Received: 09 January, 2023; **Accepted:** 28 June, 2023.
DOI. 10.30473/COAM.2023.66693.1224
Summer-Autumn (2023) Vol. 8, No. 2, (63-84)
Research Article

Open Access

Control and Optimization in Applied Mathematics - COAM

Blockchain Technology in Optimizing Logistics Information Security in Business Process Technology Transfer Management

Seyed Saman Karimi[1], **Tahmoures Sohrabi**[2*], **Amir Bayat Tork**[2]

[1]Department of Technology Management, Central Tehran Branch, Islamic Azad University, Tehran, Iran.
[2]Department of Industrial Management, Central Tehran Branch, Islamic Azad University, Tehran, Iran.

Correspondence:
Tahmoures Sohrabi
E-mail:
tah.sohrabi@iauctb.ac.ir

How to Cite
Karimi, S.S., Sohrabi, T., Bayat Tork, A. (2023). "Blockchain technology in optimizing logistics information security in business process technology transfer management", Control and Optimization in Applied Mathematics, 8(2): 63-84.

Abstract. This paper discusses the challenges facing the logistics industry in the global business environment, including issues related to tracking transactions, preserving transaction privacy, and ensuring the security of logistics information. We propose a method for addressing these challenges using a blockchain-based system that employs a smart contract to control the behaviors of all participants in the logistics and trade process. In the experiment, we use the Solidity programming language to develop a smart contract on Ethereum and tested it for common logistics and transaction uses. The results of related programming and coding in Remix IDE show that the proposed algorithm is highly implementable. To test the smart contract code and validation, we test four main functionalities, which include successful collateral deposit after the customer requests a document, unique token generation, successful payment settlement, and a refund based on dispute handling by the arbitrator. Oyente vulnerability analysis also shows that the code does not suffer security bugs. Therefore, the proposed method can effectively decrease the risk of the logistics and trade process.

Keywords. Optimization, Business process management, Internet of things (IoT), Blockchain, Smart contract, Cryptocurrency.

MSC. 35Q93;49N30.

https://mathco.journals.pnu.ac.ir

1 Introduction

The word "Logistics" has Greek roots and refers to the transfer and movement of materials, money, services, and the monitoring and transfer of shipments in the industry. It is considered one of the pillars of business processes. Logistics is a complex process that involves many parties, including importers, exporters, freight forwarders, modes of transport, shipping companies, intermodal operators, surveyors, banks, tax officers, customs offices, brokers, shipping agencies, and insurance companies [18]. Additionally, the process requires a lot of communication, document preparation and the use of various systems for information sharing until the products reach the delivery point [7].

Because of the importance of these activities, it is natural to expect that increasing the security of logistics information will provide a large part of the security of business processes. Various methods have been proposed to increase the security of this information, one of which is the use of blockchain and IoT. This approach has gained popularity due to the increasing use of digital information. Blockchain technology has widespread applications in business process management and IoT-related systems. Business process refers to a set of relevant tasks aimed at achieving a delivery process of a service or product, and business process management involves the design, implementation, monitoring, and improvement of these processes [5, 26].

In these processes, it will be possible to benefit from blockchain technology at many points. Monitoring the transactions in the logistics sector, measuring the performance of processes, and planning future activities are important activities [8]. Blockchain provides a secure platform for tracking the origin of goods and their movement [7, 8, 28]. With blockchain, data collection and near real-time transmission of data is possible. This feature can be used to monitor the status of all logistics processes, predict the course of operations, and to make faster decisions based on reliable data [7, 14]. Despite the important features of blockchains, there are various challenges and concerns that hinder their the adoption in the global business environment. Various data and privacy concerns that impact the public image of the blockchain system, which in turn have led to regulatory constraints that limit their adoption in firms and business organizations [17, 39]. In the past, it has been proven that transaction privacy is not always protected by blockchain as transactions on public are accessible by the public [15]. Business using public blockchain can face serious issues if their data is not meant for public entirely. Another challenge with the technology is security. The adoption of blockchain by business has suffered due to its close association and identification with Bitcoin in the eyes of the government bodies and policy makers. Blockchain are infamously tied to various Bitcoin frauds and scandals that have taken place in the cryptocurrency market [23]. Therefore, solving these challenges and increasing the security of this new technology has become one of the most important concerns of experts.

Considering the potential of blockchain technology, this technology promises to review and improve business process management (BPM) and create new opportunities. However, as mentioned earlier, it also comes with challenges. Business process management is associated with the design, implementation, monitoring, and improvement of business processes. This structured approach aims to improve the quality of products and services and keep the organization's processes aligned with its strategy, ultimately pursuing the greater goal of customer satisfaction [32]. BPM has a sequential workflow, including automation, monitoring, and analysis. There are different distributions of the lifecycle, including the analysis, design, implementation, monitoring, and adaptation. The lifecycle of BPM may vary in an organization or a program, but the core processes of analysis, design, execution, and implementation cannot be removed. This lifecycle provides the possibility of utilizing the management system.

This study aims to answer the question of whether a model can be developed to improve the security of logistics information using Solidity programming language. To address this research question, the following steps have been taken: First, a brief introduction is provided in the opening Section 2. Second, the research background and literature review are discussed. In Section 3, the research problem is presented. The use of blockchain technology in the traditional life cycle of business process management is discussed, including the eight stages. Finally, the research findings are presented in Section 4, followed by a discussion and conclusions in Section 5.

2 Research Background and Literature Review

The concept of blockchain was first introduced with the emergence of the Bitcoin digital currency, and it was utilized to save user information. A blockchain consists of a continuous list of blocks, each containing a set of transactions. Cryptography is the main feature of each block in the chain, and each block is responsible for saving some type of information. Blockchain was first introduced in 2008 and implemented in 2009 [5, 26]. It can be considered a general ledger where all transactions are stored in a chain of blocks that continuously grows when new blocks are added. Blockchain can work in a decentralized space enabled by the integration of several core technologies, such as cryptographic hash, digital signature (based on asymmetric cryptography), and a distributed consensus mechanism. Transactions can be done in a decentralized way using blockchain technology. Therefore, blockchain can save costs and improve efficiency [4, 30].

