Computers, Materials & Continua
DOI:10.32604/cmc.2022.019416
Article

Tech Science Press

Blockchain Based Enhanced ERP Transaction Integrity Architecture and PoET Consensus

Tehreem Aslam[1], Ayesha Maqbool[1], Maham Akhtar[1], Alina Mirza[2,*], Muhammad Anees Khan[3], Wazir Zada Khan[4] and Shadab Alam[5]

[1]Department of Computer Engineering, National University of Sciences and Technology, Islamabad, 44000, Pakistan
[2]Department of Electrical Engineering, National University of Sciences and Technology, Islamabad, 44000, Pakistan
[3]Department of Management Studies, Bahria Business School, Islamabad, 44000, Pakistan
[4]Department of Computer Science, Capital University of Science and Technology, Islamabad, 44000, Pakistan
[5]Department of Computer Science, College of Computer Science and IT, Jazan University, Jazan, 67714, Saudi Arabia
[*]Corresponding Author: Alina Mirza. Email: alinamirza@mcs.edu.pk
Received: 12 April 2021; Accepted: 27 May 2021

Abstract: Enterprise Resource Planning (ERP) software is extensively used for the management of business processes. ERP offers a system of integrated applications with a shared central database. Storing all business-critical information in a central place raises various issues such as data integrity assurance and a single point of failure, which makes the database vulnerable. This paper investigates database and Blockchain integration, where the Blockchain network works in synchronization with the database system, and offers a mechanism to validate the transactions and ensure data integrity. Limited research exists on Blockchain-based solutions for the single point of failure in ERP. We established in our study that for concurrent access control and monitoring of ERP, private permissioned Blockchain using Proof of Elapsed Time consensus is more suitable. The study also investigated the bottleneck issue of transaction processing rates (TPR) of Blockchain consensus, specifically ERP's TPR. The paper presents system architecture that integrates Blockchain with an ERP system using an application interface.

Keywords: Transaction security; enterprise systems; blockchain; transaction integrity; hyperledger sawtooth; PoET consensus

1 Introduction

Enterprise Resource Planning (ERP) is a software-based solution to manage the business processes of an organization. It combines all the aspects of business processes from planning to delivery and marketing of a product/service. The emphasis of this system is to provide a flow of information in a well-managed way [1]. The importance of Enterprise Systems provides sufficient support to unique business processes [2]. This solution of many integrated applications shares a database used by different business departments of the organization. Data is a valuable commodity for any organization. Malicious access to the database is checked by enforcing access control mechanisms like monitoring the user's insertion, modification, and deletion activities.

These access rules and logical control structures are maintained by users with administrative rights. This solves the data security and integrity issue at a broader level but results in rendering more power to the administrator. This leaves room for human malicious threats like black-hat hackers, denial-of-service attacks, sabotage, and data leaks. These threats range from altered data, ill will, processing delays, and shutdown of networks and websites. The only way to counter such an attack is through internal audits. However, audits are investigative measures by nature, not preventive. Access to the administrative role by a malicious entity leads to the problem of one-point failure as ERP lacks systemic tracking of administrators' actions, and the organization must rely on non-technical controls, which are bypassed easily. Having a single point of failure may cause massive crumbling of the entire system in case of an attack. Therefore, there is a pressing need for security in these systems. Instant assurance of trust in database transaction integrity remains a challenge, and it requires an automated solution to rectify this gap. In this paper, we discuss the possibilities of integration of database and Blockchain to bridge this gap.

Blockchain is a distributed database that holds transaction records being shared among all participating entities. This database prevents fraudulent transactions by achieving the consensus of most participants [3]. To add a record in Blockchain, it goes through the consensus process for acceptance; hence a record is never modified or deleted. Without a central authority, validation of transactions is the most critical task. Blockchain established trust in trustless infrastructure by its unique validation and consensus mechanism [4]. Presently, varied Blockchain solutions are used for varied applications; out of these three main implementations of Blockchain are; Public, Private, and Federated Blockchain. Public Blockchain allows open access to anyone who wants to participate in the network. Private Blockchain is permissioned and allows restricted access to only selected participants. Federated Blockchain is another type of Blockchain which is restricted and operates under some governing body. The most common Blockchain-based solution is Cryptocurrency, e.g., bitcoin. These solutions work with a publicly shared ledger perfect to store and perform transactions, and it guarantees those transactions to be secure. In current Bitcoin-based solutions, a block's size is 1 MB, and only 1 to 3.5 transactions per second can occur [5]. Public Blockchain suffers from issues, including privacy risks of transactions, throughput, capacity, and latency. Cryptocurrency-based solutions cannot deal with massive transactions, and it increases as well as the cost of network communication.

Private permissioned Blockchain log every action. It records all actions of the users, administrators, and anyone who attempts to change/update or tamper with data. It adds the participating nodes into the Blockchain only by invitation process and contains registered nodes' transaction data with permission. Arguably, private permissioned Blockchain is more energy-efficient and provides low costs, and consumes fewer resources with high transaction throughputs [6]. Within the Permissioned Blockchain ecosystem, user and device identity and attribute management are built-in features. Every participant node is aware of identities and transactions [7]. Records are available to track system status; thus, it can maintain and monitor transaction integrity easily. By assessing our problem against the "Do You Need a Blockchain" [8] chart, it is better to use a private permissioned Blockchain for ERP solution instead of a public Blockchain. There are many disadvantages of using a public Blockchain such as privacy, consensus and agreement, mining risks, access controls, user participation, and authorization. So, using public Blockchain in enterprise systems is a colossal risk. In this study, we analyzed the potential application of Blockchain as a transaction integrity tool. We purpose the use of a private permissioned Blockchain as an independent layer of tamper-proof record of all database transactions and operations to solve the transactional integrity issue. In this paper, we present the Blockchain architecture design details for

an automated solution to achieve instant assurance of trust in database transactions' integrity. We constructed a system model of the proposed system to evaluate system latency. Key contributions of this article are

- BlockERP Design: Explore and appreciate all essential design and architecture features of the blockchain technology required for the development of a secure and trustworthy database technology especially in the context of ERP.
- Consensus: Identification of best-suited consensus mechanism for BlockERP.
- Latency: Establishment of baseline latency performance of BlockERP.

In the rest of the paper that follows, Section 2 covers the concepts and ideas as the background to understand the core of this paper. Section 3 discussed the existing solutions and related work to this approach. Overview of Blockchain and database integration with its benefits is stated in Section 4. An integrated Blockchain and database system, along with a consensus mechanism, is elaborated in Section 5. Section 6 presents an evaluation of latency. Section 7 concludes the paper.

2 History of Blockchain and ERP

In a few years, Blockchain has emerged as one of the most innovative and trending technology in the financial world and the cybersecurity domain. This technology has been successfully applied in the finance-related application. Various economic transaction systems are using this technology. Many organizations and have successfully applied different Blockchain-based business models in the industries. It has attained ample attention from both academia and the financial world in a short time. Industries from different sectors are discussing the possibilities of creating Blockchain-based decentralized applications (Dapps) and implementing them in their organizations with desire [9]. Blockchain-powered crypto-currencies, such as Bitcoin [10] and Ethereum, have attracted many government bodies, academia, and industry.

Blockchain has evolved so much in a brief period. Blockchain 1.0 was only limited to currency in which it executed Blockchain-based financial transactions on Bitcoin technology. The popularity of Blockchain-based crypto-currencies uses Proof of Work has made scalability a significant issue. This consensus mechanism is time and energy-consuming. Bottlenecks in Bitcoin limit the ability of the Blockchain network and cannot provide support to high throughput rates and lower latencies [11]. Blockchain, 2.0 introduced smart contracts. Smart contracts are automatically executable computer programs; they define which set of rules for block validation in advance, enhance system security, and reduce the cost [12]. The most noticeable Blockchain technologies in this field are Ethereum and Hyperledger. Smart contracts and Ethereum are widespread and are being applied in many domains [13]. The use of Blockchain in the Industry sector is a significant contribution of Blockchain 3.0. Blockchain 3.0 discusses different solutions, which can make Blockchain applicable to user demands. Blockchain applications extend beyond Bitcoin and Ethereum and are applied in different domains such as finance, education, e-commerce, the internet of things [14,15], health, and governance [16,17]. In the following subsections, we will focus on the existing application of BlockChain Technology, its opportunities in distributed Product Life Cycle Management (PLM), and challenges of the existing ERP access control mechanism.

