IEEE ICBC'24 1570978835

BIoTFlex: A Scalable Architecture for Multi-Application Supporting BIoT Environments with Application Shifting at Runtime

1st Author 2nd Author 3rd Author

***Abstract*—While existing approaches that integrate IoT and blockchain technologies have demonstrated efficiency, they often exhibit limitations by focusing on a singular application scenario. These proposals require specific hardware configurations, making them less adaptable when transitioning to new scenarios. In many cases, this transition necessitates either a fresh hardware setup or manual adjustments to the existing hardware. In response, this paper introduces an innovative architecture for Blockchain-based Internet of Things (BIoT) systems, providing the flexibility to accommodate multiple application scenarios within a unified hardware infrastructure. Our methodology utilizes smart contracts for system orchestration, offering a agile and cost-effective alternative to conventional hardware-level configurations. The proposed architecture is designed mainly for IoT environments, ensuring scalability without the need for centralized authorization or causing network centralization, thus related vulnerabilities. Additionally, it facilitates real-time application scenario shifts through smart contracts, providing an economical solution for managing existing deployments. The obtained results demonstrate the proposal's capability to scale effectively in network size and handle varying numbers of requests within each identified scenario type. Additionally, the results indicate the feasibility of transitioning between scenarios through smart contracts.**

***Index Terms*—BIoT, Blockchain, IoT, Smart Contract**

I. Introduction

Blockchain is a technology that combines peer-to-peer (P2P) network [1], distributed ledger technology (DLT) and cryptology. The primary goal of any blockchain-based system is to maintain a live decentralized transaction ledger while defending against malicious actors who may attempt to manipulate the system [2].

After it is first deployment for Bitcoin in [3], blockchain has been utilised in various fields. One of the major advancements in the area is smart contracts invented by Ethereum blockchain [4]. Smart contracts are code files [5], [6] stored on blockchain [6], [7] and aim four objectives [4], i.e. observability, verifiability, privity and enforceability.

Internet of Things (IoT), on the other hand, is a groundbreaking technology that serves to transform current cities into smart cities [4] upon adding intelligence to everyday devices. It utilises various technologies such as Wireless Sensors Networks (WSN) and Radio Frequency Identification (RFID) [4], to communicate through the Internet for transmitting sensed data.

IoT devices tend to contain limited hardware capacities in processing power, storage and etc. Thus, they have various vulnerabilities, especially considering that most solutions consider building trust between IoT devices and some certain entities in the network. The major challenges IoT devices and environments face include security, privacy, encryption and performance [6].

Many researchers in the IoT field have addressed these challenges and various proposals suggest blockchain amalgamation with IoT environments as a solution [1], [8], [9]. In [9], [10], authors suggest that blockchain is indeed can be utilised in most modern IoT application areas demonstrating examples in the literature. Additionally, in [9], the blockchain is considered as a crucial component of IoT system especially when the IoT devices are assumed to trust third parties.

Though blockchain-based IoT (BIoT), has studied by many researchers, the BIoT has its own challenges such as network scalability and throughput. Researchers have been addressing these issues in different application areas. Though the proposals are effective in curing the major challenges, they are limited to the specific hardware deployment with specific configuration. This situation prevents a unifying BIoT environment that can be applied to multiple BIoT applications.

As per our foundation work [11], we explored various architecture proposals and application papers in the field of BIoT. Then, we noticed that the papers suggested various hardware setups. Considering that a BIoT environment may require switching another application, it would be infeasible to manually adjust the hardware and update their configuration file. Thus, we first catergorized the applications into three scenarios based on IoT device interaction with blockchain, and proposed a scalable architecture for BIoT environments which supports all three scenarios and capable of transitioning between them. In this paper we deployed the conceptual architecture defined in our foundation work [11], and evaluate it from latency, network scalability, cluster scalability and transition process time cost. To the best of our knowledge, the proposed BIoTFlex, designed to support various application types and facilitate seamless transitions between them, has not been previously studied in the literature. Our contributions in this work include:

- Introducing a BIoT architecture capable of performing diverse BIoT application scenarios.
- Proposal of a transition process and demonstrating how to facilitate transitions between scenarios through the

implementation of smart contract embedded device configuration (policy) files.
- Proving that upon adopting clustered approach for such BIoT systems, scalability in network size, can be achieved efficiently achieved without relying on a central authority or causing centralisation.
- Classification of BIoT applications based on IoT device interaction with blockchain.

The rest of this paper unfolds as follows: Section II explores related work, providing an in-depth analysis of the scale, advantages, and potential vulnerabilities of the proposed systems. Section III introduces BIoTFlex, the proposed architecture, explaining the process flows in each scenario with applicable scenarios and the transition processes between identified scenario types. This section also outlines the assumptions, functional and non-functional requirements, along with their justifications. Section IV details the implementation of the architecture, encompassing the utilized hardware and the processes applied at each step. Section V presents the obtained results and provides discussions. Section VI concludes the paper, offering insights into future avenues for research.

