Secure and Efficient Image Transmission in IoT Networks Using Hyperledger Besu Blockchain

Burak Ağgül, Gökhan Erdemir, *Senior Member, IEEE*, and Tayfun Acarer

***Abstract*— In recent years, secure and efficient image data transmission in Internet of Things (IoT) systems has become increasingly important, especially in real-time video and image capture applications. This study proposes a novel blockchain-based architecture that enables secure image transmission from IoT devices using Hyperledger Besu and Python-based smart contracts. The proposed system consists of TurtleBot and Raspberry Pi for dynamic and static image capture, utilizes 128 KB chunking for efficient data segmentation, and ensures data integrity through SHA-256 hashing. The Hyperledger Besu network, configured with a QBFT (IBFT 2.0) consensus algorithm, guarantees trusted data logging across four nodes for each device. Experimental studies demonstrate the proposed system's reliability in preserving data integrity, reducing latency, and optimizing resource consumption. Also, performance and security analyses validate the system's applicability for real-time, privacy-sensitive IoT environments.**

I. Introduction

With the rapid proliferation of Internet of Things (IoT) technology in recent years across various domains, the secure transmission and storage of visual data (such as images, video, and live streaming) has become a significant concern. Due to the sensitive nature of this kind of data in terms of privacy and security, higher security standards are required. However, most existing IoT infrastructures are based on centralized and cloud-based systems, which can lead to privacy breaches, security vulnerabilities, and data manipulation [1], [2], [3]. One of the main issues cloud-based IoT systems face is the risk of a single point of failure in centralized architectures.[1], [3]. Moreover, it is not always possible to guarantee the integrity and confidentiality of data in traditional centralized systems. As a solution to these problems, blockchain technology has become a secure alternative in IoT applications due to its decentralized structure, strong cryptographic algorithms, and ability to automate data flow through smart contracts [4], [2], [5]. The distributed ledger structure of blockchain ensures that data is stored securely, verifiably, and immutably. [4], [6]. Recent studies have shown that the use of blockchain in IoT systems is becoming increasingly widespread and offers significant security advantages.[1], [7], [3], [5]. However, this integration also brings challenges such as scalability, performance limitations, energy consumption, and data privacy [3], [5]. In the literature, various studies have focused on data security, privacy, and performance optimization in blockchain-based IoT systems. In particular, the blockchain-based secure image transmission model developed by [8] offers an effective method against data manipulation and attacks by dividing images received from IoT devices into segments and protecting each segment with a unique signature. The model used in Hyperledger-based systems, which includes low energy consumption, smart transaction validation, and data storage strategies, provides a significant solution for enhancing the performance of IoT systems [9]. The zero-trust approach proposed by Liu et al. [1] has achieved successful results with low latency on the Ethereum platform by providing a decentralized, fair, and verifiable information-sharing model in blockchain-based IoT systems. Also, in [4], optimization methods such as parallel execution, off-chain, and cross-chain solutions were examined to improve smart contract performance on different blockchain platforms. Lu et al. [7], focusing on integrating federated learning and blockchain in industrial IoT systems, developed a novel consensus mechanism called Proof of Training Quality (PTQ) to enable privacy-preserving data sharing. Similarly, the attribute-based access control (ABAC) method and searchable encryption algorithms proposed in [2] aim to increase data privacy in IoT devices. An innovative method for secure data transfer using blockchain and Bi-LSTM models in industrial IoT environments was presented in [10]. In [11], a solution was proposed for providing secure and efficient data transmission with the smart contract model called "aNp" developed on the private Hyperledger Besu network. The proposed model offers a practical approach, especially in critical security elements such as secure key management and authentication of IoT devices [12]. On the other hand, in [13], a Hyperledger Fabric-based blockchain model was proposed for data sharing among IoT devices, and performance optimization was achieved regarding energy consumption and data integrity. Finally, a secure digital evidence storage model was developed, integrating IPFS, blockchain, and smart contracts to prevent data manipulation in IoT-based smart environments [14]. The proposed model presents an innovative approach specifically aimed at the verifiable and secure preservation of sensitive data.

This study uses blockchain technology to present a secure and efficient image transfer system for IoT environments. The proposed model utilizes a 4-node local Hyperledger Besu blockchain, integrated with smart contracts and advanced cryptographic methods. The HTTP RPC protocol securely sends images from TurtleBot and Raspberry Pi to the blockchain. Prior to storage, data is verified with hash algorithms and optimized using a chunking method, ensuring integrity and reliable storage of IoT-generated images.