2.1 Blockchain Technology (BCT)

In addition to the financial sector, Blockchain is being adopted and integrated into various application areas. Many Blockchain-based solutions have been proposed in recent research, such as BlockChain Technology (BCT) in supply chain networks to revolutionize and reshape businesses.

Blockchain has become a new distributed information technology. It represents a new approach in the supply chain field, where visibility and transparency of production processes are the main challenges [18]. Much research is being conducted on sustainable manufacturing based on Blockchain in Industry 4.0 from the perspective of technology, business, organization, and operation. The transformation of the sustainable manufacturing paradigm driven by the Blockchain is still in the hype stage's early stage and is gradually being fully adopted [19]. Recently, blockchain technology has also aroused interest in the accounting community. A blockchain-based solution is proposed, which helps auditors by providing them reliable digital audit-proof at a reduced cost [20].

2.2 Blockchain and Product Life Cycle Management (PLM)

Traditionally, Product Life Cycle Management (PLM) is implemented based on independent and centralized systems, which makes it challenging to integrate and share. Therefore it is difficult to meet the openness and decentralization of the Industry 4.0 era Requirements. A Blockchain-based solution has been proposed to overcome this problem by facilitating the data transfer sharing in the PLM [21]. The blockchain-based solution is also being used for the world's food system, which implements blockchain technology to create transparent and traceable supply chains for food [22]. Possible uses for Blockchain in supply chains are also being explored to provide solutions for package tracking in the supply chain [23]. Although Blockchain has automated processes and reduced paperwork due to smart contracts, organizations are reluctant to adopt BCT. In other words, the Enterprise Resource Planning (ERP) transition makes the organization tired, unwilling to resist further system development, and unable to resist subsequent changes. The operation and logistics functions can be significantly improved through the Blockchain, and the integration of the Blockchain and the ERP system can completely change the way of daily operations [24].

As Blockchain technology is developing rapidly, it is offering tools to build flexible and cost-efficient enterprise systems by using its distributed ledger, smart contract, software connectors, and consensus mechanisms which provide a better solution as a Blockchain and database integration [25] Blockchain's ability to ensure transactional integrity and immutability can increase the transactional efficiency, credibility, transparency, and can also optimize the process of the enterprise systems and supply chain [26].

2.3 Conventional Access Controls

In database systems, access control models have been traditionally proposed for both application and database levels. In database systems, integrated access control model Mandatory Access Control (MAC), Discretionary Access Control (DAC) [27], and Role-Based Access Control (RBAC) [28] provide efficient security and can also be implemented with the encryption algorithms [29], but these systems do not provide decentralization, immutability, and owner-controlled access.

With all benefits, the ERP system also faces some challenges related to data integrity. Issues with current ERP systems are that performed transactions in these systems are less efficient, costly, and at risk of several attacks. Also, these transactions are costly to manage and review with much difficulty in tracking. This problem was considered lacking any kind of solution recently until Blockchain appeared on the scene. One of the other issues of enterprise systems organizational structure is the lacking of decentralization [30]. Therefore, enterprise systems can benefit from the decentralized feature of Blockchain technologies. The transactions recorded through the Blockchain might provide safe, credible, immutable, and fully transparent services to

all stakeholders worldwide [31,32]. Blockchain helps in building credibility and trust with business parties, simplifying future transactions. Unlike other decentralized applications and Blockchain technology projects, little work exists to address the integration of conventional and Blockchain-based solutions for ERP. There is a lot of research work needed to benefit from Blockchain technology for enterprise systems. In this paper, we discuss the aspects of Blockchain technology and its integration with the ERP system by proposing an architecture for this integration. We also present system models for the evaluation of latency-based performance.

3 Related Work

Enterprise systems include unique solutions such as ERP solutions for automating transactions. Supply Chain Management [33] offers additional refined planning. Product Life Cycle Management is used to integrate data related to the product with other extended business processes [34]. Another enterprise system is customer relationship management CRM, which helps manage customer relationships [35]. However, ERP is the essential category of enterprise systems [36]. ERP system is software that integrates business processes related to finance, supply, manufacturing, and resources with information management to enable resource management across the whole enterprise. Another benefit provided by ERP systems is the foundation of e-commerce. Customers can order, track, and check the delivery process through web-based customer access. In database systems, all participants have a full copy of the transaction and its history; it makes the security risk very high for the organization and its users. Lack of credibility of transactions has become a significant issue for making the development of e-commerce difficult. Adopting Blockchain protocols to existing solutions can end this issue. However, it still faces significant scalability issues such as throughput, latency, and instant transaction [37]. Encryption of Blockchain data is desired to overcome risks [38]. Blockchain technology can secure the integrity of transactions stored in the database. Well-formed transactions, access control, audit, and user authentication provided by Blockchain can decrease threats to data integrity [39]. Blockchain-based Distributed Autonomous Organizations (DAO) lack central authority. Instead of central authority, any modification in Blockchain is achieved when all the network participants reach a consensus, making transactions of the system more secure [40]. BlockchainDB [41] proposed a Blockchain-based database application platform ChainSQL with the feature of decentralization, audibility, and distributed nature. The limitation of ChainSQL is that it is a public Blockchain and is based on Proof of Work consensus. Public Blockchain networks using Proof of Work do not suit well for large-scale system implementations because of high cost and scalability issues. Transaction throughput and latency will become a bottleneck in this case [42].

In recent years, few systems are proposed as a replacement for the ERP Database with Blockchain. One of these solutions is BigchainDB. It is a database that has properties of Blockchain like immutability decentralization, assets controlled by the owner, and characteristics of a database such as high transaction rate, low latency, data querying. BigchainDB is used for business processes only, such as booking apps and delivery tracking apps. BigchainDB is famous for keeping the privacy of data, but it differs from ensuring data integrity. One of the other limitations of BigchainDB is that it only works on MongoDB.

Another solution proposed is Blockstream, which works with Sidechains. Sidechain is a Blockchain running parallel to bitcoin and benefiting from bitcoin security. Blockstream helps in securing asset transfer among chains. Another solution is Bluzelle, which is a decentralized ecosystem that allows complete control of data along with monitoring. This system uses Bluzelle cryptocurrency for payment between the parties. These systems replace the ERP database with

Blockchain, which raises performance issues. These systems offer a replacement of database systems, which is very hard to adopt in the current industry as all the organizations are using ERP systems. Replacing ERP systems in all organizations is not a feasible and affordable choice. The main issue while adopting Blockchain technology is latency and scalability. One of the main problems regarding Blockchain-based technology is its ability to scale up to industry-wide standards. In Bitcoin, a maximum of 3–4 transactions are processed per second. Ethereum has the same problem, it can process a maximum of 15 transactions per second, but in the industry, we need a large number of transactions per second. As databases work with hundreds of transactions, integrating the database with Blockchain might slow the database system.

This paper proposes an alternative model that leverages Blockchain and integrates it with a conventional solution using both systems' features. This proposed system uses a consensus mechanism and validation techniques to ensure transaction integrity. This solution identifies the scalability gap and provides a better transaction rate per second than existing solutions. The strength and limitations of different Blockchain-based systems are presented in Tab. 1.