II. Previous Work

Though in [9] is defined as a distributed append-only public ledger technology, was initially created for cryptocurrencies, today, blockchain has various applications areas. In [4], authors claim that the blockchain with smart contracts can be applied to certain application areas such as energy, insurance, mortgage, music royalty payments, real estate, gambling and betting. However, in [12]–[14] these areas are extended further with BIoT-specific and general application areas.

In Table I, with utilising the abovementioned applications areas, blockchain applications are classified and assigned to the identified scenario types. Additionally, the reviewed papers in the area are cited next to their application area, and related scenario types are indicated.

Under BIoT main area, Trusted IoT Network Decentralisation represents the works those cover blockchain utilisation to mitigate trust that IoT devices are assumed to have to certain centralised entities for various operations. Upon blockchain trustless operation can be achieved. These type of work can be assigned to Scenario-1, if the suggestion is re-routing the IoT data through multiple various entities in the network, and Scenario-3 if the aim is enabling IoT devices diretly connect to the blockchain and perform block appending processes.

BIoT Architecture, studied in [15] and [16], represents new architecture proposals for BIoT environments to solve existing challenges such as network scalability and low throughput, and presents system optimisations upon collecting data and providing AI-based solutions, assigned to Scenario-1, and as well as consensus-level solutions, assigned to Scenario-3.

BIoT automation encapsulates the solutions for enhanced BIoT system management mostly upon utilising smart contracts. The solutions analyses the IoT data stored on blockchain and enhances automation depending on the outcome of the analysis. Thus, this application area is assigned to Scenario-1.

BIoT For Industrial IoT (IIoT) covers the application specifically designed for IIoT environments aiming to increase the efficiency of the system upon system optimisations for more efficient production and reduced cost. Therefore, the applications monitor the IIoT systems and provide optimisation solutions, assigned to Scenario-1. Additionally, consensus level optimisations can be conducted through more lightweight blockchain consensus processes, assigned to Scenario-3.

Edge And Fog Computing applications as studied in [16] considers system optimisations, Scenario-1, and enabling blockchain on edge devices using edge and fog computing, Secnario-3.

Internet Of Energy applications, exemplified by the work presented in [17], comprises smart grids, energy management systems, the integration of renewable energy into established sources, and the comprehensive transition to renewable energies. It aims to solve existing challenges in energy sector by integrating various technologies including blockchain, interconnected devices, specifically IoT under this application area, and optimizations driven by data analytics.

Internet Of Healthcare area contains works that involves collecting individuals' health data and then improving and/or assisting with their health upon personalised suggestions and contacting emergency services if needed. This application area is assigned to Scenario-1 because data collecting from IoT devices is the crucial phase.

BIoT Optimisation, as studied in [15] and [18], encompasses task offloading, distribution and assignment, energy consumption reductions, network throughput enhancements, network and throughput scalability enhancements, and etc. The aim is to leverage the quality of existing BIoT environments by enhancing system responsiveness and reducing management complexity.

Micropayment applications moves micropayments to IoT devices by enabling each IoT device to transact the amount of consumed service independently aiming to provide cost efficiency, granular service usage, enhanced autonomy and etc. Thus, this application area is assigned to Scenario-2.

BIoT Mining researches the optimal solutions for IoT devices to contribute block mining processes upon efficient and effective task distribution, offloading and novel approaches in consensus algorithms including new data types to block verification approaches. Therefore, this area is solely assigned to Scenario-3.

Supply Chain Management, as studied in [19] and [20], automates the supply chains upon smart contracts utilisation. The IoT devices sends data to the system and transact a certain amount defined in related smart contract when certain conditions are met. Therefore, this application area is considered as combination of Scenario-1 and Scenario-2.

The rest of the application areas do not require IoT devices in the deployed system. However, for further justification of the general adoption of the system, we inserted these main areas with their sub-areas to show that the proposed

architecture is capable of running these applications without utilising IoT devices. In these cases, the clusters can be considered as an elemental entity in a larger system such as a smart house in a smart city, a certain department or section in a company, etc. Cluster managers can be the main server or main gateway whereas the main chain is a global chain in the environment. Depending on the application area, certain combinations of the proposed scenario types can be adapted to deploy the applications.

III. Proposed Architecture

During the planning phase of the proposed architecture, we mainly considered the functional and non-functional requirements, analysed BIoT-related proposal and application papers and the suggestions in the related survey papers.

In [6], authors defined a list of requirements from BIoT for efficient and feasible deployments, as follows.

- BIoT systems should support large number IoT devices and consider them to leave and join the network any time.
- Blockchain should resist to malicious nodes and existing attacks in IoT networks.
- Limited capabilities of IoT devices should be considered and should not cause any vulnerabilities.
- BIoT systems should process with high performance.
- BIoT systems should ensure user privacy.

In addition to the abovementioned requirements list, we defined both functional (FRs) and non-functional (NFRs) requirements to establish the framework's capabilities and performance expectations.

The architecture must exhibit scalability, adapting to varying workloads and system demands. It should prioritize the preservation of IoT-generated data privacy, adhering to established standards for data protection. Furthermore, user data privacy is a fundamental consideration, necessitating robust mechanisms for safeguarding sensitive information and compliance with relevant privacy regulations. To ensure smooth and efficient operation, the architecture must identify and address potential bottlenecks within the system, such as high process demands at any time and latency caused by large network sizes. Additionally, support for external auditing and verification processes is integral, allowing independent entities to assess the system's integrity and conformity to established standards.