II. System Architecture

Our proposed system, which was developed for the secure and efficient transfer of images in IoT systems, runs on the Hyperledger Besu infrastructure and uses a distributed architecture consisting of 4 nodes. TurtleBot and Raspberry Pi

devices collect IoT data in the proposed system, which is transferred to the blockchain network and stored securely. The QBFT (IBFT 2.0) consensus mechanism of Hyperledger Besu ensures data integrity across the nodes. The HTTP RPC (Remote Procedure Call) protocol is used during data transmission. Data transfer is initiated using the *startData()* function and added to the blockchain in small chunks via the *appendChunkData()* function. Each data chunk is hashed to preserve its integrity, and the SHA-256 algorithm is used to compute the hash value, which is then added to the blockchain. When data is retrieved, the *getChunkData()* function performs hash verification, enabling the detection of any potential data manipulation. This structure is designed to guarantee both security and data integrity.

In the proposed system, TurtleBot uses a ROS-based motion control system to capture environmental images while in motion. Meanwhile, Raspberry Pi collects images from a fixed position, offering energy-efficient, long-term operation. The system's design and data processing workflow are outlined in Algorithm 1 as pseudo-code. This ensures secure image recording on the blockchain with guaranteed data integrity. The proposed approach, which is demonstrated in Algorithm 1, includes several key functions facilitating secure image processing and blockchain integration. The functions *getImageSize()* and *loadImageWithOpenCV()* are used to load the image and check its size. Once loaded, the *splitIntoChunks()* function divides the image into smaller data chunks for efficient transmission. Each chunk's integrity is ensured using the *computeSHA256()* function, which generates a cryptographic hash. The *sendToBlockchain()* function transmits these hashed chunks to the Hyperledger Besu-based blockchain network. To confirm successful and accurate data registration, the *verifyOnBlockchain()* function is employed. Finally, the *isUserAuthorized()* function ensures that only authorized users can access the data, providing a secure access control mechanism.

Images are captured using IoT devices like TurtleBot, Raspberry Pi, or PC cameras, either in real-time or for later use. To transmit large images efficiently, data is split into 128 KB chunks. This allows faster transfer and selective re-sending of lost or corrupted parts. Each chunk's integrity is verified using SHA-256 hash values, ensuring data accuracy and security. To ensure secure data transmission, metadata such as the hash value of each chunk, the file name, and the chunk ID are recorded on the Hyperledger Besu blockchain. This process uses the IBFT 2.0 consensus algorithm, creating a decentralized verification mechanism. Following the recording on the blockchain, the Master Node performs data verification using the hash values stored on-chain. If any chunk is missing or altered, only the affected part is retransmitted, thereby preserving data integrity. A secure access control mechanism is also implemented through Python-based smart contracts, allowing only authorized users to access the data. This structure aims to establish a reliable, integrity-assured, and efficient image transmission system.

A. Hyperledger Besu

Hyperledger Besu, an enterprise-grade Ethereum client, is suitable for various use cases, particularly in blockchain-based peer-to-peer energy trading systems applications, because it can operate in permissioned and permissionless networks [15], [16]. Regarding latency and improved efficiency, implementing Hyperledger Besu's Istanbul Byzantine Fault Tolerance (IBFT) 2.0 consensus algorithm has demonstrated better performance than competing blockchain frameworks [17], [18]. In performance tests, Hyperledger Besu demonstrated significantly higher throughput and five times lower latency than Ethereum and Hyperledger Fabric's RAFT consensus mechanism [18]. Its features which offer better scalability, security, and efficiency than traditional public blockchains, make Hyperledger Besu a desirable choice for enterprise blockchain solutions [19], [17], [9].

B. System Architecture and Node Structure

The proposed architecture consists of four nodes: one node each on the TurtleBot and Raspberry Pi devices and two nodes installed on a PC. These four nodes operate on the Hyperledger Besu platform and utilize the QBFT (IBFT 2.0) consensus mechanism. The TurtleBot and Raspberry Pi devices capture image and video data using the OpenCV library and prepare this data for transmission to the blockchain network. Instead of sending the data directly to the blockchain, it is divided into smaller chunks using the chunking method, and each chunk is stored on the blockchain using the *appendChunkData()* function. This approach ensures more efficient transmission of large datasets. The system is designed to provide secure and uninterrupted visual data transmission, with each node assigned specific responsibilities. These roles are represented in Table 1.