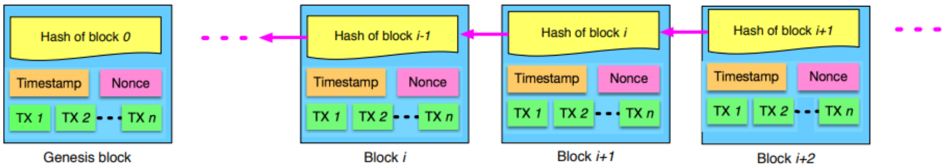
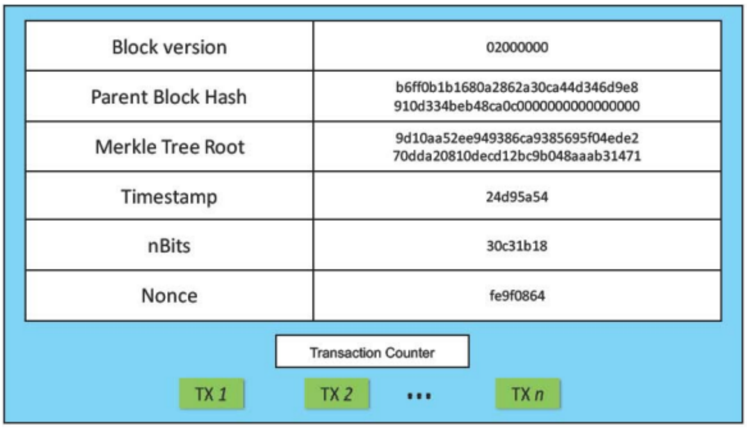


Figure 1: An example of a blockchain including a continuous sequence of blocks [4].

Figure 1 illustrates an example of blockchain, where each block points to the immediately-preceding block through a reference known as the parent block, which is essentially a hash value of the previous block. Hashing is used to maintain the connections between blocks in the blockchain. The process of hashing involves taking data of any size and passing it through a mathematical function to produce an output, which is known as a hash. A block contains a transaction counter and transactions, and the maximum number of transactions that a block can hold depends on the block size and the size of each transaction (as shown in Figure 2).

Business processes can benefit greatly from the combination of the Internet of Things (IoT) and blockchain. Blockchain-based IoT systems offer several advantages, including reducing the risk of a single point of failure, improving fault-tolerance capabilities, and enabling end-to-end communication



Block version	02000000
Parent Block Hash	b6ff0b1b1680a2862a30ca44d346d9e8 910d334beb48ca0c0000000000000000
Merkle Tree Root	9d10aa52ee949386ca9385695f04ede2 70dda20810decd12bc9b048aaab31471
Timestamp	24d95a54
nBits	30c31b18
Nonce	fe9f0864

Figure 2: The constituents of each block [4].

without the intervention of a single device. The tamper-proof data storage capability of the blockchain also allows for secure software updates for IoT devices, while storing data and event reports in an immutable way ensures traceability and accountability [21]. However, there are concerns regarding data corruption of sensor data stored in the blockchain, as explored by Saxena et al. (2021) [30]. To address this issue, sensors retrieving stored data from the blockchain (or cloud) must verify data generated by other sensors. Wang et al. proposed an approach to verify sensor data by implementing a collaborative certificate authentication protocol that uses the cache space on all IoT devices to create large memory storage for storing valid certificates. An IoT device retrieving data from the cloud must verify the data's authenticity before using it. The device retrieves the stored certificate corresponding to the retrieved data and asks the certificate producer (i.e., the owner) to verify it [36].

Fuchang Li chose Yunnan Matsutake as the research objective to design an optimization scheme for the routine logistics distribution problem. The optimization measures include conducting joint distribution mode, building a safe and reliable cold chain information-sharing platform, and setting up the distribution center, etc. [16]. In a related study, Zhongwei et al. analyzed the logistics distribution of fresh agricultural products during the development and quality safety process. Regarding the safe distribution model, Weidong Li [12] identified some safety problems in the distribution process, including the risk of cooperation between companies, the risk of product distribution and infrastructure, the risk of the distribution process and product value, and the risk of the uncontrolled external environment that affects logistics industry development. The author proposed the creation of a safety model in supply chain management to ensure the effectiveness of the distribution process.

Research on logistics security often involves studying secure transportation mechanisms for specific items, analyzing secure delivery models, and exploring logistics tracking. In terms of secure transport mechanisms for specific items, Xueyu Mi [19] developed an information system to optimize warehousing, storage operations, driven routes, and vehicle localization. The author also conducted studies on safe transport mechanisms and management of explosives for use in mines.

Business process management systems are software tools that used to manage the business process lifecycle, including modeling, implementation, and monitoring. However, each solution offers different features, and deciding which one is right for a particular situation can be challenging. Business process management software (BPMS) often provides a centralized and feature-rich solution to address the lack of trust and transparency between different parties in a given space. Blockchain technology has been applied in various fields to facilitate the operation of new business processes. For example, Nofer et al. [25] listed applications in the financial sector, including digital currency transactions, securities trading and settlement, and insurance, as well as non-financial applications such as notary services, music distribution, and various services to prove existence, authenticity, or storage. Other works describe application scenarios related to blockchain technology in logistics and supply chain processes, such as in the agricultural sector [31]. Large parts of the inter-organizational control flow and business logic of business processes can be converted from process models into smart contracts that ensure the joint process is implemented correctly. In other words, trigger can connect these inter-organizational process implementations to web services and internal process implementations, acting as a bridge between blockchain and organizational applications. The concept of cryptocurrency enables optional implementation of conditional payment and internal deposit management at defined points in the process where it is desirable.