Table 1: Strength and limitation of different blockchain-based systems

Solution	Strength	Limitation
ChainSQL	Public blockchain based on proof of work consensus	Using proof of work does not suit well for large-scale system implementations because of high cost and scalability issues.
BigchainDB	Proposed as a replacement for the ERP Database with Blockchain	Used for business processes only. Only works on MongoDB.
Blockstream	Sidechain-blockchain running parallel to bitcoin	These systems replace the ERP database with Blockchain, which raises performance issues like latency and scalability.
Bluzelle	An ecosystem that allows complete control of data along with monitoring	
Bitcoin	Digital currencies—can be used in financial transactions	Slow transaction processing.
Ethereum		Databases work with hundreds of transactions, integrating the database with Blockchain might slow the database system.

4 Blockchain and Database Integration

The database provides many benefits as it centralizes business data and provides a single point of contact. In databases, it provides full control of the operations to the database administrator. This has proven to be useful for making informed decisions for the future. The system is usually fully functional as it performs data updates in real-time. With all these benefits, databases still have some limitations which can be addressed by integrating Blockchain with the database.

4.1 Data Integrity Assurance

Data integrity means that only authorized persons can access and modify the data, and the data is always accurate and consistent. Data integrity assurance issues arise as there is no proper system to evaluate erroneous access/modifications in the database. There is a risk of compromised data integrity because of threats like human errors, transfer errors, compromised hardware, cyber-attacks, security errors, and malware, etc. With all these associated risks, data integrity assurance has become extremely challenging. The existing solutions for trust management of ERP have predominately centralized management ways, which create the risk of the data being tampered with by an inside malicious intent manager or attacked from outside incursion. Blockchain help caters to this risk by ensuring that data cannot be freely modified, deleted, or forged anyway. Blockchain is a fault-tolerant database, and data stored in this database is immutable. The architecture of Blockchain is decentralized, which makes it robust against any hacker and security breaches. It also relies on public key infrastructure; using such infrastructure provides encryption, which is expensive and resistant to crack [43]. In Blockchain multiple copies of data are maintained at participating nodes, it is recommended to never store personal identification information or any other sensitive data on the blockchain. In Blockchain pseudonym identities are used instead of actual identities, which help ensure trust among the network. Mostly, pseudonym identities are used instead of real identities which help build trust among the network. Instead of actual data hash of encrypted data can be stored on the chain for validation purposes while maintaining actual data either in a co-located, off-chain database or an existing source system. Digital assets such as transactions or files can be hashed using hashing algorithms and then the hash of data can be stored onto a private (permissioned) blockchain.

4.2 Single Point of Failure

A single point of failure is a common issue in database systems. There are many areas where a single point of failure can occur in a database system. A single point of failure, such as a database server's failure, can cause the entire system to shut down. These failures can compromise the availability and functionality of the entire system. Integrating Blockchain with a database can cover this issue as there is no central authority in Blockchain, and the system is decentralized. Blockchain has a copy of the same information across the network, and the same information is broadcasted to all the participating nodes. A single person does not control it; ledgers are unmodifiable, which protects the transactions. In Blockchain, no one is in charge of Blockchain, which eliminates a single point of failure. Keeping these issues in view has become evident that a database and Blockchain integration can solve these critical problems.

5 Proposed Solution

Integrating Blockchain with database systems can solve the issues discussed previously, as Blockchain can provide the cross-checking ability to assess the erroneous behavior of a database using smart contracts. Blockchain and database integration may provide the benefits of verifying and authenticating identities. Transparent systems will help in reducing the cost of tracking and reporting, which is significant. Implementing smart contracts will help to reach an agreement, formalization, and enforcement automatically. An enhanced security setting is provided by which system security and data privacy are ensured, and solid ground for assurance of risk-free financial transactions is provided. We propose a system BlockERP in which Blockchain runs in synchronization with the database and validate the transactions as they happen. The architecture of the BlockERP comes next in this section.

5.1 BlockERP Architecture

We have proposed BlockERP architecture after analyzing data integrity issues in Enterprise systems. Then we explored Blockchain technology and its applications in particular private permissioned blockchain and its related projects. After highlighting several design features in this article we focused on the selection of consensus that improves latency of ERP-Blockchain hybrid system.

Transaction integrity is an uncompromising proposition. Data is not static; it can change throughout its existence. When a transaction is processed, one or more records in the database get updated, and changes are made. Updates and modifications are acceptable as long as it is consistent across the systems and storage. During the transaction process, if an error occurs, it affects the entire database. It is essential to achieve data integrity, to maintain quality processes, and adequately working systems. In the proposed architecture, we present two scenarios.

User queries are only processed through database validation and access control mechanisms and are recorded in the database. Data that is selective and important for which database validation and access control mechanisms are not enough is copied on the Blockchain. These transactions are validated after achieving consensus. Validated transactions are broadcasted to the Blockchain network and are stored in the database. The status of both databases and the Blockchain is maintained so that transaction integrity could be assured. The proposed system's architecture is presented in Fig. 1 with components described in this section.

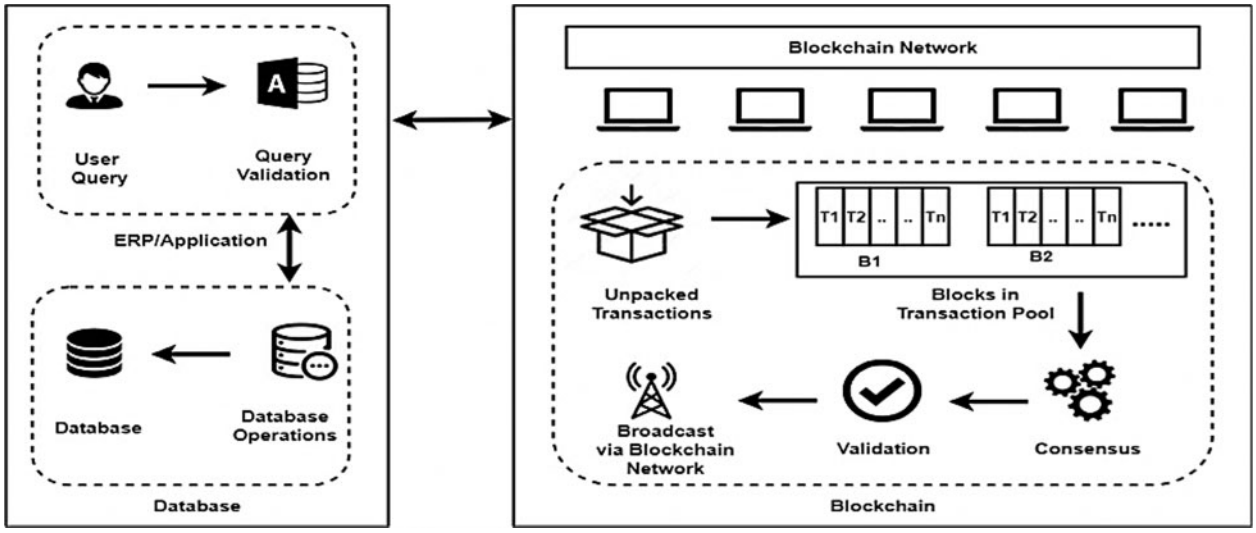

Figure 1: Architecture of the proposed system block ERP

5.1.1 Application Interface

When a user queries the database, the query, and the user profile details are sent to the database as a transaction where the database validation business rules validate it. If the transaction is invalid, an error message is shown to the user via the application interface. If the transaction is valid, it is sent to the Blockchain network via the application interface.

5.1.2 Blockchain Network

A Blockchain network is a group of devices or connected computers. These devices or connected computers are called nodes. These nodes make up a Blockchain network. The Blockchain network has no central authority. Transactions generated by users through the application interface are forwarded to the Blockchain network. At the Blockchain network, consensus occurs, and the nodes in the network validate each of the transactions. After validation, these transactions are broadcasted to every peer node in the network, and the record of these transactions is recorded on their copy of the ledger.