Algorithm 1. Secure Transmission of Image Data via Blockchain

```
Input: image_file (input image)
Output: Securely recorded image chunks on the
blockchain

if getImageSize(image_file) < 128 then
    print "Image size too small"
    return
end if
image ← loadImageWithOpenCV(image_file)
chunks ← splitIntoChunks(image)
for each chunk in chunks do
    hash ← computeSHA256(chunk)
    if not verifyIntegrity(chunk, hash) then
        continue
    end if
    success ← false
    retry_count ← 0
    while retry_count<MAX_RETRIES & !success do
        success ← sendToBlockchain(chunk, hash)
        retry_count ← retry_count + 1
    end while
    if not success then
        print "Chunk transmission failed"
        continue
    end if
    if verifyOnBlockchain(hash) then
        print "Chunk recorded successfully"
    else
        print "Blockchain verification failed"
    end if
end for
if isUserAuthorized(current_user) then
    data ← accessBlockchainData()
    print "Access granted"
else print "Access denied"
end if
```

TABLE I. NODE STRUCTURE AND TASKS

Node Nr.	*Device*	*Tasks*
1	Master Node on PC	Blockchain network management Authorization and access control
2	On PC	Network load balancing Data flow management
3	TurtleBot	Data collection on the move (Image)
4	Raspberry Pi	Data collection from a fixed point (image)

Master Node: The master node manages the blockchain network and ensures data security. It is responsible for core functions such as adding new blocks, handling node communication, and controlling the consensus algorithm. It also performs user authentication and defines which devices are authorized to access the data. This function ensures the effective operation of secure access control systems.

Node 2: This node facilitates the efficient flow of data traffic within the network. A load balancing mechanism helps more stably manage the massive data streams generated by IoT devices. During high traffic conditions, it reduces packet loss, thereby maintaining data integrity.

Node 3 (TurtleBot): The TurtleBot's integrated camera collects visual data and transmits it to the blockchain network. Thanks to its mobility, the TurtleBot can continuously gather environmental data while in motion, and the hash values of this data are stored on the blockchain. This method is especially crucial for ensuring continuous image transmission in dynamic environments.

Node 4 (Raspberry Pi): The Raspberry Pi platform is widely used in IoT applications due to its low cost and energy efficiency. It collects environmental image data through its integrated camera and stores it on the blockchain network. Its low power consumption makes it advantageous for long-term operations, supporting uninterrupted field deployment of IoT devices.

C. PoA (Proof of Authority) Consensus Algorithm

The QBFT (IBFT 2.0) consensus algorithm used in the Hyperledger Besu platform is based on the Proof of Authority (PoA) model. This mechanism enhances the security of the blockchain network by allowing only authorized nodes to perform block production and validation functions.

D. OpenCV (Visual data collection)

TurtleBot and Raspberry Pi capture image and video data using OpenCV and prepare it for blockchain transmission. Instead of direct transfer, data is split into smaller chunks and stored on the blockchain via the *appendChunkData()* function, ensuring efficient handling of large datasets.

E. Remote Procedure Call (RPC) Protocol

The HTTP RPC protocol performs data transmission and query operations to the blockchain. The *run()* function connects to the blockchain network through this protocol and executes data insertion and query tasks. The JSON-RPC protocol enhances data transmission speed while minimizing resource consumption on IoT devices.

F. Retry Algorithm

To enhance reliability in data transmission, the system uses *get.chunk()* and *get.chunk.length()* functions to detect and resend only missing chunks. This reduces network load, optimizes bandwidth use, and improves transmission efficiency. Data integrity is ensured through SHA-256 hash verification, enabling quick tampering detection. Efficient transfer is achieved by resending only incomplete parts, and secure access is enforced via Python-based smart contracts, allowing only authorized users. The immutability and consistency of the data are guaranteed by the IBFT 2.0 consensus algorithm of Hyperledger Besu, making the system a reliable, secure, and scalable data transmission solution.

