data record will be saved in DFSN Id0, which corresponds to the special week id. If there are any disputes regarding any product expiring during the week, the smart contract will not delete the DFSN until the the dispute is resolved. This is required since we are using immutable DFSNs, which do not allow to change the entries. Therefore, it is necessary to delete the entire DFSN at once rather than deleting specific entries. This process is summarized in Algorithm 1. It is important to note that only the registered users are allowed to perform the specific transaction. Moreover, only specific entities are allowed execute the functions in the smart contract, restricting unauthorized entities from performing any task.

Algorithm 1 Storage Recycling

```
Require: $currentWeekId$
  $dfsnName$ ←queryDDST1(getDFSN($currentWeekId$))
  $drfValue$ ←queryDDST2(getDRF($dfsnName$))
  if $drfValue == null$ then
    $result$ ←deleteDFSN($dfsnName$)
  end if
```

The total onchain and offchain storage size for proposed model after X years with DRP set to one year can be calculated as follows.

- The expected data volume in the blockchain after X years, $V_{onchain}$, is given by

$$V_{onchain} = H + N \times iHash \tag{1}$$

- The expected offchain data volume after X years, $V_{offchain}$, is given by

$$V_{offchain} = \sum_{i=1}^{M} T_i \tag{2}$$

The symbols are defined in Table II

TABLE II
THE IMPLICATION OF SYMBOLS IN FORMULA

Symbol	Definition
N	The number of all transactions in the blockchain after X years
M	The number of transactions in the blockchain for previous year
$iHash$	The size of the DFS hash corresponding to each transaction
H	The total data volume of all block header for X years
T_i	The original data volume of the i^{th} transaction

IV. Numerical Results

In this section, we compare the storage growth of the proposed scheme with legacy storage mechanism in blockchain. For both schemes, we compute the offchain storage growth in a single node. The storage volumes are calculated using actual bitcoin transaction statistics, namely, the average block size [17], average number of transactions per block [18], and the average daily transaction count [19], from 2015 January to 2020 December. In here, we consider the bitcoin transaction dates as product expiry dates.

The expected offchain data volume after X years in a conventional system, V_{legacy}, can be obtained as

$$V_{legacy} = \sum_{i=1}^{N} T_i. \tag{3}$$

We assume IPFS is used to implement the DFSN. To calculate the onchain data volume, we multiply the average daily transaction count by the IPFS hash size of 46 bytes. Then we add the block header of 80 bytes by multiplying with the daily average number of blocks, as given in eq. (1). To use the bitcoin data for the proposed system, first, we need to select the DRP for the system. Based on the chosen DRP, the cumulative data volume of the system will vary. We evaluate the storage volume of the proposed scheme when the DRP is set to one year, one month and one day. Furthermore, we assume that all records are free of disputes and are eligible for recycling.

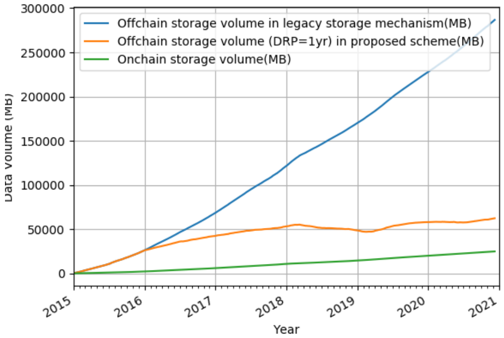

Fig. 3. Data Storage Comparison (MB)

The cumulative data storage comparison between the legacy offchain solution and the proposed solution with DRP set to one year, is shown in Fig. 3. In addition, onchain storage volume is also shown. It is clear that the storage volume grows at a very slow rate with the proposed scheme, while the legacy scheme shows a linear growth with time. This behaviour is observed since the proposed scheme recycle the storage every week, and keeps only the records from the previous year in the system, while the legacy offchain storage solution has accumulated all the records over the years. Therefore, Fig. 3 clearly shows the storage cost savings achievable with our scheme.