5.1.3 Transaction Pool

After being forwarded from the application interface, it places transactions as unpacked transactions in the transaction pool. It updates the transactions and puts them into the transaction pool, and these pooled transactions are packed into packs. Each pack contains 'N' transactions. These packs are updated in the form of a block by the block generator. The block generator creates a new block after consensus.

5.2 Consensus and Validation

BlockERP Blockchain comprises four phases. During the first phase, the user initializes the transaction. It stores the transactions in the form of unpacked transactions. The second function is the block generator. Transactions are packed and submitted in the batch, and a block is generated. 'n' number of transactions can be submitted into a block. This block is forwarded to the validator nodes and for the process of consensus. After a consensus is achieved, and all the validatory pass the validation, block validation is finished. Block is published and broadcasted to all the network nodes, and the chain is updated.

In these access controls, business rules are defined to avoid conflicting transactions and issues like system forks. These rules validate if a user has permission to perform a transaction and transaction integrity is validated as well. After validation, if the transaction is valid then the transaction is forwarded to the ERP database for database operations and the Blockchain network for consensus and agreement. Otherwise, it is denied, and a message is sent back to the user through the ERP.

5.3 MultilevellHierarchical Blockchain

Our proposed system works with a multilevel hierarchical blockchain. In Fig. 2 multilevel hierarchical blockchain is presented. Three levels of this blockchain are as follows.

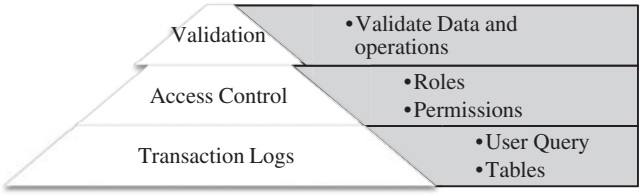

Figure 2: Multilevel hierarchical blockchain

5.3.1 Level 1: Transaction Logs or Data (Data Restoration)

The first level of blockchain can provide data restoration for the database by maintaining the ERP status so in case of any issue or error the database can roll back to its previous state. This level is for maintaining the transaction status both in blockchain and ERP. This level provides two options for this purpose: First to store the actual data on the blockchain in compressed form (Merkle tree) and cryptographically encrypted. Different data backup options like full backup, incremental backup, differential backup, and mirroring are used.

Second to Store the transaction logs of the Data on the blockchain In this case, data can be stored as transaction logs of the database which maintain the status of the database as well as the transactional history. Using the two, it is always possible to roll back our database to any desired previous state in case of any issue or error.

5.3.2 Level 2: Access Control/Business Rules

The second level of blockchain holds the business rules and access control details. This level holds information regarding:

(1) User access rights granted to the ERP system.
(2) Activities of Authorized Users who have system access.

5.3.3 Level 3: ERP Data Validation:

This level ensures the integrity of the transactions and makes sure that the same records are placed on the database of the ERP and the Blockchain. This level holds rules through which ERP and Blockchain will be able to communicate with each other at any point in time to synchronize themselves with each other.

5.4 System Model

As shown in our BlockERP system's architecture, it comprises three parts, i.e., a client application that sends transaction requests both to the database and Blockchain network. Transactions are processed and saved into the database. At the same time, transactions are sent to Blockchain for record-keeping and immutability. In this section, we discuss the Blockchain network. For transaction processing of our system, after transaction initialization, blocks are generated which gather, validate, and pack the transactions into a block. Then block is broadcasted for consensus, and after consensus, the block is added to the Blockchain system. The key steps of this model are given below.

5.4.1 Transaction Initialization

In this system, transactions are initialized. The user sends a transaction request, and this request is added to the transaction pool of unpacked transactions. These transactions are then packed and sent for block generation.

5.4.2 Block Generator

In this system, we assume that there are N nodes and T transactions, and a newly generated block is denoted by B_i, which can contain $[t_1, t_2, \ldots, t_n] \in$ T transactions. The Block state is BSO if it is open for consensus, and after consensus, the block state is BSC. Whenever transactions are requested, they are packed and Block B_i is generated, then forwarded for consensus.

5.4.3 Consensus and Validation Model

For consensus, we are using the PoET consensus mechanism, in which the shortest allocated time gets to be the winner of the block. Time is allocated to all validator nodes in the block, and the one with the shortest wait time is the leader of the block. The client sends a request which is forwarded to all validator nodes, all the nodes reach a consensus, and the block is committed, and the response is sent to the client.

In this paper, we discuss the latency of the proposed system, which is evaluated by the Blockchain's ability to complete a transaction. In PoET, two main steps are considered as discussed in the system model, the first step is block generation, and the second is consensus achievement. So, latency or final time includes both block generation and consensus achievement and is denoted by:

$$TF = TI + TB + TC \tag{1}$$

TF is the time of finalizing the transaction, TI is the time of transaction initialization, TB is blocked generation time, and TC is time to achieve consensus. Consensus latency TC can be measured in two steps; time is taken to deliver messages TDM and verify message TVM and is given by:

$$TC = TDM + TVM \tag{2}$$

5.5 PoET Consensus Model

Different types of consensus algorithms exist, which makes it very difficult to select one according to the requirements. In a decentralized system, there is no perfect consensus algorithm available There is always a trade-off between different properties of consensus mechanisms according to the system requirement. [44,45] presents the comparison of existing consensus models it shows that PoET is only private/public blockchain consensus with the minimum cost. BlockERP achieves consensus using the Proof of Elapsed Time (PoET) consensus mechanism. We chose PoET after reviewing all available consensus mechanisms and comparing them against system requirements and have modeled its network latency in detail in Section 5.6. The concept of PoET was proposed by Intel, in 2016. It provides high technology tool for solving the computing problem of the "random leader election." This consensus algorithm is mostly used and is suitable for private permissioned Blockchain networks to decide the block winner/validator on the network. It is being used by Hyperledger for implementation and experimenting with private Blockchain. The basic strategy of PoET is that every participant waits a random amount of time in the Blockchain network. The leader of the block is selected according to the finish waiting time. The participant who wins gets to bet the leader of the block.

In our network, we have three types of nodes; Complete node, Partial node, and Reader Nodes. There are 'N' nodes in the Blockchain network out of which some complete nodes conduct PoET consensus among them. There are some parts and reader nodes in the Blockchain network as well, but they have limited access. Consensus is achieved by complete nodes only. Consensus outline using PoET is presented in Fig. 3.

5.6 Consensus

PoET can tolerate $(N-1)/3$ faulty nodes in the entire Blockchain system. PoET communication protocol has five phases, as shown in Fig. 4. We have adopted this model from [46] and designed it specifically for PoET consensus. In the first phase, a request is sent by client zbc for

validation of a block. Client sends this request to primary validator *zbp* ($p = 1, \ldots, N, p\,c$). In request, a signature is included, which is verified against all validators. In the second forward phase, a batch of requests is processed, and block B is generated, which is forwarded to all validators in the Blockchain networks $N-1$ Message Authentication Codes (MACs) are generated by the primary validator, and every node needs to verify MAC.