A key non-functional requirement is the architecture's compatibility with existing systems and devices. It should seamlessly integrate with and utilize current technologies, promoting interoperability and minimizing disruption during implementation. These defined requirements collectively shape the criteria for our proposed architecture, setting standards for effectiveness, security, scalability, and compatibility with existing technologies.

During the feasibility analysis phase of the proposal, to satisfy the abovementioned requirements we made certain decisions as follows.

The decision to dismiss the standardisation of direct connection of IoT devices to the main blockchain is motivated by the non-functional requirement of efficiently utilizing existing IoT devices. By not forcing direct connections, the architecture enhances security and privacy, mitigating vulnerabilities associated especially with highly constrained-IoT devices as they do not hold the minimum amount of capacity to securely process blockchain operations. Indirect connections through a sufficiently capable cluster manager offer a more secure and private solution.

Similarly, the dismissal of multiple main blockchains deployment is based on the aim to mitigate potential system bottlenecks. The complexity introduced by deploying multiple blockchains and the potential need for additional transition layers could lead to inefficiencies.

In contrast, opting for a single main blockchain, supported by promising Layer-1 and Layer-2 scalability solutions ensures optimal performance. Solutions such as Ethereum's Fantom, plasma, rollups, sharding, and child chains offer proven scalability benefits as mentioned in [25]. By leveraging these solutions, the proposed architecture may be capable of addressing scalability requirements while benefiting from advancements and optimisations provided by established blockchain platforms.

These design choices are meticulously considered to meet the defined functional and non-functional requirements. They aim to ensure the efficiency, scalability and security of the proposed architecture within the dynamic and diverse landscape of IoT environments.

In Figure 1, the proposed architecture and the process flow in each scenario type are illustrated. In the rest of this section, each scenario type is explained and example application areas are provided.

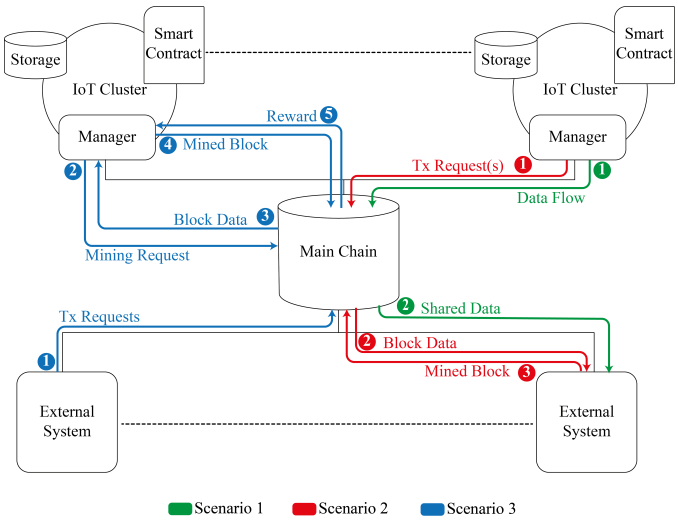

Fig. 1. Use-case scenarios on the proposed architecture (Taken from our foundation work [11])

A. Examples of Adaptable Scenarios and Applications

In our foundation work [11], we explained in detail that piece BIoT applications can be classified into three main scenarios based on IoT device interaction with blockchain.

TABLE I
BLOCKCHAIN APPLICATIONS APPLICABILITY TO IDENTIFIED SCENARIO TYPES

Main Area	Sub-Area	Scenario-1	Scenario-2	Scenario-3
BIoT	Trusted IoT Network Decentralisation	✓		✓
	BIoT Architecture [15], [16]	✓		✓
	BIoT Automation	✓		
	BIoT For Industrial IoT	✓		✓
	Edge And Fog Computing [16]	✓		✓
	Internet Of Energy [17]	✓		
	Internet Of Healthcare	✓		
	BIoT Optimisation [15], [18]	✓	✓	✓
	Micropayment		✓	
	BIoT Mining			✓
	Supply Chain Management [19], [20]	✓	✓	
Security and Privacy	Intrusion Detection	✓		
	Access Management [21]	✓		
	Cyber Security [22],	✓		✓
Blockchain and AI	Forecasting	✓		
	Marketing	✓		
	System Optimisation	✓		
	Data Analysis [23]	✓		
Lightweight Blockchain	MicroBlockchain	✓		✓
	IoT [21],	✓		✓
Storage	Collaborative Video Delivery	✓		
	Streaming Services (Music and Video)	✓		
	File Storage	✓		
BaaS and DApps	Intelligent Transportation	✓	✓	
	Banking And Payments	✓	✓	
	Energy Management [17]	✓	✓	
	Healthcare	✓	✓	
	Supply Chain Management [19], [20]	✓	✓	
	Government Systems	✓	✓	
	E-Voting	✓	✓	
	Charity	✓	✓	
	Insurance	✓	✓	
Blockchain Network	Secure Communication	✓		✓
	Software Defined Networks (SDN)	✓		✓
	Network Function Virtualization (NFV)	✓		✓
	5G And B5G	✓		✓
	6G	✓		✓
	Network Decentralisaition [24]	✓		✓