Fig. 4 shows the percentage of storage saving of the proposed solution compared to the legacy offchain solution. Since we set DRP to one year, in first year both systems accumulate equal data volume. However, after the first year, proposed solution results in significant storage savings compared to the legacy system. It is important to note that percentage saving increases with time due to the storage recycling operation of the proposed scheme. From the numerical results, one can observe that, after 5 years, the proposed scheme results about 78% of storage space compared to the legacy solution.

Fig. 5 shows the offchain storage volume growth of the proposed solution with different DRPs. The amount of data

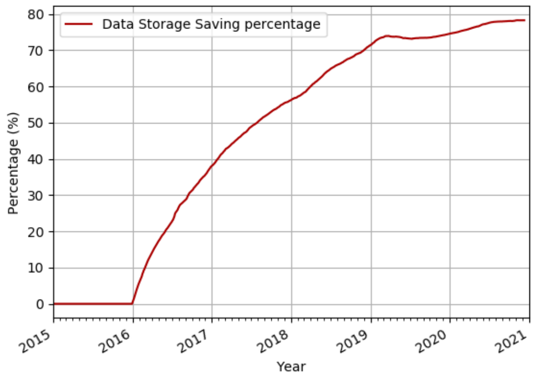

Fig. 4. Reduction of storage volume compared to the legacy scheme.

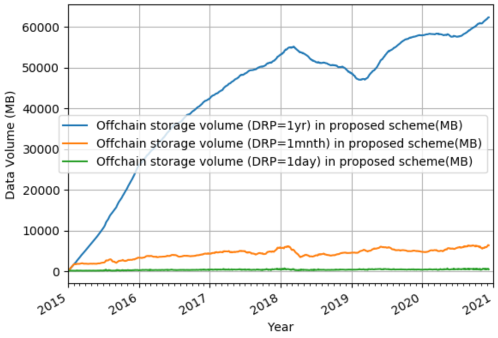

Fig. 5. Offchain storage growth with different DRPs

we keep offchain depends on the DRP. For an example, if we set DRP as one year, system will only keep previous year's records. Similarly, if we choose the DRP as one month, system will only keep the previous month's records. Therefore, shorter DRP leads to a smaller storage volume. However, since we are recycling the storage only after the product expiration date, DRP must be selected based on the life cycle of the products in the SC.

From numerical results, it is evident that the proposed storage recycling scheme yields significant storage requirement reduction and improves the scalability of blockchain enabled IoT equipped FSCs.

V. Experimental Results

Since the proposed scheme uses immutable DFSNs, it is important to show that the data deletion process can be implemented within an acceptable latency. Therefore, to demonstrate the technical feasibility of the proposed scheme, we implement it on Hyperledger blockchain platform.

A. Experimental setup

The computing infrastructure utilized for the implementation consists of a virtual machine (VM) and one host machine. The VM runs Ubuntu 20.04 64-bit with 8GB RAM and four processor cores with 2.40GHz speed. The host machine consists of Intel(R) Core i5 -7200 CPU with dual cores. Fig 6 illustrates the implementation setup. The DFSN is implemented using IPFS, and it is configured and implemented on top of the docker containers.

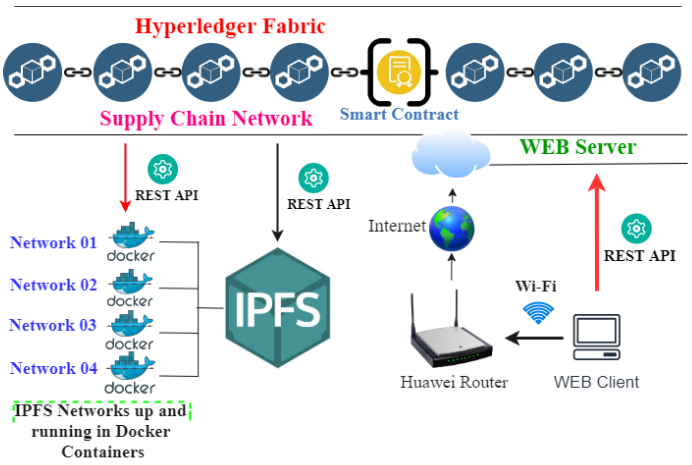

Fig. 6. The implementation setup

The Hyperledger Fabric blockchain platform (HFBP) is connected to the IPFS as the extension of storage. The IPFS is connected to smart contracts via REST API to store the objects in the distributed storage. The HFBP is connected to an external web client via the REST API to submit and retrieve transactions. The smart contracts are deployed to intervene for different steps of the blockchain. Within the implementation process, we test the solution using one blockchain node and two different IPFS networks, which are setup inside the docker containers. The simulation of transaction traffic is performed by the software running on the external web client.