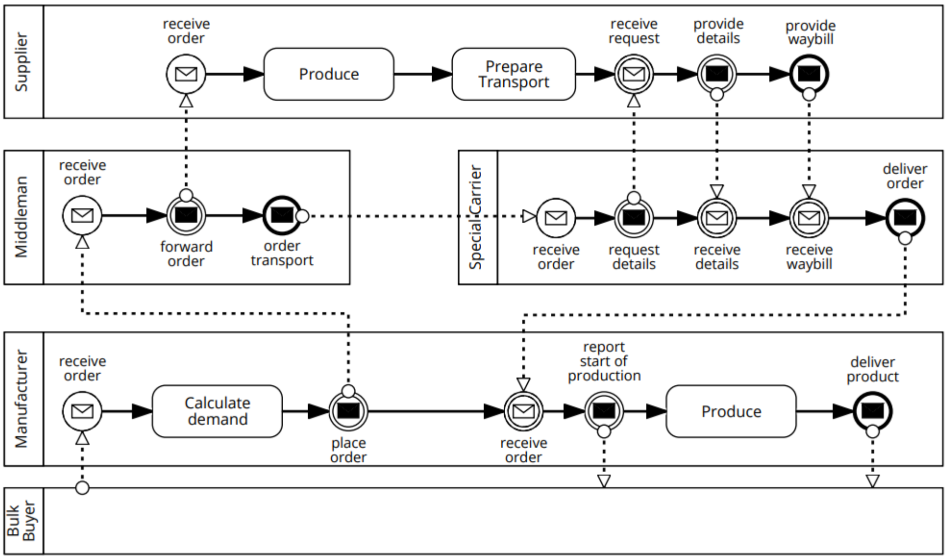
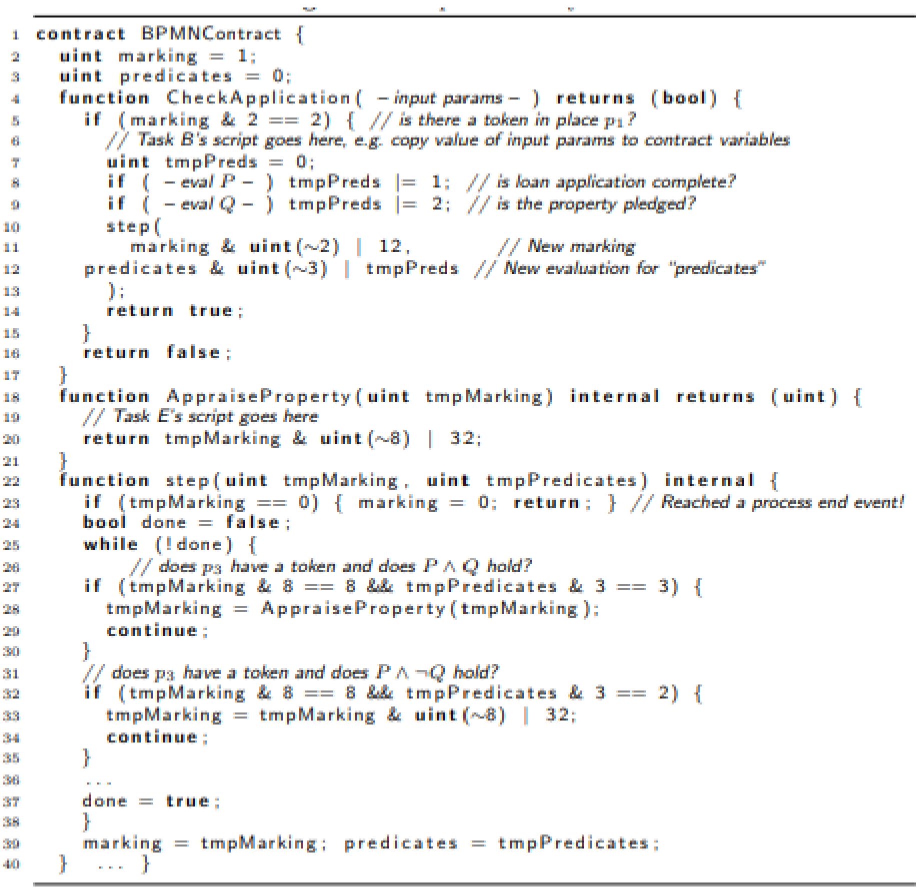


Figure 3: Supply chain scenario.

To demonstrate these capabilities, Figure 3 depicts a simplified supply chain scenario in which a buyer orders goods from a manufacturer. The manufacturer, in turn, orders the goods through an intermediary, which are then sent from the supplier to the manufacturer through a special carrier. Model-based approaches such as those presented by García-Bañuelos et al. [11], generate smart contract codes that implement the process, as shown in Figure 4.

The implementation of inter-organizational processes can be simplified by utilizing smart contracts on a blockchain. There are several benefits to using blockchain technology for managing business processes. Firstly, blockchain can act as an immutable public ledger, allowing participants to examine a reliable history of messages to determine the source of faults. In other words, all state change messages must be recorded on the blockchain. Secondly, smart contracts can provide independent monitoring of the process from a global perspective, ensuring that only expected messages are accepted if they are sent



```
contract BPMNContract {
  uint marking = 1;
  uint predicates = 0;
  function CheckApplication( – input params – ) returns (bool) {
    if (marking & 2 == 2) { // is there a token in place p1?
      // Task B's script goes here, e.g. copy value of input params to contract variables
      uint tmpPreds = 0;
      if ( – eval P – ) tmpPreds |= 1; // is loan application complete?
      if ( – eval Q – ) tmpPreds |= 2; // is the property pledged?
      step(
        marking & uint(~2) | 12,          // New marking
    predicates & uint(~3) | tmpPreds // New evaluation for "predicates"
      );
      return true;
    }
    return false;
  }
  function AppraiseProperty(uint tmpMarking) internal returns (uint) {
    // Task E's script goes here
    return tmpMarking & uint(~8) | 32;
  }
  function step(uint tmpMarking, uint tmpPredicates) internal {
    if (tmpMarking == 0) { marking = 0; return; } // Reached a process end event!
    bool done = false;
    while (!done) {
        // does p3 have a token and does P ∧ Q hold?
    if (tmpMarking & 8 == 8 && tmpPredicates & 3 == 3) {
      tmpMarking = AppraiseProperty(tmpMarking);
      continue;
    }
    // does p3 have a token and does P ∧ ¬Q hold?
    if (tmpMarking & 8 == 8 && tmpPredicates & 3 == 2) {
      tmpMarking = tmpMarking & uint(~8) | 32;
      continue;
    }
    ...
    done = true;
    }
    marking = tmpMarking; predicates = tmpPredicates;
  }  ... }
```

Figure 4: A fragment of a coded smart contract [11].

from a player registered for the corresponding role in the process instance. Lastly, encryption can ensure that only data that should be visible are made public, while the remaining data are only readable by the process participants who need it.

It is important to note that smart contracts are programs located on the Ethereum blockchain. These contracts enable the creation of decentralized programs that can continue to function indefinitely without any intermediaries, controls, or interruptions. Like traditional computer programs, Ethereum contracts are written in a programming language called Solidity. While other programming languages can be used to create smart contracts on the Ethereum network, Solidity is currently the preferred choice for most programmers in this field. Solidity is designed to be easily learned by programmers who have worked with one or more programming languages.

Smart contracts are comprised of two distinct and essential parts: data and code. By incorporating this specialized language, these contracts are executed automatically and without the involvement of a third party. The Ethereum blockchain has been designed with security and transparency in mind, which are important features provided by this language. The most significant use of the Solidity language is the linking of previous and subsequent blocks in the blockchain network to create a connected and chained set of blocks. Once this blockchain network is established, there is no chance of hacking or tampering with transactions, ensuring their security and integrity.

Blockchain technology can address the nonrepudiation problem, which is a crucial component of cybersecurity objectives such as confidentiality, integrity, availability, authenticity, authorization, and accounting (nonrepudiation). To protect against nonrepudiation issues, cryptographic methods such as digital signatures are commonly used. However, digital signatures require key management systems like Public Key Infrastructure (PKI), which are centralized. In this paper, we propose a decentralized or distributed system to address the nonrepudiation problem. Blockchain technology is a distributed system that guarantees security objectives, including accounting, through the use of cryptographic wallet addresses [6].