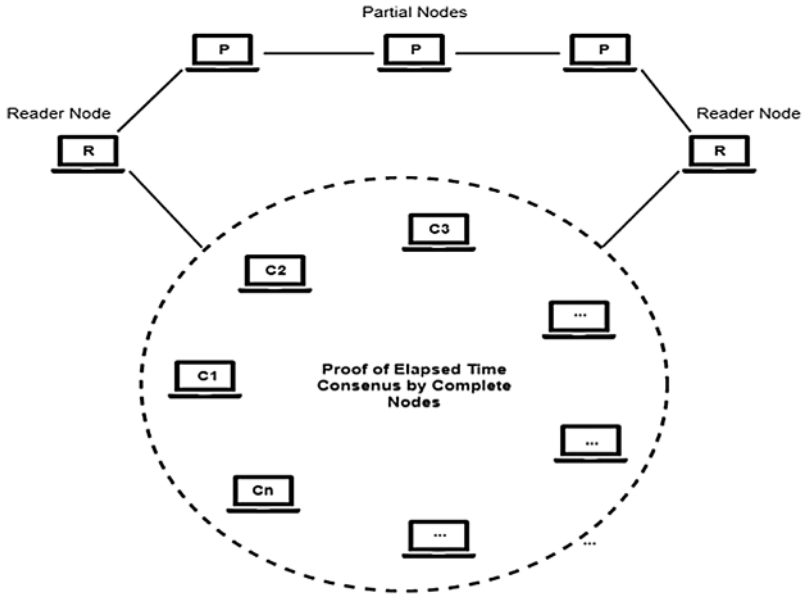

Figure 3: Consensus outline using PoET

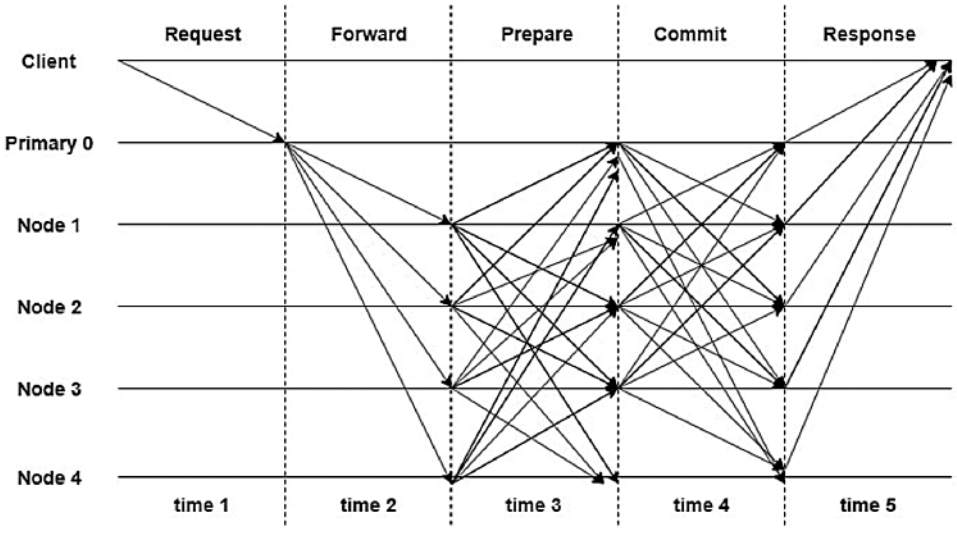

Figure 4: Protocol communication pattern of PoET

In the preparation phase, authentication of the forwarded message in the second phase is done, and each node generates $N-1$ MACs, and every node verifies $N-2$ MACs. At the same

time, the primary validator receives MAC from all nodes and verify $N-1$ MACs. In the commit phase, the message is exchanged by all nodes. The primary validator node and other nodes send and receive $N-1$ commit messages. In the response phase, a final MAC is generated by the primary validator, and all other nodes for every request and response are sent back to the client.

So, for each block B, primary validator zbp needs to process and verify B signatures and also complete $2B+4(N-1)$ MAC operations, and every node $zbi(ip, c)$ is required to verify B signatures and complete $B+4(N-1)$ operations. In the worst-case scenario, computational load for each batch for other nodes zbi is:

$$o_{bi} = B\alpha + [B+4(N-1)]\beta \tag{3}$$

where α and β are the required CPU cycles, and for primary validator is:

$$o_{bp} = B\alpha + [2B+4(N-1)]\beta \tag{4}$$

By this computational load, validation time can be calculated for each request by:

$$T_V = \frac{1}{B_{n-1\ldots N}}\left[\frac{O_{bn}}{C_{bn}}\right] \tag{5}$$

And time to deliver a message for each request can be calculated by the following equation where S is the block size, τ is the timeout in which any message is delivered and $R_{bi,bj}$ is the data transmission rate that connects the validator.

$$T_D = \frac{1}{N}(t_1+t_2+t_3+t_4+t_5) = \frac{1}{N}\left(\left\{\frac{BS}{R_{b_C,b_p}},\tau\right\}+\left\{\max_{i\neq c,p}\frac{BS}{R_{b_p,b_i}},\tau\right\}+\left\{\max_{i\neq j;\,i,j\neq c}\frac{BS}{R_{b_i,b_j}},\tau\right\}\right.$$
$$\left.+\left\{\max_{i\neq j}\frac{BS}{R_{b_i,b_j}},\tau\right\}+\left\{\max_{i\neq c,}\frac{BS}{R_{b_i,b_j}},\tau\right\}\right) \tag{6}$$

Algorithm 1: Proof of Stake Working

function CONSENSUS (Bi: Block of undecided transactions N Validator Nodes)
Receive Transactions $t_1, t_2, t_3,\ldots, tn$
for all transaction $[t_1, t_2, t_3,\ldots, t_n] \in$ T **do**
Generate Block ($Bi \leftarrow [t_1, t_2, t_3,\ldots, t_n]$)
$B_{SO} \leftarrow \Phi$ // Block close for consensus
$B_{SC} \leftarrow \Phi$ // Block open for consensus
$B_{SO} \leftarrow [t_1,t_2,t_3,\ldots, tn] \in T = B_i$
Send new block B_0 to validator nodes $[n_1, n_2, n_3,\ldots, n_k] \in N$
$B_{SC} \leftarrow$ Consensus Achieved (B_{SO})
if $B_{SC} = \Phi$ **then repeat**
$N \leftarrow \forall t \in B_0 |rt| = r_{min}$
$B_{SC} \leftarrow$ Consensus Achieved
(B_{SO})) **until** $B_{SC} = \Phi$
Apply B_C
$B_i \leftarrow B_i + 1$
return B_i

5.7 Algorithm for BlockERP System Routine

In Algorithm 1, an outline of a consensus routine is provided. System consensus has many phases. In the first phase, transactions are received, and several transactions are packed, and a block is generated, which is open for consensus. Transaction in the block needs to be validated for ensuring transaction integrity. Until consensus is not achieved, the block is open for consensus, and validator nodes need to reach an agreement to achieve consensus. Block is sent to validator nodes. After consensus achievement, the block is closed for consensus. For consensus achievement, random time "r_t" is allocated to the validator nodes. The node with the shortest wait time "r_m" wins and gets to be the block's leader for block validation. Block is added to B_{SC} after achieving consensus, and new block B_i (block of undecided transactions) is added for the consensus.

6 Evaluation of Proposed System

To prove the concept of our proposed architecture, we used Hyperledger Sawtooth (https://sawtooth.hyperledger.org/docs) Sawtooth is a framework designed for enterprise-related distributed ledgers as the private Blockchain. We used the ledger state of Hyperledger Sawtooth to represent our transactional data, while the validator was used to enforce rules and policies. Implement ion our work is done using Docker [30], which uses Ubuntu images and containerizes the application using Docker containers.

Hyperledger Sawtooth works with transaction families. The Sawtooth application layer of the system is separable from the core system, so it provides the possibility to build and define a transaction family on top of built-in Blockchain architecture. The transaction family includes a transaction processor, also known as a smart contract in other Blockchain architecture. In this transaction processor, rules and conditions are defined for the Blockchain incoming transactions. The same validator can be associated with more than one transaction processor, which is used to manage block validation. An addressing space is assigned to every transaction processor in which information acquired from the family name is stored. After passing through, the log transactions are directed to the associated transaction processor, and the payload of the transaction is consumed before passing the transaction to the validator. Rest API works as an interconnect between the validator and the client, passing requests from the client to the validator. Data is transferred between both sides in encoded form, and only a participant with RSA key pair can send the transaction to the network. If there is a need to send over one transaction, multiple transactions can be encapsulated in a batch, and the batch is transmitted.