G. Smart Contracts

Smart contracts operating on blockchain technology offer promising applications for Internet of Things (IoT) systems. These self-executing contracts can automate multi-step processes, facilitate resource sharing, and enable service marketplaces for IoT [20]. By supporting decentralized and verifiable interactions, they enhance security, privacy, and trust in IoT environments [20], [21]. Smart contracts simplify the creation of service markets between IoT devices and can increase trust in process execution and data via consensus mechanisms like Practical Byzantine Fault Tolerance (PBFT) [22]. Recent innovations include embedding contracts directly into blockchain transactions and reducing deployment costs in IoT scenarios [23]. Although challenges such as transaction privacy and asset valuation remain, integrating blockchain and smart contracts with IoT holds strong potential to transform industries and enable novel decentralized applications [20], [24].

III. IMPLEMENTATION

The proposed system is designed to securely and efficiently record data from IoT devices onto the blockchain. The process begins with the system initialization and connection of IoT devices to the local network. At this stage, Node 1 and other nodes are integrated into the Hyperledger Besu platform via the local network, establishing the blockchain connection. Once activated, smart contracts create empty blocks, marking the system as operational. Large image data received from IoT devices is divided into smaller data blocks using the chunking method to minimize data loss and network load. These chunks are verified and recorded on the blockchain. If data is missing or corrupted, only the incomplete segments are resent, reducing processing costs and network traffic. The Master Node performs verification to ensure the accuracy of the recorded data. Data stored on the blockchain is accessible only to authorized users, providing strong data security. The block diagram of the proposed model is shown in Figure 1.

Examining the technical details of the methods implemented to ensure the system's successful operation is very important. In particular, the chunking method used to enhance the efficiency and security of data transmission and the operation of smart contract functions are among the key elements that strengthen the system's overall performance.

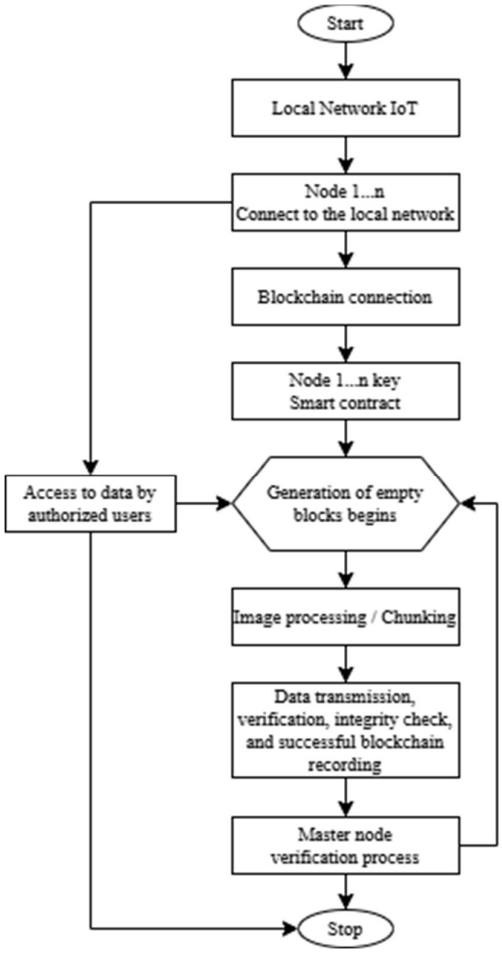

Figure 1. Proposed model's implementation

In this study, a Solidity-based smart contract was developed to securely and efficiently record large-scale IoT data on the blockchain. Data is divided into 128 KB chunks to minimize latency and transmission errors, allowing only missing segments to be resent, reducing network traffic and improving performance. The *startData()* function verifies each chunk using unique hashes to prevent duplication, while *appendChunkData()* adds only missing or updated data, lowering transaction costs. Data management is supported by functions like *getChunkLength()*, *getChunkData()*, *getFileData()*, and *getHashes()*, enabling flexible access and integrity verification. Web3.py in Python connects IoT devices to the blockchain, enhancing system modularity. A test environment (Figure 2) was created using a Lego-based physical city model to simulate smart city scenarios, including traffic flow and pedestrian crossings. This modular setup allows for flexible testing. Within this environment, TurtleBot and Raspberry Pi devices collect image data and store it on the blockchain, demonstrating the system's real-world applicability.