As blockchains continue to transform the way businesses operate and are managed, it is crucial to explore methods that can enhance the level of trust and security in this technology. By doing so, organizations can reap the benefits of blockchain sooner. This research aims to investigate how mathematical and programming techniques can be utilized to increase the security level of blockchains.

3 Problem Statement

BPM is a constantly evolving and broad discipline that encompasses various organizational roles, rules, tactics, and business goals. BPM has incorporated a range of optimization methodologies over the years, including Six Sigma, lean management, and Agile [33]. Despite the seemingly simple BPM Lifecycle, each phase can be time-consuming and requires careful planning, particularly since many business processes involve multiple departments and systems. According to one definition, the BPM lifecycle comprises three main phases [34]: (re)design, implement/configure, and run and adjust. The (re)design phase involves creating a new process model or adjusting an existing one. In the implement/configure phase, the model is configured into a running system, typically a Business Process Management System (BPMS). In the run and adjust phase, the processes are enacted and adjusted as needed using predefined controls [13]. Business process management systems (BPMS) are software tools that manage the business process lifecycle, providing an integrated solution approach for modeling, implementing, and monitoring business processes. Each solution approach offers different features, making it challenging to decide which one is best for a particular situation. As such, BPM can be seen as a combination of modern technologies, advanced software, and management theories.

BPMS often offers a centralized and feature-rich solution approach to address the lack of trust and transparency between different parties in a particular space. The traditional BPM lifecycle includes the following stages: Identification, discovery, analysis, redesign, implementation, monitoring, adaptation, and evolution. Using the traditional BPM lifecycle as a reference framework enables us to discuss numerous incremental changes that are likely to result from the adoption of blockchain technology.

Over the last three decades, several researchers have proposed solutions to address specific issues such as the challenges of smart contracts in BPMS [10, 19, 24, 29]. The introduction of Bitcoin and the blockchain network in 2009 provided a suitable environment for the development and introduction of smart contracts. Today, these contracts are widely used in commercial and economic transactions, and many people around the world are familiar with them. Smart contracts are implemented as coded

computer programs on the blockchain platform, adhering to the network's algorithms.
Smart contracts are continuously expanding the possibilities of what can be accomplished on the blockchain. These on-chain digital applications enable DAO voting, play-to-earn games, interactive NFTs, and economic and commercial contracts. However, popularity also makes them more attractive to hackers, and they are not infallible. Therefore, smart contract security should be a top priority when developing, deploying, or interacting with smart contracts.

Currently, BPMSs are not fully equipped to handle the critical dimension of process design and execution related to smart contract security. This paper addresses this gap in the literature by presenting a new approach for improving security supported by BPMS and tested in common logistics scenarios. To ensure logistics security at each stage, we propose employing blockchain technology to address logistics information security issues. We compare logistics business process based on blockchain technology with conventional logistics business processes to demonstrate our research ideas. We then introduced the design of smart contract as a proposed solution. Detail introductions are presented below.

3.1 Identification

Process identification involves a company's high-level description and evaluation of a process-oriented perspective, linking strategic alignment to process improvement. Identification is often studied from an introspective perspective [9]. Blockchain technology provides an additional relevant perspective for assessing high-level processes in terms of strengths, weaknesses, opportunities, and implicit threats. For instance, how can a company systematically identify the most appropriate processes for blockchains or those most at risk? There is a need for research on integrating this perspective into the identification phase. Since blockchains are closely linked to supporting inter-organizational processes, process identification may need to consider not only the needs of an organization but also those of known and even unknown partners.

3.2 Discovery

Process discovery refers to the collection of information about the current method of a process and its representation as a model of the process as it exists. Process discovery methods often rely on interviews, studies, and document analysis, which can be supplemented by automated process discovery techniques on unencrypted event reports generated by process-aware information systems [11]. Blockchain technology presents new challenges for process discovery methods. Information may be fragmented and encrypted. Accounts and keys can change frequently, while shipment data may be stored partially on-chain and partially off-chain. For instance, how can a company discover an overall process from blockchain transactions when they may not be logically linked to a process identifier? This fragmentation may require frequent alignment of information from all relevant parties active in the blockchain.

3.3 Evaluation

Process analysis involves gaining insight into issues related to the operation of a business process. Process analysis is often based on data available within organizations or from shared perceptions between internal and external process stakeholders. The history of the processes executed on the blockchain provides valuable information that can help evaluate file load, duration, frequency of routes, involved parties, and correlation between unencrypted data items. These pieces of information can be used to discover processes, detect deviations, and perform root cause analysis from small groups of companies to the entire industry. The question is, what effort is required to convert existing blockchain transaction data into a format that enables such analysis?

3.4 Redesign

Process redesign involves the systematic improvement of a process. Approaches such as redesign explorations are based on the assumption that there are recurring patterns of how a process can be improved [35]. Blockchain technology provides new ways to improve certain business processes or solve specific problems. For instance, instead of engaging a custodian to release payment when an agreed condition is met, a home buyer and seller may agree on a smart contract. The question is where blockchains can optimize existing interactions and where new interaction patterns can be created without a trusted central party that potentially uses insights resulting from relevant research on web service interactions [22].

3.5 Implementation

Process implementation refers to the process of converting a future model into software components that implement the business process. Business processes are often implemented using process-aware information systems or business process management systems within individual organizations. In this regard, the question is how the involved parties can ensure that the blockchain implementation supports their process ideally. Some challenges related to converting a process model into blockchain artifacts are discussed. Important engineering challenges in implementation include the identification and definition of abstractions for designing blockchain-based business process implementations. There is a need for libraries and operations for engines, along with basic modeling, and BPMN language extensions. Software patterns and anti-patterns greatly can significantly assist engineers in designing blockchain-based processes. New approaches to quality assurance, accuracy, and verification, as well as corresponding accuracy criteria, are also required.

3.6 Execution

Execution refers to the sampling of individual cases and the processing of their information using information technology. This implementation is facilitated by process-aware information systems or business process management systems. The actual execution of a process on a blockchain differs in several ways from traditional methods. During the execution of an instance, messages between participants need to be transferred to the smart contract as blockchain transactions. The resulting messages must then be observed from the blocks of a blockchain.