B. Latency Calculation

1) Data Storing Latency: We emulate an FSC environment and evaluate the food item data storage latency through the IPFS network to the blockchain. Here, we set up the IPFS network on top of the docker containers. For latency calculations, we considered the activities of transaction request initiation via the external web client to the smart contract, storing the transaction details in the IPFS network via the REST API call from the smart contract, getting the hash of the transaction data from IPFS, storing the hash in the distributed ledger and the time it takes to mine a block. The latency is calculated for block generation times of 500ms, 1s, 2s, 4s and 8s. For each block generation time, the latency values are obtained by running the experiment for 100 instances, and taking the average latency value with a 95% confidence interval. The results are tabulated in Table III.

The results show that the latency varies with block generation time. However, if we split the time taken for each activity, and ignore the mining delay in each test scenario (assuming the mining delay is identical to the block generation time), the average time for the storing the transaction details in the IPFS network and returning the hash is around 850 milliseconds

TABLE III
LATENCY MEASUREMENTS FOR DIFFERENT BLOCK GENERATION TIMES

Block Generation Time	Data Storage Delay (ms)	Data Storage Recycle Delay (ms)
500ms	1314.22±12.3	4448.83±34.14
1s	1841.02±15.1	5441.87±22.85
2s	2824.75±14.9	7425.07±25.84
4s	4825.24±13	11424.05±28.36
8s	8865.37±14.8	19411.62±27.61

for all block generation times. This confirms that data storing and hash acquisition from the IPFS network which run on top of the docker container can be completed with an acceptable level of latency.

2) Storage Recycling Latency: In the experiment, we emulate the storage recycling process by triggering a deletion event hourly. Here, we consider a single, pre-configured IPFS network. From an automated external application, transaction deletion request is initiated to the smart contract with the IPFS network name. Then the record is stored in the distributed ledger. Once it is stored, the smart contract calls an external REST API to create the new IPFS network and deletes the existing IPFS network. The deleted transaction record is stored in the distributed ledger. Once a block is mined, we calculated the time taken for the entire process to complete as the latency of the food item data storage deletion process. The latency is calculated for block generation times of 500ms, 1s, 2s, 4s and 8s. For each block generation time, the latency values are obtained by running the experiment for 100 instances, and taking the average latency value with a 95% confidence interval. The results are tabulated in Table III.

The results show that the latency varies with block generation time. Similar to the previous experiment, if we ignore the mining delay, the average time it takes for the IPFS network creation and deletion activity is around 3500 milliseconds. This affirms the recycling can be implemented with an acceptable level of latency. To scale-up the implementation setup, it is possible to configure several VMs as blockchain nodes, pre configure several IPFS networks and can follow the same procedure as above in each node. With this approach, it is possible to recycle the offchain storage, maintaining the accuracy and the integrity of the FSC.

VI. CONCLUSION

This paper proposed a scheme to reduce the storage volume growth rate of blockchain enabled IoT equipped FSCs by recycling the offchain storage. This is achieved by periodically deleting the DFSNs related to the food items, which are no longer required to be stored in the blockchain, due to their expiration. The proposed scheme can be implemented using immutable DFSN and DDS to maintain the authenticity and integrity of the transaction data. Numerical results demonstrated that the proposed scheme leads to 78% saving in storage volume after 5 years, which can result in significant cost savings and improves the scalability of blockchain equipped FSCs. Furthermore, the proposed scheme was implemented on Hyperledger blockchain platform to verify the technical feasibility. The experimental results confirmed that the proposed scheme can be implemened with an acceptable level of latency, while maintaining all the necessary features of blockchains.