6.1 Transaction Family for Proposed System

For proof of concept, we developed our own transaction family and implemented it in Hyperledger Sawtooth. In our transaction family, operations are performed at the schema level. For instance, storing table name. When a new table is created, or an already existing table is updated. In that case, the client saves these transactions in Blockchain, which will provide immutability. Data can never be modified once stored, which ensures the data's transaction integrity. In case of modification of table, the ERP schema only maintains the new table however in our BlockERP the details of the old and new table will be logged. Old table details being immutable stays the same and modifications are entered a new record.

The transaction family allows the client to set data parameters and retrieve them. These transactions sent by the client are validated by validators and processed by the transaction processor. A client can view all the transactions stored in Blockchain and list the states of Blockchain's available blocks. The client uses the command line application interface and performs operations

of set data, getting data, viewing blocks, and viewing the state list. Transactions are requested, and data is retrieved.

6.2 Transaction Payload

In the BlockERP transaction family, we have two object encoded fields, and these fields are the table name and action performed. Only these two fields can be passed through the payload by the client. In the transaction payload, we also added the date and time functions that display the transaction date and time with transactional data Transaction header parameters in our defined payload are input, output, and dependencies. The parameters of the transaction header are the inputs and outputs (the address generated using the device URI), the dependencies (in our case, None since our transaction family does not depend on any other transaction family), the family name (i.e., Simplestore version is 1.0), and the encoding (the encoding field needs to be set to application).

6.3 Address and State

In Hyperledger Sawtooth, data is stored in a Merkle Tree. In our system, data is stored in the state. Addresses are constructed in a format that an address contains a hexadecimal string of 70 characters. The first six characters are formed SHA512 hash, which we are using to encrypt namespace prefix, and the rest of the 64 characters are from the unique encoding format's SHA512 hash. Both namespace prefix and unique encoding format are used to generate the address. A state is an important place as it handles the layer of data. We created a class in which context was taken as a constructor. This context includes the data layer object, which is being used for setting transaction data and retrieving it. We used state entries for setting the state, and it contains the addresses with data. New blocks are built on top of the Blockchain, and the state can be checked.

6.4 Execution

Transactions are executed in Hyperledger Sawtooth in a sequence. The transaction processor receives the client's request, and the payload is verified according to the formal validation rules defined in the transaction processor. Transaction processor checks if data entered is sent by a permission participant who has authorization to the network, signatures are verified. Operations entered are checked if they are valid operations or not. Chain validation is checked before adding the block to the Blockchain invalidation. If the transaction is valid and validators pass the transaction, then the transaction requestor public key is verified against the signature, and block validation is finished. Block is added to the Blockchain and broadcasted on the network.

6.5 Transaction Flow of BlockERP System

This paper provides proof of concept for our proposed architecture and consensus by using Hyperledger Sawtooth We test our proposed architecture with PoET. The transaction processor SDK is used to verify against defined rules and policies. The transaction family is registered and broadcasted to the validators. A transaction is requested through the client application and is sent to the validator through REST API. The validator acts as an interconnect between both transaction families, and the client requested a transaction after the transaction satisfies all the validation rules for integrity. PoET consensus is used in which validators are provided wait time, and one with the shortest wait time wins and gets to be the block leader. If consensus is not achieved, the client is notified, and the transaction is terminated. If consensus is achieved, the process of block validation starts. If validation fails, the transaction is terminated, and the client

is notified. On the other hand, if validator nodes pass the validation, the block is finalized, and the block is published. The transaction flow of our system is presented in Fig. 5.

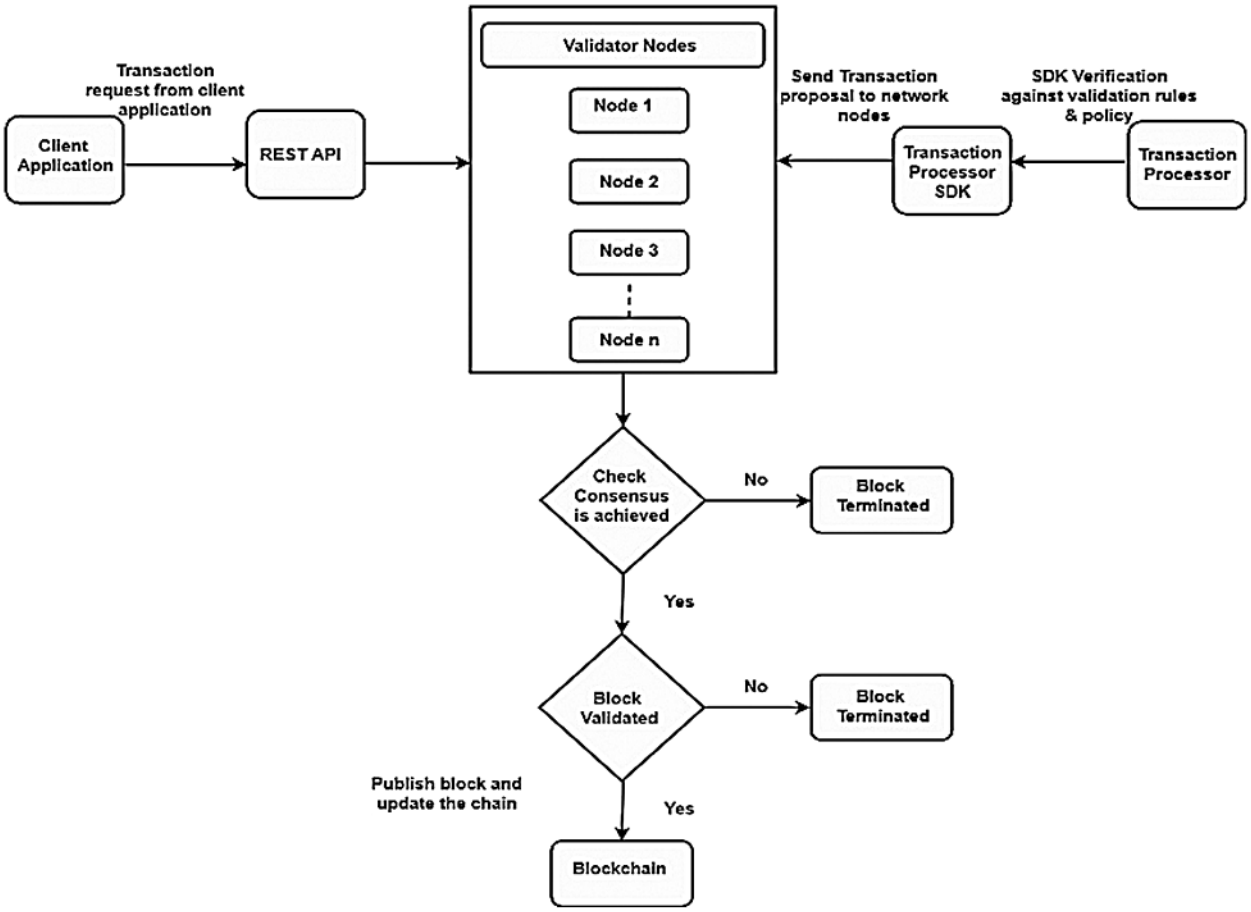

Figure 5: Transaction flow diagram for proposed system

6.6 Performance Analysis

In Blockchain, there is always trade-off security, decentralization, scalability, and latency of the system. The theorem of robust distributed systems is that "A robust distributed system can only simultaneously provide two out of the three properties," just like CAP theorem [47], which provides a concept that only two properties can be achieved out of availability, consistency, and partition tolerance. In Blockchain, scalability is the ability of transaction processing of any Blockchain system. Blockchain faces the issue of scalability, and researchers are working to make Blockchain more scalable, which will be able to handle an increasing number of transactions. Decentralization is the Blockchain system's fundamental characteristic, which means that there is no need for any central control, and third-party, which provides the system immunity from different attacks and eliminates a single point of failure. The third property of the Blockchain system is security; the blockchain system comes with features of immutability, which makes the system secure, and security should be assured for different types of attacks such as the Sybil