1) Scenario 1: In Scenario-1, shown with a green marker in Figure 1, IoT devices in each cluster generate data according to the configuration file to be saved to the main chain. A local storage is defined for each cluster and the generated IoT data is saved to the local storage initially. This local storage can be the cluster manager's storage, cluster-specific data server or a child chain of the main chain. Then the cluster manager of each cluster saves it to the blockchain. However, as explained in Section III-B, if an IoT device is capable of process related blockchain operations, it can directly send data to the main chain.

Scenario-1 type of applications include electronic voting (E-Voting), supply chains, smart grids, anomaly detection, marketing and etc.

2) Scenario 2: In this scenario, clusters utilise resources like electricity, water and a certain service, with agreed conditions on smart contracts. Payments occur after resource consumption and transaction records are stored on the main chain. External systems handle block creation using a preferred consensus mechanism, broadcasting the created block.

Scenario-2 type of applications include payment phases of supply chains, smart grids, marketing and etc.

3) Scenario 3: Scenario 3 involves harnessing the combined processing power of IoT devices for reaching consensus and block forming. Depending on the preferred consensus approach and its specific hardware capacity requirement, capability degrees can be assigned to devices within clusters, considering factors like hashing power, storage space and etc. Two highly popular consensus approaches, Proof of Work (PoW) for the competition-based consensus and Proof of Stake (PoS) for selection-based consensus, are explored.

For PoW-based consensus, where high processing power is crucial, the cluster manager divides the block creation task based on the capability degree of each IoT device. If a device lacks the required hashing power, it is excluded. Concurrently, devices perform the hashing process, and upon successful completion, the mined block is broadcasted, signaling the end of the process.

In contrast, for PoS-based consensus, miners are ranked based on metrics like stake amount or reputation. A smart contract ranks clusters, and one is selected as a validator for block creation. The process involves initiating pre-block creation tasks assigned to IoT devices based on their capability degree. After successful block creation, the manager

broadcasts the block, collects the reward, and other clusters halt their operations.

B. Scenario Transition

The proposed architecture is capable of transitioning from one scenario to another as well as from one application to another upon smart contract embedded configuration files. Though this capability grants system control and management to the software level, the transition process requires certain features to be carefully designed as the process should be completed across the network smoothly.

Prior to formulating the scenario transition process, an comprehensive analysis is conducted to examine existing solutions in the literature. While there are not many proposals specifically addressing transition processes for BIoT environments, [26] discusses transitions and considers various perspectives, including security, data completeness, and etc.

As illustrated in Algorithm 1, the transition process is outlined in three primary steps. In the first step, following the migration of a smart contract-embedded configuration, the cluster managers read the updated configuration file from the blockchain, and IoT devices receive the updated configuration from their respective cluster managers. Then, in the second step, the IoT devices commence the configuration update, and upon successful completion, they proceed to the third step, where they notify their cluster managers of the successful outcome.

In the event of an unsuccessful configuration update, the IoT device checks the transition period threshold. If the period has not yet elapsed, the device recommences the updating process. Conversely, if the threshold is exceeded, the device notifies the cluster manager of the unsuccessful result. Moreover, upon reaching the threshold, all devices involved in the update process cease their operations and collectively notify the cluster manager of the unsuccessful outcome.

IV. Implementation

The simulation environment setup consists of Docker containers each consisting of IoT devices, in our case we defined seven IoT devices for deployment convenience, and the Ethereum blockchain as the main chain. IoT devices are created in ContikiNG Cooja simulator, illustrated in Figure 2.

To test the transaction and network size scalability features, we created simulations for both Scensario-1 and Scenario-2 in iterations with various transaction numbers for each network size. To be exact the the network size is tested for 2, 4, 8, 12 and 16 clusters each with 100, 200, 400, 800, 1.600, 3.200, 6.400 and 12,800 requests.

The test phase is conducted on a DELL OptiPlex 5000 which contains 32GB RAM and Intel i7-12700 with 20 cores.

In the setup of the simulation environment, three distinct smart contracts are defined, each corresponding to implemented scenarios and the transitioning process. The subsequent part of this section provides a detailed explanation for Scenario-1, Scenario-2 and transition process along with its specific smart contract.