ACKNOWLEDGMENT

This work has been performed under the framework of 6Genesis Flagship (grant 318927) and 5GEAR projects.

REFERENCES

[1] N. Kawaguchi, “Application of blockchain to supply chain: Flexible blockchain technology,” *Procedia Computer Science*, vol. 164, pp. 143–148, 2019.

[2] A. Manzoor, A. Braeken, S. S. Kanhere, M. Ylianttila, and M. Liyanage, “Proxy re-encryption enabled secure and anonymous iot data sharing platform based on blockchain,” *Journal of Network and Computer Applications*, vol. 176, p. 102917, 2021.

[3] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda, “Blockchain-based traceability in agri-food supply chain management: A practical implementation,” in *2018 IEEE IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany)*, 2018, pp. 1–4.

[4] T. M. Hewa, Y. Hu, M. Liyanage, S. Kanhare, and M. Ylianttila, “Survey on blockchain based smart contracts: Technical aspects and future research,” *IEEE Access*, 2021.

[5] R. A. Mishra, A. Kalla, A. Braeken, and M. Liyanage, “Privacy protected blockchain based architecture and implementation for sharing of students’ credentials,” *Information Processing & Management*, vol. 58, no. 3, p. 102512, 2021.

[6] J.-H. Tseng, Y.-C. Liao, B. Chong, and S.-w. Liao, “Governance on the Drug Supply Chain via gCoin Blockchain,” *International journal of environmental research and public health*, vol. 15, no. 6, p. 1055, 2018.

[7] Y. Madhwal and P. B. Panfilov, “Blockchain and supply chain management: Aircrafts’parts’business case.” *Annals of DAAAM & Proceedings*, vol. 28, 2017.

[8] F. Tian, “An agri-food supply chain traceability system for china based on rfid & blockchain technology,” in *2016 IEEE 13th international conference on service systems and service management (ICSSSM)*, 2016, pp. 1–6.

[9] M. V. Kumar, N. Iyengar *et al.*, “A framework for blockchain technology in rice supply chain management,” *Adv. Sci. Technol. Lett*, vol. 146, pp. 125–130, 2017.

[10] F. Tian, “A supply chain traceability system for food safety based on haccp, blockchain & internet of things,” in *2017 IEEE International conference on service systems and service management*, 2017, pp. 1–6.

[11] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, “Solutions to scalability of blockchain: A survey,” *IEEE Access*, vol. 8, pp. 16 440–16 455, 2020.

[12] H. Dang, T. T. A. Dinh, D. Loghin, E.-C. Chang, Q. Lin, and B. C. Ooi, “Towards scaling blockchain systems via sharding,” in *Proceedings of the 2019 international conference on management of data*, 2019, pp. 123–140.

[13] A. Singh, K. Click, R. M. Parizi, Q. Zhang, A. Dehghantanha, and K.-K. R. Choo, “Sidechain technologies in blockchain networks: An examination and state-of-the-art review,” *Journal of Network and Computer Applications*, vol. 149, p. 102471, 2020.

[14] I. Williams, “Cross-chain blockchain networks, compatibility standards, and interoperability standards: The case of european blockchain services infrastructure,” in *Cross-Industry Use of Blockchain Technology and Opportunities for the Future*. IGI global, 2020, pp. 150–165.

[15] H. Mei, Z. Gao, Z. Guo, M. Zhao, and J. Yang, “Storage mechanism optimization in blockchain system based on residual number system,” *IEEE Access*, vol. 7, pp. 114 539–114 546, 2019.

[16] Q. Zheng, Y. Li, P. Chen, and X. Dong, “An innovative ipfs-based storage model for blockchain,” in *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, 2018, pp. 704–708.

[17] “avg-block-size,” [Online]. Available at https://www.blockchain.com/charts/avg-block-size, Accessed: 10.12.2020.

[18] “n-transactions-per-block,” [Online]. Available at https://www.blockchain.com/charts/n-transactions-per-block, Accessed: 10.12.2020.

[19] “n-transactions-total,” [Online]. Available at https://www.blockchain.com/charts/n-transactions-total, Accessed: 10.12.2020.