3.7 Monitoring

Process monitoring involves collecting process execution events, displaying them in an understandable manner, and creating alerts and escalations when there is undesirable behavior. Such process execution data is recorded by systems that support process execution. However, during the analysis phase, problems may arise in terms of data segmentation and encoding. For instance, data on the blockchain alone may not be insufficient to monitor the process and may require integration with local off-chain data. Once the tracing is done, each party can independently monitor the global view of the process. This provides a suitable basis for ongoing adaptation, review, and monitoring of service level agreements. Secondly, the monitoring data exchanged via the blockchain is used to verify whether a process instance adheres to the original process model and the contractual obligations of all involved process stakeholders. To this end, blockchain technology can be employed to store process execution data and transfer data among the process participants.

3.8 Adaptation and evolution

Runtime adaptability refers to the concept of changing a process during its execution. In traditional approaches, this can be achieved by allowing process participants to modify the model while it is being executed. Blockchain can enforce model adaptation, enabling participants to rely on a common model that is followed [37].

While there are many challenges [3, 27] in blockchain, this manuscript focuses on challenges posed by blockchain mechanisms in BPM and IoT concepts. The proposed lifecycle faces various challenges during its development, including scalability, security, complexity, cost, speed, and domain dependence.

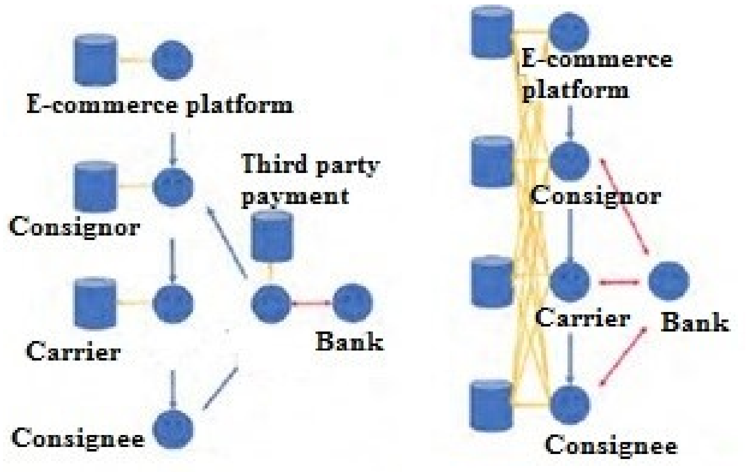
1- **Blockchain scalability** refers to the network's constant growth and large size.

2- **The cost of exchange transactions and network speed** is a challenge, particularly given the multiple aspects involved when using Bitcoin, such as storing information and ways of obtaining miner information or re-registering them.

3- **Blockchain complexity** has made cryptography mainstream, and recent studies support various types of glossaries and search indexes to make them easier to understand.

4- **Specific domain** structures underlie each blockchain, with their own initial rules and conditions. It is impossible to enter multiple data on different topics to cover different themes in each block, posing challenges in managing resources and data [1, 9].

Consider the drug use case scenario shown in Figure 5. In this cross-organizational process, a pharmacy places an order for medical supplies with its distributor, who then requests the manufacturer to produce the drugs. After manufacturing the drugs, they are delivered to the distributor, who then consigns them to the pharmacy.



(a) Conventional logistics business management process.

(b) Business management process based on logistics technology.

Figure 5: Comparative plan.

Implementing this process may lead to conflicts between different parties. For example, if the distributor fails to deliver the ordered drug on time, they may blame the manufacturer for the production delay, or they may challenge the date and time of receiving the original order. Therefore, using blockchain to record process transactions can be beneficial. Additionally, any organization can have full control over its private business process and share information about selected activities, including inter-organizational interactions, as shown in Figure 5. There are several desirable features of this approach. First, the parties involved this process do not need to agree on a common inter-organizational process. They may even be on different blockchain platforms as long as they are compatible. Second, the need for less transparency increases the parties' willingness to cooperate with each other. Third, this arrangement offers greater scalability because a pharmacy may deal with several distributors, and a distributor, in turn, may deal with several manufacturers. Therefore, this use case requires a more flexible, decentralized, connected, and distributed approach based on platform heterogeneity, both for two-way and multi-way interactions, which minimizes the need to interact with the blockchain platform.
We propose an integrated federated and blockchain BPMS architecture to address the issues identified above. The architecture should provide the following features:

- **Separation of concerns**: To minimize the performance impact on blockchain operations and maximize suitable capabilities for purpose of BPMS and blockchain platforms of BPMS and blockchain platforms, a clear separation of capabilities should be maintained between business logic operations and distributed transaction execution records.

- **Platform heterogeneity**: The architecture must allow the use of more than one adaptable blockchain platform within and across a hybrid set of interactive process instances.

- **Segmentation of interactions**: The requirement that all interactions between two participating parties be transparent to all parties should not be imposed. A blockchain-centric architecture may support it, for example, through the use of separate authorization channels, but it should not be considered a necessary realization, and still imposes the requirement that they must share the same blockchain platform.

- **One-way interaction**: The architecture must not assume that all interactions between a business process and a blockchain include multi-way communication. Hence, it should support simple one-way interaction between an organization's business process and its corresponding blockchain.

In our approach, each organization hosts a discrete BPMS that encapsulates a service or middleware through which it assigns tasks designed to perform a necessary inter-organizational activity, in a process execution instance. The service then interacts with a properly-configured blockchain network. Each service participating in an inter-organizational process is granted separate authorization of a channel (or other secure authentication pipelines) in the blockchain network. A channel is a private cover that classifies a blockchain network to create isolation and data confidentiality [26]. Whenever a new block is written, an event notification is generated by the blockchain platform and then sent to BPMS through the service. By default, this service listens to events as they occur, but it may be configured to periodically request event history from past blocks to consider those deployments in which connectivity to the blockchain network is not always available. Depending on the service configuration for each event, the service performs one of three actions for each received event notification: (1) release the task that waits for the event, (2) start a new process instance using the event as a trigger, or (3) ignore the event. Therefore, the only information exchanged between organizations is the information necessary to deliver and perform work in each organization, such as a purchase order, invoice, contract, and schedule. The status of a process instance can be inferred from its relevant data history in the blockchain, for example, an order, a shipment, and a payment. This eliminates the need to share additional information. Any exact state of the process on the blockchain or any process definitions, business logic, and rules, organizational data, or resource allocations must remain private to the organization. A transaction (e.g. a purchase order), sent to the blockchain by an organization is written in a block on the blockchain in a short period.

Once a blockchain transaction is verified by other peer nodes in the network using a validation algorithm, it is sorted into a block structure along with other transactions. The creation of a new block triggers an event notification that may be used by another organization to complete a task in one of its processes or to start a new process instance. The proposed approach's interior architecture is shown in Figure 5. An organization's BPMS assigns the execution of certain tasks to the blockchain service (middleware component) using the appropriate API along with the necessary data. The middleware sub-components include smart contract invocation, which executes a smart contract that requests the blockchain to query the current instance data written in the chain or request to create a new transaction to store data to share with another organization.