The TurtleBot navigates the Lego roads to gather environmental image data, while the Raspberry Pi is placed in a fixed position, continuously capturing visual information from a designated area. The data collected by both devices is segmented using the chunking method and transmitted to the blockchain, optimizing network traffic and transaction costs during large data transfers. This Lego-based test environment is critical in assessing the system's performance and data security, enabling scaled-down modeling of real traffic scenarios. It provides an effective platform for testing blockchain-based data validation, secure data transmission, and environmental monitoring within smart city applications.

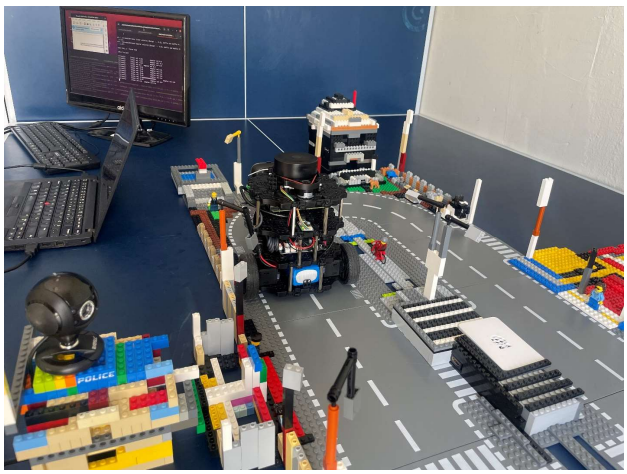

Figure 2. The test environment.

IV. Experimental Studies

In experimental studies, images captured at different time spans from Node 3 and Node 4 were transmitted to Node 1 via the blockchain to verify and analyze the system's performance. The results for Node 3 are presented in Table 2, while the results for Node 4 are provided in Table 3. During this experimental process, the data collection system's memory usage, CPU usage, HDD usage, and processing time parameters were monitored and shown in Tables 2 and 3 for each node. The performance of the proposed system was evaluated based on two key factors: the chunking method and Hyperledger Besu's efficiency. Key metrics included data transfer speed and processing time. The transmission of data from TurtleBot and Raspberry Pi was also analyzed. Results showed that TurtleBot's ROS-based motion did not cause noticeable latency during image transmission, while Raspberry Pi ensured seamless, continuous data transfer thanks to its low energy consumption.

The 128 KB chunking method enabled faster and more efficient transmission of large image files to the blockchain network. Compared to sending large files in a single piece, transfer speed significantly improved, allowing for more seamless data transmission, particularly under the limited bandwidth constraints of IoT devices. Regarding data loss prevention, chunking allowed only the missing parts to be retransmitted in case of a failed transfer, thereby reducing network load and optimizing transmission time. This method also enhanced the energy efficiency of IoT devices, strengthening the system's overall resilience.

TABLE II. EXPERIMENTAL RESULTS FOR NODE 3

Beginning time	Image Shooting Time(s)	Image Size (MB)	Finishing time	Process Duration (s)	CPU Usage (%)	Memory Usage (%)	HDD Usage (MB)
2025-03-19 17:18:56.35	0.71	0.1	2025-03-19 17:20:16.52	80.17	100.0% → 30.6%	57.6% → 59.2%	11621.97 MB → 11623.55 MB
2025-03-19 17:21:13.37	0.7	0.09	2025-03-19 17:21:29.89	16.52	100.0% → 41.5%	59.3% → 59.6%	11623.79 MB → 11623.96 MB
2025-03-19 17:21:57.05	0.7	0.08	2025-03-19 17:22:33.71	36.66	100.0% → 53.5%	59.6% → 60.3%	11624.66 MB → 11687.37 MB
2025-03-19 17:22:56.45	0.71	0.1	2025-03-19 17:23:55.38	58.94	100.0% → 41.4%	60.0% → 66.7%	11687.52 MB → 11689.25 MB
2025-03-19 17:24:16.74	0.71	0.1	2025-03-19 17:25:17.32	60.59	100.0% → 58.6%	67.1% → 68.5%	11689.48 MB → 11691.89 MB
2025-03-19 17:25:37.96	0.71	0.1	2025-03-19 17:25:58.75	20.78	100.0% → 38.6%	68.7% → 69.3%	11692.04 MB → 11692.31 MB
2025-03-19 17:26:27.70	0.71	0.08	2025-03-19 17:26:58.98	31.28	100.0% → 57.8%	69.1% → 69.5%	11692.47 MB → 11693.61 MB
2025-03-19 17:27:20.32	0.71	0.09	2025-03-19 17:27:58.97	38.65	100.0% → 53.5%	69.5% → 69.6%	11693.71 MB → 11695.01 MB
2025-03-19 17:28:20.88	0.71	0.09	2025-03-19 17:29:03.01	42.12	100.0% → 47.0%	69.7% → 70.1%	11695.13 MB → 11696.32 MB
2025-03-19 17:29:28.69	0.71	0.11	2025-03-19 17:30:15.33	46.64	100.0% → 42.1%	70.0% → 70.4%	11696.48 MB → 11698.18 MB