Algorithm 1: Scenario Transition Process Algorithm

```
// Step 1: New Smart Contract Embedded Configuration is Requested
Notify IoT devices through cluster managers with the new configuration;
// Step 2: IoT Device Configuration Update
foreach IoT device in the cluster do
    StartConfigurationUpdatedevice, NewConfiguration;
// Step 3: Update Process Result Notification
foreach IoT device in the cluster do
    if Successful Update then
        NotifyClusterManagerdevice, Success;
    else
        if Transition Threshold not achieved then
            return to Step 2;
        else
            NotifyClusterManagerdevice, Unsuccessful;
```

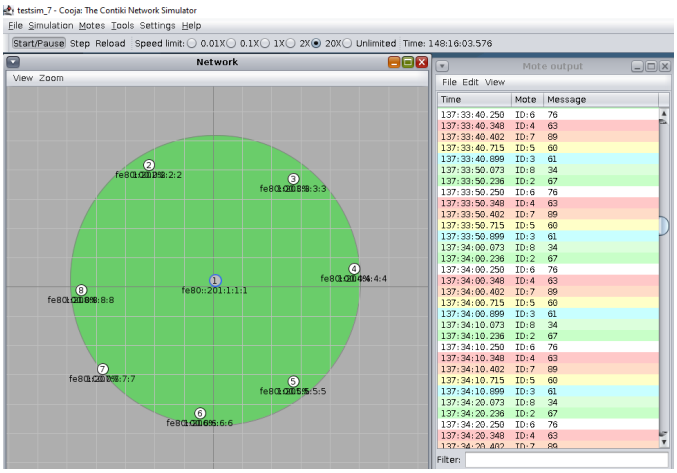

Fig. 2. Cluster setup in Contiki-NG

A. Implementation of Scenario-1

To prove that our proposal is capable of running Scenario-1 type of applications, we simulated a marketing application example. In the simulation, we assume that each cluster represents a house each containing seven IoT devices.

In each iteration, each IoT device sends numerical data through its manager to the blockchain from which the company fetches the data, then, processes it and improves the quality of service for enhanced customized service experience.

DataStorage smart contract is defined to automate and facilitate the transmission of data from IoT clusters to the main blockchain. In each iteration, IoT devices generate data,

random numerical data in our specific case, and store it to the cluster manager's local storage. Subsequently, the cluster manager reads this data and invokes the smart contract to store the generated data on the main blockchain.

B. Implementation of Scenario-2

In Scenario-2, a grid-system simulation is adopted. We assume that each IoT device consumes random generated amount of energy in each process. Then they request a transaction through their own manager to the imaginary energy supplier which is, for easy deployment, a dummy account deployed on the main chain.

The Transaction smart contract is created to automate transactions. While the cluster manager is proficient in conducting the necessary transactions at the end of each iteration, the defined smart contract enhances automation and ensures trustless operation. In every iteration, IoT devices consume a specific amount of energy, generated randomly in our case, and calculate the payment. Then the payment is requested to be transacted by the cluster manager. Subsequently, the cluster manager aggregates all payment amounts and invokes the smart contract to conduct the transaction for the total amount.

C. Implementation of Scenario-3

In Scenario-3, IoT devices pose a pivotal role in consensus processes, requiring the deployment of IoT-specific consensus algorithms tailored to the constrained capacities of these devices. This involves efficient task distribution and assignment within the limited capabilities of IoT devices. Given the focus of our work, we reference existing attempts in this direction instead of developing a new consensus algorithm or optimizing existing ones acknowledging that creating or optimizing consensus algorithms would extend beyond the intended boundaries of this study.

In [27]–[31], researchers proposes approaches to achieve consensus and blockchain block appending solely through IoT devices employing lightweight consensus algorithms, utilizing non-linear data types for block formation, and exploring diverse approaches for efficient consensus task assignment and distribution.

D. Scenario Transition Process

As explained in Section III-B, we proposed a three-step transition process. To simulate the process, a smart contract named Transition is created with two primary functionalities. The first function is *PushUpdatedPolicy()* permits authorised users to update IoT device configuration files. Authorised users initiate change requests using their private keys, and the request is processed only if their identity is recognised as authorized. The second function is *PullUpdatedPolicy()* allows any node in the network to access and read the updated configuration file.

During the simulation, clusters start processing with Scenario-1, and a transition to Scenario-2 is requested immediately after the corresponding configuration file is pushed to the blockchain. Once a new configuration file is pushed, cluster managers invoke the *PullUpdatedPolicy* function to store the updated configuration file. The motivation for cluster managers to pull the new configuration file lies in satisfying to the defined FRs and NFRs and optimising the system by reducing total requests, thereby enhancing efficiency and decreasing communication ratios. After a successful pull process, cluster managers notify the system of the time cost incurred during the process. This notification is solely for the purpose of evaluating the time complexity of each step.

Subsequently, IoT devices initiate a configuration file update process. After conducting the update, IoT devices notify their managers about the time complexity of the update process and the outcome (successful or unsuccessful). Here, we also measured the time complexity of the notification-back process.

V. Results and Discussion

In this section, we present the outcomes derived from the conducted simulations. The results collectively indicate the scalability of the proposed approach, revealing no significant rise in latency concerning both a single request and the total volume of requests with an increase in network size, total number of requests, and IoT devices within a single cluster. Notably, throughout the simulations, neither system unavailability nor bottlenecks were observed.