4 Findings

The present manuscript proposed a method for ensuring logistics information security based on blockchain technology, using smart contract to standardize and inhibit the behaviors of all participants in the logistics and trade process. In the test, we developed an experimental smart contract on Ethereum using Solidity and tested it for common logistics and transaction uses. Smart contracts are series of programming codes that run on the blockchain platform. The main advantage of using such contracts is that they eliminate the need for intermediaries and increase the security of information transfer. The security of such contracts is higher because the contract version is shared on a distributed network of all blockchain members, making its manipulation by an organization or a specific person is impossible. In the present manuscript, we used Solidity as the programming language, which is the usual programming language in smart contracts on the blockchain platform.

Figure 5 compares the logistics business process based on blockchain technology with conventional logistics business processes. Figure 5a shows the current conventional logistics business process, which includes various roles such as the e-commerce platform, consignor, carrier, third-party payment, bank, and receiver. The inclusion of third-party payment guarantees the security of the common plan in the logistics and trade process. The data generated throughout the process is stored in the database of the e-commerce platform, consignor, and carrier. However, given the technical capacity and business motivation, there are risks of data corruption, manipulation, and falsification. For instance, if the consignor fails to deliver the package, they can update the status to “delivered” without any mechanism to verify and limit such behavior information. Additionally, transaction security relies on third-party payment, which increases complexity and cost.

Figure 5b illustrates the logistics business processes based on blockchain technology. Data is stored in the distributed database of all participants using blockchain technology. Data can only be added, and cannot be deleted or modified, meaning that fake information is recorded forever. When smart contracts are used to control the entire business process, the whole operation is automatically completed by setting the rules when all businesses start, work, and end, without the need to pay a third party. This plan guarantees the security of logistics information and improves transaction efficiency. The smart contract design process of the present manuscript adopts the Zero Trust security principles. The identity and authority of all participants must be confirmed, and all behaviors are limited by the rules at each stage when the logistics order is established and initiated. Auto-receive defaults and other restrictions are defined, and conditional restrictions include the impossibility of adding data after the order is closed, etc. The identity of all roles must be verified, and the smart contract does not trust any host or IP address. Only authorized roles can invoke the smart contract function to add or read logistics data information. Therefore, the smart contract ensures that all data in the entire logistics business process are correct and recorded in the blockchain system according to the Zero Trust principle [37] when designing the contract. In testing, we utilized conventional and unusual use cases to test the performance of smart contracts. In testing conventional use cases, we mainly tested adding data and the automatic execution of the smart contract.

Below is the code and logic that has been designed:

```
contract Logistics{
// All participants in logistics process
```

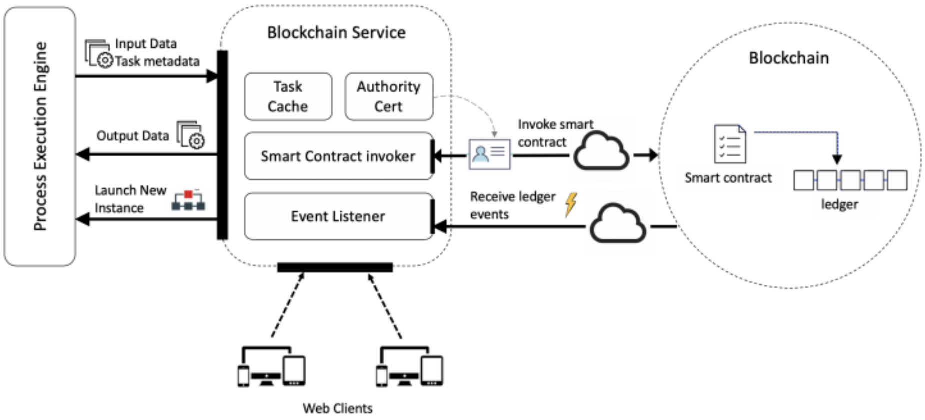


Figure 6: Conceptual interior architecture.

```
struct Role{}
// Basic order information
struct Order{}
// Simple logistics information
struct LogisticsInfo{}
// Init logistics order information
function initLogistics{}
// Init identity and authority
function initRole{}
// Consignor send package
function sendPackage{}
// Carrier collect package
function collect Package{}
// Carrier transport package
function transportPackage{}
// Carrier deliver package
function deliverPackage{}
// Consignee receive package
function receivePackage{}
// Judge when abnormal behavior
function judge{}
//Award when finish as expect
function award{}
// Rollback when trigger particular conditions
function rollback{}
// View the record of logistics
function viewRecord{}
```

```
// Automatic transfer money when order accomplish
function payment{}
}
```

The following steps were performed while testing the conventional use cases:

1- The initRole{} function is invoked to examine the validity of all participants.

2- The e-commerce platform invokes the initLogistics{} function to add order and logistics information to the blockchain databases.

3- The consignor invokes the sendPackage{} function to notify the carrier of package collection.

4- The carrier invokes the collectPackage{} function to collect the package.

5- The carrier invokes the transportPackage{} function at each stage when transporting the package.

6- The carrier invokes the deliverPackage{} function to notify the package consignee.

7- The consignee invokes the receivePackage{} function to confirm the package receipt.

8- The smart contract automatically invokes the payment{} function to transfer the consignee's money to the consignor's account.

9- The smart contract automatically invokes the award function{}.

In an unusual status, it is assumed that the delivery term is sent in exchange for not receiving the goods. The following steps is performed:

1- Invoking the initRole{} function is invoked to examine the validity of all participants.

2- The e-commerce platform invokes the initLogistics{} function to add the order and logistics information to the blockchain databases.

3- The consignor invokes the sendPackage{} function, but it is not actually sent the package.

4- The smart contract detects abnormal behavior and invokes the judge function() to judge the issue.

The smart contract invokes the rollback() function to return logistics information to the consignor.
The following steps are performed in an unusual scenario when the package is lost:

1- The initRole{} function is invoked to examine the validity of all participants.

2- The e-commerce platform invokes the Logistics{} function to add the order and logistics information to blockchain databases.

3- The consignor invokes the sendPackage{} function to notify the carrier of package collection.

4- The carrier invokes the collectPackage{} function to collect the package.

5- The carrier invokes the transportPackage{} function at each stage when transporting the package.

6- The smart contract detects that the carrier does not invoke the deliverPackage() function on time or does not add the missing package information by itself. It then invokes the judge() function to assess the situation based on the constraint conditions defined in the order information.

7- Based on the judgment result, the smart contract decides to either roll back the operation or terminate the process.