TABLE III. EXPERIMENTAL RESULTS FOR NODE 4

Beginning time	Image Shooting Time(s)	Image Size (MB)	Finishing time	Process Duration (s)	CPU Usage (%)	Memory Usage (%)	HDD Usage (MB)
2025-03-19 17:11:00.13	0.62	0.08	2025-03-19 17:11:16.21	16.08	100.0% → 48.8%	50.4% → 50.7%	12595.83 MB → 12596.05 MB
2025-03-19 17:11:35.44	0.63	0.08	2025-03-19 17:11:47.64	12.21	100.0% → 46.0%	50.6% → 51.0%	12596.25 MB → 12596.37 MB
2025-03-19 17:12:09.65	0.64	0.07	2025-03-19 17:12:20.27	10.62	100.0% → 61.1%	51.0% → 51.2%	12596.57 MB → 12596.68 MB
2025-03-19 17:12:32.58	0.62	0.06	2025-03-19 17:12:51.71	19.13	100.0% → 55.6%	51.2% → 51.5%	12596.81 MB → 12597.96 MB
2025-03-19 17:13:11.68	0.62	0.09	2025-03-19 17:13:19.97	8.29	100.0% → 78.6%	51.5% → 51.8%	12598.23 MB → 12598.48 MB
2025-03-19 17:13:43.75	0.62	0.08	2025-03-19 17:14:20.87	37.12	100.0% → 50.9%	51.7% → 52.0%	12598.66 MB → 12599.99 MB
2025-03-19 17:14:50.00	0.62	0.08	2025-03-19 17:15:24.95	34.94	100.0% → 57.2%	52.0% → 52.3%	12600.21 MB → 12601.52 MB
2025-03-19 17:15:53.06	0.62	0.08	2025-03-19 17:16:06.68	13.62	100.0% → 45.4%	52.5% → 52.8%	12601.70 MB → 12601.82 MB
2025-03-19 17:16:27.37	0.62	0.08	2025-03-19 17:17:04.30	36.93	100.0% → 53.4%	52.6% → 53.5%	12672.41 MB → 12673.62 MB
2025-03-19 17:17:19.56	0.62	0.08	2025-03-19 17:17:50.72	31.17	100.0% → 62.3%	53.1% → 56.9%	12673.67 MB → 12674.75 MB
2025-03-19 17:11:00.13	0.62	0.08	2025-03-19 17:11:16.21	16.08	100.0% → 48.8%	50.4% → 50.7%	12595.83 MB → 12596.05 MB

Hyperledger Besu's IBFT 2.0 consensus algorithm significantly improved the system's performance. Optimizing block validation times enabled near real-time data recording from IoT devices onto the blockchain with minimal latency. Measurements showed that block production times ranged between 0.5 and 2 seconds with high data transfer efficiency. The 4-node architecture enhanced the load-balancing mechanism, accelerating data processing and effectively managing network traffic, especially during high-volume image transmissions. Node 3's CPU usage ranged between 40% and 60%, peaking at 58.6%, remaining within safe limits. However, its processing times fluctuated, with a maximum of 80.17 seconds, indicating potential latency. Memory usage peaked at 71.10%, approaching its limit and suggesting a need for future expansion. Disk usage remained stable between 11.623 MB and 11.699 MB. In contrast, Node 4 showed higher

CPU usage, fluctuating between 50% and 80%, and peaking at 78.6%, which exceeds the 70% threshold and indicates a potential resource risk. Despite this, it delivered more efficient processing, with a maximum duration of 37.12 seconds. Memory usage was more stable, reaching up to 56.90%, while disk usage peaked at 12.674 MB, signaling the need for possible storage optimization. In summary, Node 3 offers more stable resource usage but suffers from variable processing times, requiring CPU and latency improvements. Node 4 performs more efficiently but at the cost of higher resource consumption, suggesting the need for better memory and disk management to enhance system stability and scalability.