Figures 3 and 4 illustrate the latency results for Scenario-1 and Scenario-2 across various network sizes, each with different request volumes. For a smaller number of requests, the time cost is consistently similar across all network sizes, approximately 2 seconds for 100 requests, 4 seconds for 200 requests, 7 seconds for 400 requests, and 15 seconds for 800 requests. However, as the number of requests increases to 1,600 and beyond, distinctions in the results become noticeable. Nonetheless, even in the scenarios with the largest network size and highest volume of requests, the observed differences remain under 50 seconds for Scenario-1 and 100 seconds for Scenario-2.

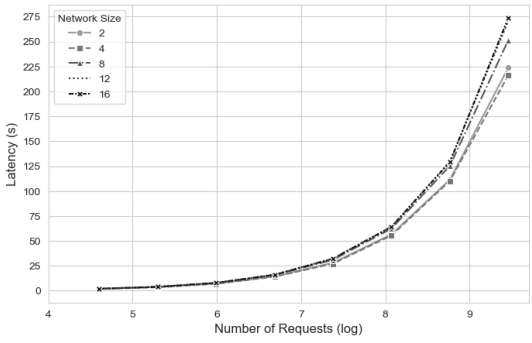

Fig. 3. Scenario-1 total time cost of requests in various network sizes

In Figure 5, change in the latency results for different network sizes, each considering the mean latency value for varying numbers of requests in scalability tests, are depicted for both Scenario-1 and Scenario-2. The results reveal that the trend of latency per request demonstrates a predominantly

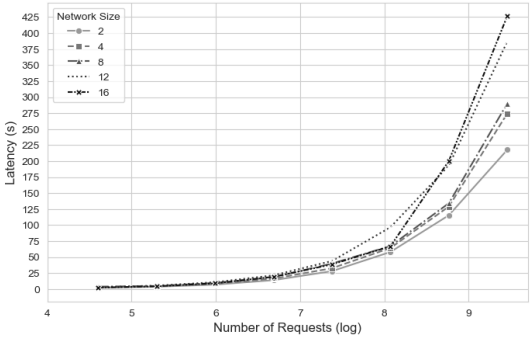

Fig. 4. Scenario-2 total time cost of requests in various network sizes

positive linear correlation with an overall larger scope for Scenario-2. However, the latency remain at the millisecond level as the network size changes, specifically, 5ms being the largest observed difference. Consequently, the figure illustrates that the proposed system exhibits scalability as increases in network size abstain from resulting in an significant rise in the time cost per request.

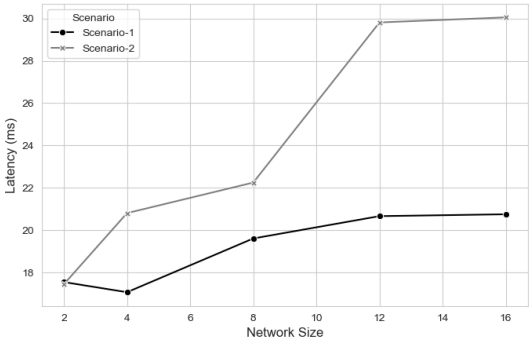

Fig. 5. Scenario-1 and 2 latency values over various network sizes

Figures 3, 4, and 5 collectively demonstrate the scalability of the proposed solution, as none shows a throughput bottleneck with increasing total number of requests and network size. Moreover, for an equivalent number of requests across different network sizes, the observed differences are notably lower than a 1:1 proportion. This finding underscores that an increase in both network size and the total number of requests does not result in a substantial rise in latency and throughput. Consequently, the proposed architecture scales effectively with network size, avoiding the need for central authorization and preventing centralization within the network.

TABLE II
TIME COST OF SCENARIO TRANSITIONING PROCESS (MS)

Network Size	Receive	Update	Notify Back
2	60.306	0.366	0.0002
4	85.257	0.415	0.0005
8	132.301	0.417	0.0002
12	83.453	0.437	0.0004
16	89.058	0.623	0.0003

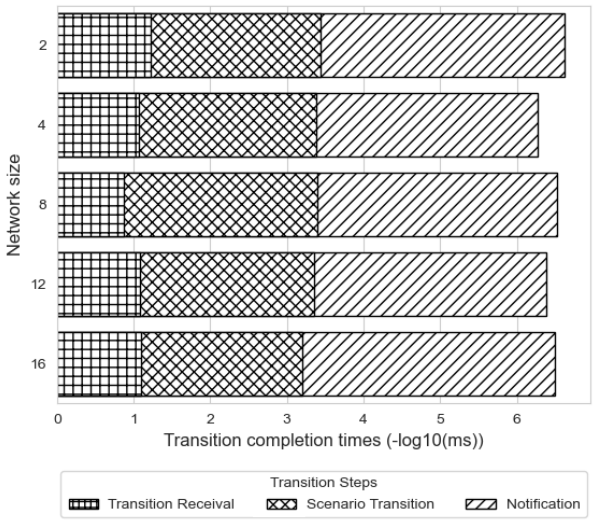

Fig. 6. Time cost of scenario transitioning process for each step (ms)

In Figure 6, the time costs of the transition process for different network sizes are presented and in Table II the transition latency values for each step are given in ms. The results indicate that, overall, the latency in the transition process remains unaffected by the tested network sizes during the simulations, except for the configuration file update process of IoT devices, which exhibits a linear correlation with the network size. The predominant portion of the latency arises from the reception of a new configuration file from the blockchain and transmitting it to IoT devices, followed by the update of the configuration file in IoT devices, accounting for approximately 1/100th of the previous process, and subsequent notification back processes, constituting approximately 1/1000th of the configuration file update process. Additionally, none of cluster managers and IoT devices exhibited any failures during the reception of the configuration file from the main chain, and IoT devices did not experienced any failure during the update and notify-back processes.