Based on the provided explanations, we will implement algorithm and conduct tests. The contract was created and tested using the Remix IDE [38]. Remix provides an environment for writing smart contracts in Solidity and for testing and debugging them. At the beginning of the work, after the customer and the file server pay the collateral, the smart contract automatically generates a unique token for the customer through the function *GenerateToken*(). The token is a hash created using the keccak 256 built-in function of Solidity. The hash is made using the following components: the customer's Ethereum Address (EA), Number of Successful Sales (NoSS), Number of Customers (NoC), Block Time stamp (BT), and Token Validity (1). The combination of those components ensures that the token is unique for each customer. Therefore, the token is generated as an event all participating entity as follows:

$$\text{TokenCustomer} = \text{keccak256}(\text{EACustomer}, \text{NoSS}, \text{NoC}, \text{BT}, 1).$$

Once created, the generated token is used by the customer for authentication when communicating with the file server to access the goods. Upon receiving the message from the customer, the file server verifies the signature, hash and received message components with the information received through the smart contract. After authenticating the customer, the file server replies back with a message containing the server's IP address, timestamp, server's public key as well as the EA of the server and customer. The purpose of this message is to authenticate the file server to the customer.

Therefore, the customer first requests the goods. When the contract receives a request from the customer, it indicates that the customer agreed to the terms and conditions of the contract and has already deposited the deposit collateral. The file server would then deposit the same amount as collateral. This action also informs the sender of the goods about the created request and gets them ready to enter the stage of transporting and sending the goods. Similarly, the recipient of the goods is also informed and the smart contract is called. This incentivizes every entity to act honestly and ensures that all participating entities are equal in their authority levels. The contract then automatically creates a unique token for the customer that has validity depending on the type of the goods requested.

To test the smart contract code and validate its functionality, four main functionalities are tested, including successful collateral deposit after a customer requests a document, unique token generation, a successful payment settlement and refund based on dispute handling by the arbitrator. The testing also involves multiple customers to demonstrate that the code is designed to handle multiple requests from different customers with different Ethereum addresses.
The Ethereum addresses of the owner, file server and arbitrator respectively are:

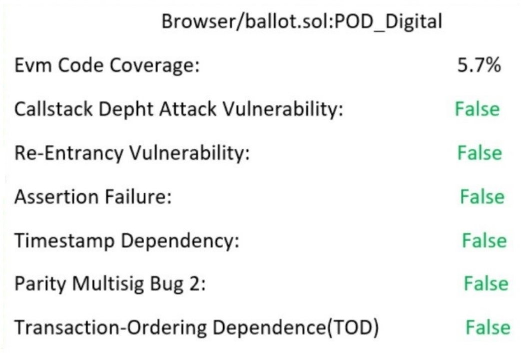
1) "0xca21b2d434412ef521ade3024dfe5f32e8f a621c",

2) "0x14723a09acff6d2a60dcdf7aa4aff308fddc670c",

3) "xcc140fa1b5c6722f2bd7f34112821c16f7789012".

Unless they are designed for customers with any Ethereum address, all functions can only be executed by specific entities with corresponding Ethereum addresses. As a result, all restricted functions with modifiers have undergone successful testing, and they are programmed to revert to their original state if an unauthorized caller attempts to execute them.

When writing code, it is crucial to ensure that it is free of bugs and is not vulnerable to security threats. This is especially critical when it comes to safeguarding smart contract code against vulnerabilities, risks, and attacks, as smart contracts deals directly with assets of high value [2]. A security attack on such a contract can result in severe financial loses. Therefore, it is important to thoroughly test the code and use essential security analysis tools to eradicate any security vulnerabilities. Oyente is an open-source tool used for analyzing smart contract code against known bugs and vulnerabilities. It is necessary to eliminate security holes that allow an adversary to manipulate the execution of a smart contract. Oyente works on the bytecode and not on the high level language such as Solidity [20]. Figure 7 shows the Oyente vulnerability analysis report on our smart contract code. None of the vulnerabilities that the tool checks for are "True". The results indicate that the code is not affected by security bugs, as evidenced by the "False" result displayed in green.



```
                 Browser/ballot.sol:POD_Digital
Evm Code Coverage:                                   5.7%
Callstack Depht Attack Vulnerability:                False
Re-Entrancy Vulnerability:                           False
Assertion Failure:                                   False
Timestamp Dependency:                                False
Parity Multisig Bug 2:                               False
Transaction-Ordering Dependence(TOD)                 False
```

Figure 7: Security vulnerability report.

Naturally, as the value of Ethereum increases, the use of this simple algorithm will become more cost-effective. However, it should be stated that the model used in this paper has been simplified to a certain degree and there is still much work to be done to apply the scheme in real business. Additionally, performance is a key factor when deploying in actual scenarios, which needs to be considered in the future.

Furthermore, since these contracts are valued and coded on the Ethereum platform, a question arises about what will happen to Ethereum and miners after the merge, and whether Ethereum mining will continue. The merge is a significant event for Ethereum and the entire cryptocurrency, which can be placed among the five biggest events in the history of digital currencies. There is a possibility that Ethereum mining will continue after the merge, especially if we witness the formation of a fork of Ethereum with a proof-of-work consensus mechanism. If a significant number of the miners want to continue working on it, the formation of this fork will be inevitable. Currently, people like Justin Sun (founder of Tron) and Chander Guo (a prominent Chinese miner) support the idea of an Ethereum network fork.

Contrary to the ideas that exist among the user community of the crypto world, the implementation of the upgrade phase to Ethereum 2 and the merge will not involve any noticeable changes for users. The main change in Ethereum 2 after the merge update is the switch from the proof-of-work algorithm to the proof-of-stake and the way nodes reach consensus will change. Therefore, although some changes will be required, it will still be possible to use this algorithm despite its simplicity.