V. Conclusion

The proposed system, leveraging Hyperledger Besu, significantly strengthens data security, integrity, and scalability in blockchain-based IoT applications. It employs the IBFT 2.0 consensus algorithm, which validates blocks through a two-thirds majority, and a role-based and permission-based access model further safeguards against unauthorized access. TurtleBot enables dynamic image collection, while Raspberry Pi supports low-power, long-term data capture in fixed locations. The 128 KB chunking method improves efficiency in transferring large images by minimizing bandwidth use and allowing only missing chunks to be resent, crucial for constrained IoT networks.The developed smart contracts simplify blockchain integration. However, enhancements such as asynchronous programming and multithreading are recommended to reduce latency and boost responsiveness. Experimental results confirm the system's robustness and performance advantages over traditional methods. Future improvements will target scalability, energy efficiency, and enhanced security, including quantum-resistant algorithms and faster data processing.

Acknowledgment

The authors acknowledge the support of the University of Tennessee at Chattanooga. The research reported in this publication was supported by the FY2024-25 Center of Excellence for Applied Computational Science competition.

References

[1] A. Liu et al., “ORE Open Research Exeter TITLE A Blockchain-based Decentralized, Fair and Authenticated Information Sharing Scheme in Zero Trust Internet-of-Things A NOTE ON VERSIONS A Blockchain-based Decentralized, Fair and Authenticated Information Sharing Scheme in Zero Trust Internet-of-Things,” 2022. [Online]. Available: http://hdl.handle.net/10871/129781

[2] B. Hu, Y. Chen, H. Yu, L. Meng, and Z. Duan, “Blockchain-Enabled Data-Sharing Scheme for Consumer IoT Applications,” IEEE Consumer Electronics Magazine, vol. 11, no. 2, pp. 77–87, Mar. 2022, doi: 10.1109/MCE.2021.3066793.

[3] N. K. Tran, M. Ali Babar, and J. Boan, “Integrating blockchain and Internet of Things systems: A systematic review on objectives and designs,” Jan. 01, 2021, Academic Press. doi: 10.1016/j.jnca.2020.102844.

[4] Y. Liu et al., “An overview of blockchain smart contract execution mechanism,” Sep. 01, 2024, Elsevier B.V. doi: 10.1016/j.jii.2024.100674.

[5] D. Commey, B. Mai, S. G. Hounsinou, and G. V. Crosby, “Securing Blockchain-Based IoT Systems: A Review,” IEEE Access, vol. 12, pp. 98856–98881, 2024, doi: 10.1109/ACCESS.2024.3428490.

[6] P. Martina, R. Dhanvardini, R. Vijay, R. Amirtharajan, and P. Pravinkumar, “Design development and execution of Smart Contract : An Overview,” in 2023 International Conference on Computer Communication and Informatics (ICCCI), IEEE, Jan. 2023, pp. 1–5. doi: 10.1109/ICCCI56745.2023.10128536.

[7] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, “Blockchain and Federated Learning for Privacy-Preserved Data Sharing in Industrial IoT,” IEEE Trans Industr Inform, vol. 16, no. 6, pp. 4177–4186, Jun. 2020, doi: 10.1109/TII.2019.2942190.

[8] Y. Li, Y. Tu, J. Lu, and Y. Wang, “A Security Transmission and Storage Solution about Sensing Image for Blockchain in the Internet of Things,” Sensors, vol. 20, no. 3, p. 916, Feb. 2020, doi: 10.3390/s20030916.

[9] Z. Leng, K. Wang, Y. Zheng, X. Yin, and T. Ding, “Hyperledger for IoT: A Review of Reconstruction Diagrams Perspective,” Electronics (Basel), vol. 11, no. 14, p. 2200, Jul. 2022, doi: 10.3390/electronics11142200.

[10] T. Skaria, A. M. A. Bamini, and R. Chitra, “Secure Data Transmission in IIoT Using Blockchain Based BI-LSTM,” in 2024 International Conference on Advancement in Renewable Energy and Intelligent Systems (AREIS), IEEE, Dec. 2024, pp. 1–6. doi: 10.1109/AREIS62559.2024.10893684.