Finally, we examined the impact of cluster scalability on latency by augmenting the number of IoT devices in a single cluster, specifically from 7 to 1000, across 10 and 100 iterations. The simulation was conducted with four clusters in the environment and the outcomes are depicted in Figure 7. The total number of requests exhibits a linear correlation with the latency time, as the total requests increase from 10 to 100, the latency experiences an approximate tenfold increase. In both 10 and 100 iteration scenarios for both Scenario-1 and Scenario-2, latency values demonstrate an upward trend. However, in the case of Scenario-2 (transaction), the latency increase is comparatively less than in Scenario-1 (data store). This outcome aligns with expectations, given that the rise in IoT devices directly influences the total volume of generated data, subsequently affecting data transmission and blockchain storing time. It is noteworthy, though, that despite the exponential increase in the number of IoT devices, latency

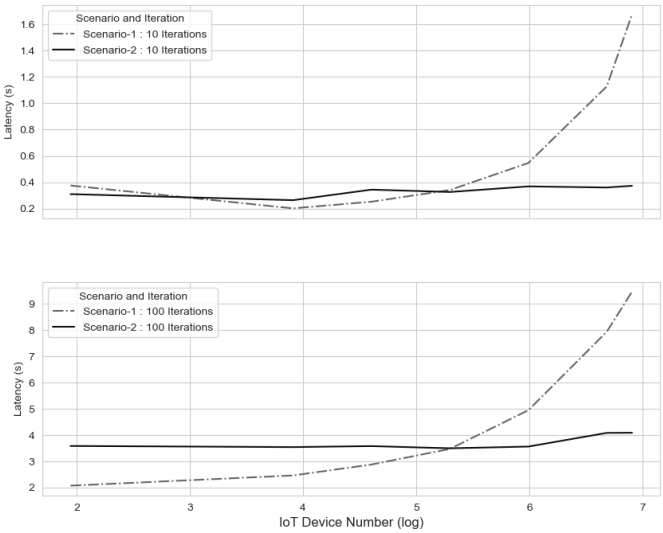

Fig. 7. Time cost per request as the clusters scale up

experiences a gradual increment.

VI. Conclusions and Future Work

In existing literature, the amalgamation of IoT and blockchain technologies has primarily focused on single-application scenarios, posing challenges when transitioning to new application types. In our foundational study, we proposed a novel BIoT architecture capable of supporting multi-application scenarios through the utilization of smart contracts.

The outcomes of conducted simulations highlight the adaptability of the proposed architecture across various applications and identified scenarios, facilitating seamless transitions without disrupting the continuous operation of the entire system. The observed scenario transition results show that the proposal is adaptable for the systems where uninterrupted availability is paramount.

During the simulation phase, we evaluated the scalability and latency of the system, gradually scaling from 2 to 16 containers, each managing operations ranging from 100 to 12,800. The absence of system bottlenecks or unavailability during the simulation phase underscores the proposal's suitability for BIoT systems, within the scope of proving the concept. However, recognizing the necessity for more comprehensive insights into much larger environments, we acknowledge the desirability of employing High-Performance Computers for future studies. As as extension to testing the proposal in larger systems, it is also desirable to conduct a security analysis within an environment where various malicious actors are present.

In prospect, our future endeavors include the development of a consensus algorithm distinct from that of Ethereum, which was employed during our simulations, as in the current version of the system, direct connectivity between IoT devices and the main chain is restricted due to the defined FRs and NFRs. While various attempts have been made to create consensus algorithms tailored for IoT devices, the majority rely on oracles or analogous centralized entities for fundamental blockchain processes such as mining, block formation, conflict resolution, and overall management. Such approaches compromise the decentralization in blockchain systems, introducing vulnerabilities reminiscent of centralized systems and requiring trust between the system entities undermining the fundamental trustless operational nature of blockchain systems.

REFERENCES

[1] T. Ali Syed, A. Alzahrani, S. Jan, M. S. Siddiqui, A. Nadeem, and T. Alghamdi, "A comparative analysis of blockchain architecture and its applications: Problems and recommendations," *IEEE Access*, vol. 7, pp. 176 838–176 869, 2019.

[2] O. Aluko and A. Kolonin, "Proof-of-reputation: An alternative consensus mechanism for blockchain systems," *arXiv preprint arXiv:2108.03542*, 2021.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized business review*, p. 21260, 2008.

[4] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with iot. challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173–190, 2018.