attack, DDOS, and 51% attack, etc. The fourth and most important property of the Blockchain is Latency, which is discussed in this paper. In Blockchain, latency is measured by the Blockchain system's time to finalize a transaction and make it irreversible. In the existing Blockchain, there is a trade-off between these properties. In Public Proof of Work Blockchain systems such as Bitcoin and Ethereum, decentralization and adequate security are achieved while compromising scalability and latency. Some Blockchain systems compromise on decentralization for achieving scalabilities such as EOS and Cardano. While in some Blockchain systems such as AION and Cosmos, systems gain fast scalability and latency with decentralization, but they sacrifice security and risk cyberattacks. While our proposed system is decentralized, secure, and provides fast latency than Bitcoin and Ethereum. In this article, we have tried to establish the baseline latency delay between centralized ERP and the blockchain. Specifically, we have focused on the evaluation of consensus algorithms and their impact on latency. As a centralized system, ERP has very high processing rates. Establishing the baseline delay of the blockchain system is crucial to the configuration of BlockERP. The decision of either slowing the ERP or introducing buffers for pending transactions are depended upon the results we have presented in the evaluation section. For establishing latency delays we present the result of thousand transactions using PoET on our defined validator on ERP Schema are presented in Tab. 2.

Table 2: Comparison of different blockchain systems

Blockchain system	Transaction rate per second
Bitcoin	4–5
Ethereum	15
BlockERP	45

For our architecture implementation, we used Hyperledger Sawtooth. In Bitcoin, a maximum of 3–4 transactions can be processed per second. Ethereum has the same problem, it can process a maximum of 15 transactions per second, but in the industry, we need a large number of transactions per second. To improve our transaction rate, we worked with Hyperledger Sawtooth as it also provides the option of building a private Blockchain, which is our basic system requirement. As for database selection, we used MongoDB, which processes thousands of transactions per second. As the database work with hundreds of transactions, integrating the database with Blockchain might slow the database system. To cater to this, we shall be processing selective data transactions through Blockchain, and the database access control will handle the rest of the transactions.

In Hyperledger Sawtooth, our Blockchain transaction processing is shown in Fig. 6. The number of transactions is displayed on the x-axis processing time in milliseconds is presented on the y-axis. The processing of different numbers of transactions is calculated in Milliseconds and displayed in the following graph.

A comparison of Blockchain with MongoDB, which processes thousands of transactions in a second transaction processing time of Blockchain, is slow. However, transaction processing time can be optimized with Hyperledger. A proposed solution for making sure that Blockchain works parallel with the database we define two scenarios.

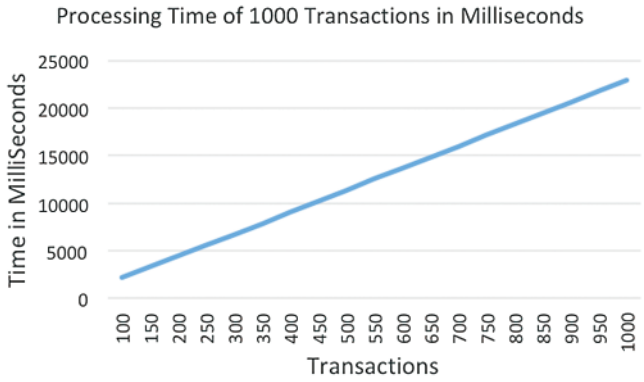

Figure 6: Processing time of 1000 transactions in millisecond

(1) Only selective data should be processed through Blockchain.
(2) In Blockchain, we propose using buffers on every node, which can contain thousands of transactions and make it possible for Blockchain to work in parallel with the database.

In the graphs, Blockchain's transaction processing rate is approximately 45 transactions per second, which is much better than Bitcoin, which processed only 3 to 4 transactions per second, and Ethereum with a transaction rate of 15 per second. The transaction processing rate of the BlockERP system compared with Bitcoin and Ethereum is presented in Tab. 2.

7 Conclusion

The paper evaluates the blockchain technology for ensuring transaction integrity in ERP solutions. We propose BlockERP as an integrated architecture of private Blockchain and database systems to address the challenges of a single point of failure. The paper provides a detailed architecture of integrated ERP and Blockchain highlighting the selection of PoET consensus to reduce latency. We present a detailed model of PoET to establish the performance of Block-ERP along with experimental results of a customized transaction processor. Our results and architectural details are significant in establishing the baseline latency of BlockERP. Proposed PoET based BlockERP outperforms etherum and bitcoin in terms of several transactions ber millisecond. However, the blockchain is much slower than centralized ERP. In organizations where access to sensitive data is of the highest priority, this latency may seem affordable. The blockchain application proposed here can be deployed easily by organizations using database systems. This deployment will strengthen data and transaction integrity. Companies can keep their database systems; they will only need to integrate Blockchain into the existing system, as suggested in our proposed system.

8 Future Work

For future work, Decentralized Application is to be developed using Hyperledger Sawtooth, which integrates both systems. Several possibilities of future work related to this paper are in the pipeline form the point of view of system implementation such as scalability, validation testing, implementation of complex query databases, performance analysis of such systems, suggesting and implementing a new encryption algorithm for providing more security to the data.

Funding Statement: The authors extend their appreciation to the National University of Sciences and Technology for funding this work through the Researchers Supporting Grant, National University of Sciences and Technology, Islamabad, Pakistan.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] T. Cragg and T. McNamara, "An ICT-based framework to improve global supply chain integration for final assembly SMES," *Journal of Enterprise Information Management*, vol. 31, no. 5, pp. 634–657, 2018.
[2] S. Kurnia, T. Linden and G. Huang, "A hermeneutic analysis of critical success factors for enterprise systems implementation by SMEs," *Enterprise Information Systems*, vol. 13, no. 9, pp. 1195–1216, 2019.
[3] D. Efanov and P. Roschin, "The all pervasiveness of the blockchain technology," *Procedia Computer Science*, vol. 123, no. 3, pp. 116–121, 2018.
[4] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda and V. Santamaria, "To blockchain or not to blockchain: That is the question," *IT Professional*, vol. 20, no. 2, pp. 62–74, 2018.
[5] D. Puthal, N. Malik, S. Mohanty, E. Kougianos and C. Yang, "The blockchain as a decentralized security framework future direction," *IEEE Consumer Electronics Magazine*, vol. 7, no. 2, pp. 18–21, 2016.
[6] R. Arenas and P. Fernandez, "CredenceLedger: A permissioned blockchain for verifiable academic credentials," in *IEEE Int. Conf. on Engineering, Technology and Innovation*, Stuttgart, Germany, pp. 1–6, 2018.
[7] D. W. Kravitz and J. Cooper, "Securing user identity and transactions symbiotically: IoT meets blockchain," in *2017 IEEE Global Internet of Things Summit*, Geneva, Switzerland, pp. 1–6, 2017.
[8] M. Peck, "Blockchain world-do you need a blockchain? This chart will tell you if the technology can solve your problem," *IEEE Spectrum*, vol. 54, no. 10, pp. 38–60, 2017.
[9] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng *et al.,* "Decentralized applications: The blockchain empowered software system," *IEEE Access*, vol. 6, pp. 53019–53033, 2018.
[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: https:// Bitcoin.org/Bitcoin.pdf.
[11] K. Croman, C. Decker, I. Eval, A. E. Gencer, A. Juels *et al.,* "On scaling decentralized blockchains," in *Int. Conf. on Financial Cryptography and Data Security*, Berlin, Heidelberg, Springer, pp. 106–125, 2016.
[12] M. Xu, X. Chen and G. Kou, "A systematic review of blockchain," *Financial Innovation*, vol. 5, no. 1, pp. 1–4, 2019.
[13] A. Srivastava, P. Bhattacharya, A. Singh and A. Mathur, "A systematic review on evolution of blockchain generations," *Information Technology & Electrical Engineering*, vol. 7, no. 6, pp. 1–8, 2018.
[14] M. A. Togou, T. Bi, K. Dev, K. McDonnell, A. Milenovic *et al.,* "DBNS: A distributed blockchain enabled network slicing framework for 5G networks," *IEEE Communications Magazine*, vol. 58, no. 11, pp. 90–96, 2020.
[15] M. A. Togou, T. Bi, K. Dev, K. McDonnell, A. Milenovic *et al.,* "A distributed blockchain-based broker for efficient resource provisioning in 5G networks," in *IEEE Int. Wireless Communications and Mobile Computing*, Limassol, Cyprus, pp. 1485–1490, 2020.
[16] S. Hakak, W. Z. Khan, G. A. Gilkar, N. Haider, M. Imran *et al.,* "Industrial wastewater management using blockchain technology: Architecture, requirements, and future directions," *IEEE Internet of Things Magazine*, vol. 3, no. 2, pp. 38–43, 2020.
[17] S. Hakak, W. Z. Khan, G. A. Gilkar, M. Imran and N. Guizani, "Securing smart cities through blockchain technology: Architecture, requirements, and challenges," *IEEE Network*, vol. 34, no. 1, pp. 8–14, 2020.
[18] R. Azzi, R. Chamoun and M. Sokhn, "The power of a blockchain-based supply chain," *Computers & Industrial Engineering*, vol. 135, no. 9, pp. 582–592, 2019.