[11] Md. R. H. Shahrukh, Md. T. Rahman, and N. Mansoor, “Exploration of Hyperledger Besu in Designing Private Blockchain-based Financial Distribution Systems,” Nov. 2023, [Online]. Available: http://arxiv.org/abs/2311.08483

[12] S. Mercan, M. Cebe, R. S. Aygun, K. Akkaya, E. Toussaint, and D. Danko, “Blockchain-based video forensics and integrity verification framework for wireless Internet-of-Things devices,” SECURITY AND PRIVACY, vol. 4, no. 2, Mar. 2021, doi: 10.1002/spy2.143.

[13] H. H. Pajooh, M. Rashid, F. Alam, and S. Demidenko, “Hyperledger fabric blockchain for securing the edge internet of things,” Sensors (Switzerland), vol. 21, no. 2, pp. 1–29, Jan. 2021, doi: 10.3390/s21020359.

[14] D. Rani et al., “A secure digital evidence preservation system for an iot-enabled smart environment using ipfs, blockchain, and smart contracts,” Peer Peer Netw Appl, vol. 18, no. 2, p. 5, Apr. 2025, doi: 10.1007/s12083-024-01855-z.

[15] H. M. Kim, H. Turesson, M. Laskowski, and A. F. Bahreini, “Permissionless and Permissioned, Technology-Focused and Business Needs-Driven: Understanding the Hybrid Opportunity in Blockchain Through a Case Study of Insolar,” IEEE Trans Eng Manag, vol. 69, no. 3, pp. 776–791, Jun. 2022, doi: 10.1109/TEM.2020.3003565.

[16] “https://besu.hyperledger.org/.”

[17] J. Abdella, Z. Tari, A. Anwar, A. Mahmood, and F. Han, “An Architecture and Performance Evaluation of Blockchain-Based Peer-to-Peer Energy Trading,” IEEE Trans Smart Grid, vol. 12, no. 4, pp. 3364–3378, Jul. 2021, doi: 10.1109/TSG.2021.3056147.

[18] N. R. Pradhan, A. P. Singh, N. Kumar, M. M. Hassan, and D. S. Roy, “A Flexible Permission Ascription (FPA)-Based Blockchain Framework for Peer-to-Peer Energy Trading With Performance Evaluation,” IEEE Trans Industr Inform, vol. 18, no. 4, pp. 2465–2475, Apr. 2022, doi: 10.1109/TII.2021.3096832.

[19] V. Capocasale, D. Gotta, and G. Perboli, “Comparative analysis of permissioned blockchain frameworks for industrial applications,” Blockchain: Research and Applications, vol. 4, no. 1, p. 100113, Mar. 2023, doi: 10.1016/j.bcra.2022.100113.

[20] K. Christidis and M. Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things,” IEEE Access, vol. 4, pp. 2292–2303, 2016, doi: 10.1109/ACCESS.2016.2566339.

[21] M. Hassan, C. Jincai, A. Iftekhar, and X. Cui, “Future of the Internet of Things Emerging with Blockchain and Smart Contracts,” International Journal of Advanced Computer Science and Applications, vol. 11, no. 6, 2020, doi: 10.14569/IJACSA.2020.0110676.

[22] L. Da Xu and W. Viriyasitavat, “Application of Blockchain in Collaborative Internet-of-Things Services,” IEEE Trans Comput Soc Syst, vol. 6, no. 6, pp. 1295–1305, Dec. 2019, doi: 10.1109/TCSS.2019.2913165.

[23] H. Su, B. Guo, Y. Shen, and X. Suo, “Embedding Smart Contract in Blockchain Transactions to Improve Flexibility for the IoT,” IEEE Internet Things J, vol. 9, no. 19, pp. 19073–19085, Oct. 2022, doi: 10.1109/JIOT.2022.3163582.

[24] K. Peng, M. Li, H. Huang, C. Wang, S. Wan, and K.-K. R. Choo, “Security Challenges and Opportunities for Smart Contracts in Internet of Things: A Survey,” IEEE Internet Things J, vol. 8, no. 15, pp. 12004–12020, Aug. 2021, doi: 10.1109/JIOT.2021.3074544.