[5] T. Hewa, M. Ylianttila, and M. Liyanage, "Survey on blockchain based smart contracts: Applications, opportunities and challenges," *Journal of Network and Computer Applications*, vol. 177, p. 102857, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804520303234

[6] S. Brotsis, K. Limniotis, G. Bendiab, N. Kolokotronis, and S. Shiaeles, "On the suitability of blockchain platforms for iot applications: Architectures, security, privacy, and performance," *Computer Networks*, vol. 191, p. 108005, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621001225

[7] V. Y. Kemmoe, W. Stone, J. Kim, D. Kim, and J. Son, "Recent advances in smart contracts: A technical overview and state of the art," *IEEE Access*, vol. 8, pp. 117 782–117 801, 2020.

[8] H. D. Zubaydi, P. Varga, and S. Molnár, "Leveraging blockchain technology for ensuring security and privacy aspects in internet of things: A systematic literature review," *Sensors*, vol. 23, no. 2, p. 788, 2023.

[9] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, "Blockchain technologies for the internet of things: Research issues and challenges," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2188–2204, 2018.

[10] M. Krichen, M. Ammi, A. Mihoub, and M. Almutiq, "Blockchain for modern applications: A survey," *Sensors*, vol. 22, no. 14, p. 5274, 2022.

[11] Anonymised, "Anonymised," in *Anonymised*. Anonymised, Anonymised, p. Anonymised.

[12] J. Abou Jaoude and R. George Saade, "Blockchain applications – usage in different domains," *IEEE Access*, vol. 7, pp. 45 360–45 381, 2019.

[13] T. A. Syed, A. Alzahrani, S. Jan, M. S. Siddiqui, A. Nadeem, and T. Alghamdi, "A comparative analysis of blockchain architecture and its applications: Problems and recommendations," *IEEE access*, vol. 7, pp. 176 838–176 869, 2019.

[14] S. Cho and S. Lee, "Survey on the application of blockchain to iot," in *2019 International Conference on Electronics, Information, and Communication (ICEIC)*. IEEE, 2019, pp. 1–2.

[15] M. Hou, T. Kang, and L. Guo, "A blockchain based architecture for iot data sharing systems," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp. 1–6.

[16] X. Hao, P. L. Yeoh, T. Wu, Y. Yu, Y. Li, and B. Vucetic, "Scalable double blockchain architecture for iot information and reputation management," in *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, 2021, pp. 171–176.

[17] C. Xu, K. Wang, and M. Guo, "Intelligent resource management in blockchain-based cloud datacenters," *IEEE Cloud Computing*, vol. 4, no. 6, pp. 50–59, 2017.

[18] K. Košt'ál, P. Helebrandt, M. Belluš, M. Ries, and I. Kotuliak, "Management and monitoring of iot devices using blockchain," *Sensors*, vol. 19, no. 4, p. 856, 2019.

[19] N. Tsolakis, R. Schumacher, M. Dora, and M. Kumar, "Artificial intelligence and blockchain implementation in supply chains: a pathway to sustainability and data monetisation?" *Annals of Operations Research*, pp. 1–54, 2022.

[20] M. A. Baig, D. Ali Sunny, A. Alqahtani, S. Alsubai, A. Binbusayyis, and M. Muzammal, "A study on the adoption of blockchain for iot devices in supply chain," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[21] T. Le and M. W. Mutka, "A lightweight block validation method for resource-constrained iot devices in blockchain-based applications," in *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, 2019, pp. 1–9.

[22] I. Weber, Q. Lu, A. B. Tran, A. Deshmukh, M. Gorski, and M. Strazds, "A platform architecture for multi-tenant blockchain-based systems," in *2019 IEEE International Conference on Software Architecture (ICSA)*, 2019, pp. 101–110.

[23] S. K. Singh, S. Rathore, and J. H. Park, "Blockiotintelligence: A blockchain-enabled intelligent iot architecture with artificial intelligence," *Future Generation Computer Systems*, vol. 110, pp. 721–743, 2020.

[24] A. Henninger and A. Mashatan, "Distributed renewable energy management: A gap analysis and proposed blockchain-based architecture," *Risk and Financial Management*, vol. 15, no. 5, p. 191, 2022.

[25] "Scaling | ethereum.org," Nov. 2023, [Online; accessed 15. Nov. 2023]. [Online]. Available: https://ethereum.org/en/developers/docs/scaling

[26] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm (extended version)," 2013.

[27] S. Popov, "The tangle," *White paper*, vol. 1, no. 3, p. 30, 2018.

[28] S. Müller, A. Penzkofer, N. Polyanskii, J. Theis, W. Sanders, and H. Moog, "Tangle 2.0 leaderless nakamoto consensus on the heaviest dag," *IEEE Access*, vol. 10, pp. 105 807–105 842, 2022.

[29] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, "Pobt: A lightweight consensus algorithm for scalable iot business blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2343–2355, 2020.

[30] A. Dorri and R. Jurdak, "Tree-chain: A fast lightweight consensus algorithm for iot applications," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, 2020, pp. 369–372.

[31] M. Uddin, M. Muzammal, M. K. Hameed, I. T. Javed, B. Alamri, and N. Crespi, "Cbciot: a consensus algorithm for blockchain-based iot applications," *Applied Sciences*, vol. 11, no. 22, p. 11011, 2021.