[19] J. Leng, G. Ruan, P. Jiang, K. Xu, Q. Liu *et al.,* "Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: A survey," *Renewable and Sustainable Energy Reviews*, vol. 132, no. 10, pp. 110112–110132, 2020.

[20] N. Vincent, A. Skjellum and S. Medury, "Blockchain architecture: A design that helps CPA firms leverage the technology," *International Journal of Accounting Information Systems*, vol. 38, no. 3, pp. 100466–100479, 2020.

[21] X. Liu, W. Wang, H. Guo, A. Barenji, Z. Li *et al.,* "Industrial blockchain based framework for product lifecycle management in industry 4.0," *Robotics and Computer Integrated Manufacturing*, vol. 63, no. 2, pp. 101897– 101911, 2020.

[22] D. Bumblauskas, A. Mann, B. Dugan and J. Rittmer, "A blockchain use case in food distribution: Do you know where your food has been?," *International Journal of Information Management*, vol. 52, no. 9, pp. 102008– 102018, 2020.

[23] P. Helo and Y. Hao, "Blockchains in operations and supply chains: A model and reference implementation," *Computers & Industrial Engineering*, vol. 136, no. 3, pp. 242–251, 2019.

[24] A. Papathanasiou, R. Cole and P. Murray, "The (non) application of blockchain technology in the Greek shipping industry," *European Management Journal*, vol. 38, no. 6, pp. 927–938, 2020.

[25] K. Korpela, J. Hallikas and T. Dahlberg, "Digital supply chain transformation toward blockchain integration," in *Proc. of the 50th Hawaii Int. Conf. on System Sciences*, Hilton Waikoloa Village, Hawaii, pp. 1–10, 2017.

[26] G. Perboli, S. Musso and M. Rosano, "Blockchain in logistics and supply chain: A lean approach for designing real-world use cases," *IEEE Access*, vol. 6, pp. 62018–62028, 2018.

[27] R. Ausanka-Crues, *Methods for Access Control: Advances and Limitation*, Claremont, CA, USA: Harvey Mudd College, 2001.

[28] D. Ferraiolo, J. Cugini and R. Kuhn, "Role-based access control: Features and motivations," in *Proc. of the Annual Computer Security Applications Conf.*, Los Alamitos, California, IEEE Press, 1995.

[29] W. Rjaibi and P. Bird, "A multi-purpose implementation of mandatory access control in relational database management systems," in *Proc. of 13th Conf. on Very Large Databases*, Toronto, Canada, vol. 30, pp. 1010–1020, 2004.

[30] M. Markus and C. Tanis, "The enterprise system experience from adoption to success," *Framing the Domains of IT Management*, vol. 173, no. 2000, pp. 173–207, 2000.

[31] D. R. Wong, R. Daniel, S. Bhattacharya and A. Butte, "Prototype of running clinical trials in an untrustworthy environment using blockchain," *Nature Communications*, vol. 10, no. 1, pp. 1–8, 2019.

[32] M. Niranjanamurthy, B. N. Nithya and S. Jagannatha, "Analysis of blockchain technology: Pros, cons and SWOT," *Cluster Computing*, vol. 22, no. 6, pp. 14743–14757, 2019.

[33] K. Hendricks, V. R. Singhal and J. K. Stratman, "The impact of enterprise systems on corporate performance: A study of ERP, SCM and CRM system implementations," *Journal of Operations Management*, vol. 25, no. 1, pp. 65–82, 2007.

[34] M. Trotta, "Product lifecycle management: Sustainability and knowledge management as keys in a complex system of product development," *Journal of Industrial Engineering and Management*, vol. 3, no. 2, pp. 309–322, 2010.

[35] H. Hassan and S. Bin Nashwan, "Impact of customer relationship management (CRM) on customer satisfaction and loyalty: A systematic review," *Research Journal of Business Management*, vol. 6, no. 1, pp. 86–107, 2017.

[36] S. Shang and P. Seddon, "Assessing and managing the benefits of enterprise systems: The business manager's perspective," *Information Systems Journal*, vol. 12, no. 4, pp. 271–299, 2002.

[37] X. Min, Q. Li, L. Liu and L. Cui, "A permissioned blockchain framework for supporting instant transaction and dynamic block size," *IEEE TrustCom/BigDataSE/ISPA*, vol. 1, pp. 90–96, 2016.

[38] Y. Wang and A. Kogan, "Designing confidentiality preserving blockchain-based transaction processing systems," *International Journal of Accounting Information Systems*, vol. 30, no. 5, pp. 1–18, 2008.

[39] I. Zikratov, A. Kuzmin, V. Akimenko, V. Niculichev and L. Yalansky, "Ensuring data integrity using blockchain technology," in *20th Conf. of Open Innovations Association*, St. Petersburg, Russia, pp. 534–539, 2017.

[40] S. DiRose and M. Mansouri, "Comparison and analysis of governance mechanisms employed by blockchain-based distributed autonomous organizations," in *13th Annual Conf. on System of Systems Engineering*, Paris, France, pp. 195–202, 2018.

[41] M. Muzammal, Q. Qu and B. Nasrulin, "Renovating blockchain with distributed databases: An open source system," *Future Generation Computer Systems*, vol. 90, no. Supplement C, pp. 105–117, 2018.

[42] K. Kuhi, K. Kaare and O. Koppel, "Ensuring performance measurement integrity in logistics using blockchain," in *IEEE Int. Conf. on Service Operations and Logistics, and Informatics*, Singapore, pp. 256–261, 2018.

[43] X. Liang, J. Zhao, S. Shetty and D. Li, "Towards data assurance and resilience in IoT using blockchain," in *IEEE Military Communications Conf.*, Baltimore, MD, USA, pp. 261–266, 2017.

[44] S. Zhang and J. H. Lee, "Analysis of the main consensus protocols of blockchain," *ICT Express*, vol. 6, no. 2, pp. 93–97, 2019.

[45] M. Salimitari and M. Chatterjee, "A survey on consensus protocols in blockchain for IoT networks," 2018. [Online]. Available: https://arxiv.org/abs/1809.05613.

[46] M. Liu, F. Yu, Y. Teng, V. Leung and M. Song, "Performance optimization for blockchain-enabled industrial internet of things (IIoT) systems: A deep reinforcement learning approach," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3559–3570, 2019.

[47] A. Fekete, "CAP Theorem," in *Encyclopedia of Database Systems*. Berlin, Germany: Springer, 2017.