5 Conclusions and Directions for Future Research

This paper presents a study of blockchain technology and its effect in business process management. Our solution and framework are generic enough that it can be used to orchestrate and govern the sale and delivery of any good or asset. Blockchain is an emerging technology for implementing decentralized applications. This technology is also useful in some other applications such as business process security. The introduction of consensus protocols, which allow handling manipulation and security issues along with tracing the sources of actions, is a key feature of the blockchain. The present manuscript in a special way examined the applications of such complex technology in the field of logistics information processes in business. for this purpose, it used solidity programming language to provide a model for improving the security of logistics information. This paper has presented a study on blockchain technology and its effect on business process management. Our solution and framework are generic enough to orchestrate and govern the sale and delivery of any goods or asset. Blockchain is an emerging technology for implementing decentralized applications, and is also useful in other applications, such as business process security. The introduction of consensus protocols, which allows for handling manipulation and security issues while tracing the sources of actions, is a key feature of the blockchain. This paper has examined the applications of such complex technology in the field of logistics information processes in business, using Solidity programming language to provide a model for improving the security of logistics information. Based on the Presented process, a smart contract was designed and introduced. In the experiment, a normal use case and two abnormal use cases were introduced to demonstrate the processes of smart contracts. The results of this experiment have shown that the designed smart contract can effectively ensure logistics information security and decrease the risk of logistics and trade processes. All transactions and interactions for sale and commodity sending are controlled by Ethereum smart contracts. The smart contracts were designed, implemented, and fully tested with various mechanisms and algorithms to automate payment in Ethereum, handles disputes, and set penalties to incentivize participants to act honestly. We have provided a security analysis in which we demonstrated that our smart contract code is safe and free of known exploitable security vulnerabilities and bugs. However, this research has faced some limitations. We have proposed, demonstrated, and evaluated a conceptual framework based on blockchain technology, but the effects of network speed and hardware resources were not evaluated. Thus, the experimental results cannot be used as a reference in real-life applications. The implementation of blockchain technology in our country (i.e., IRAN) faces serious limitations such as internet infrastructure, low speed, and poor quality of internet networks.

Basic reforms in software and hardware infrastructure are needed to be successful in this field. Regarding security issues and policy, blockchain technology is a tool to control and monitor the secured process only. Traceability is an essential characteristic of this technology. However, as explained in the previous section, it cannot prevent all security flaws. The policy needs to control and incorporate the blockchain's features to increase the completeness of security prevention. We introduce the security policy that is controlled by the proposed framework as follows:

i. The mistaken item problem is eliminated because the producer declares the items to the distributor. Then, the distributor confirms the type and number of items via our framework and notifies the customer as an alert message. This includes the details of the order to double-check the correctness.

ii. The package loss problem is prevented because the distributor confirms the item's correctness in the face of the producer. In addition, the customer verifies the items after receiving the order and confirming it with the distributor. The additional technique is taking a photo as evidence in case an item is lost. The IPFS technology is a tool for securing media files.

iii. The problem of fake orders or locations will be eliminated when cashless payment is used in this service. Payment with a credit card guarantees that fake order and location events are prevented. Moreover, blockchain technology introduces coins or coupons of the business. The service creates a secured coin or coupon to reduce the cash for the promotion or campaign. The coin will support the transaction fee payment between the producer, distributor, and platform. The cashless method will also reduce the fraud fee because nobody touches money in the system. In addition, all the transactions are recorded and monitored by the miner nodes in the blockchain network.

Here are some suggestions for future research.

i. Designing secure business processes for blockchains.

ii. Identifying barriers to blockchain adoption and exploring the securities services value network.

iii. Increasing the security and traceability of biological samples in biobanks using blockchain technology.

iv. Examining the impact of blockchain technology adoption on business performance in large enterprises.

v. Security challenges and exploring defense approaches for blockchain-based services.

vi. Investigating blockchain as a cutting-edge technology impacting business.

vii. Enhancing BIM security in emergency construction projects using lightweight blockchain-as-a-service Blockchain technologies.

vii. Developing a Raspberry Pi-based Blockchain application for IoT Security.

ix. Exploring a flexible approach to multi-party business process execution on blockchain.

x. Conducting a survey on the efficiency, reliability, and security of data query in blockchain systems.

xi. Exploring blockchain technologies for the interoperation of business processes in smart supply chains.

xii. Investigating blockchain technologies for the interoperation of for business management: Applications, challenges, and potentials of blockchain.

Declarations

Availability of supporting data

All data generated or analyzed during this study are included in this published paper.

Funding

This study received no funds, grants, or other financial support.

Competing interests

The authors declare no competing interests are relevant to the content of this paper.

Authors' contributions

The main manuscript text is collectively written by all authors.

References

[1] Adams, M., Suriadi, S., Kumar, A., Ter Hofstede, A.H.M. (2020). "Flexible integration of blockchain with business process automation: a federated architecture", In: Herbaut, N., La Rosa, M. (eds) Advanced Information Systems Engineering. CAiSE 2020. Lecture Notes in Business Information Processing, vol 386. Springer, Cham.

[2] Atzei, N., Bartoletti, M., Cimoli, T. (2017). "A survey of attacks on Ethereum smart contracts (SoK)", In: Maffei, M., Ryan, M. (eds) Principles of Security and Trust. POST 2017. Lecture Notes in Computer Science, vol 10204. Springer, Berlin, Heidelberg.

[3] Bae, H., Lee, S., Moon, I. (2014). "Planning of business process execution in business process management environments", Information Sciences, 268, 357-369.

[4] Bhushan, B., Sahoo, G. (2020) "Requirements, protocols, and security challenges in wireless sensor networks: An industrial perspective", In: Gupta, B., Perez, G., Agrawal, D., Gupta, D. (eds) Handbook of Computer Networks and Cyber Security. Springer, Cham.

[5] Burattin, A. (2015). "Introduction to business processes, BPM, and BPM systems", In: Process Mining Techniques in Business Environments. Lecture Notes in Business Information Processing, vol 207. Springer, Cham.

[6] Chukleang, T., Jandaeng C. (2022). "Security enhancement in smart logistics with blockchain technology: A home delivery use case", Informatics, 9, 70.

[7] DHL (2018). "Blockchain in logistics. perspectives on the upcoming impact of blockchain technology and use cases for the logistics industry", DHL Customer Solutions & Innovation, Germany.

[8] Dobrovnik, M., Herold, D.M., Fürst, E., Kummer, S. (2018). "Blockchain for and in logistics: What to adopt and where to start", Logistics, 2(3), 18, 1-14.

[9] Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A. (2013) "Fundamentals of business process management", Springer.

[10] Gao, Z. (2020). "When deep learning meets smart contracts' ", In Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering, Virtual Event, Australia, 1400-1402.

[11] García-Bañuelos, L., Ponomarev, A., Dumas, M., Weber, I. (2017). "Optimized execution of business processes on blockchain", BPM'17: International Conference on Business Process Management At: Barcelona, Spain.

[12] Guo, Zh., Zhang, Zh., Li, W. (2012). "Establishment of intelligent identification management platform in railway logistics system by means of the Internet of things", Procedia Engineering, 29, 726-730.

[13] Ilahi, L., Ghannouchi, S.A., Martinho, R. (2017). "BPFlextemplate: a business process template generation tool based on similarity and flexibility", International Journal of Information Systems and Project Management, 5(3), 67-89.

[14] Issaoui, Y., Khiat, A., Bahnasse, A., Ouajji, H. (2019). "Smart logistics: Study of the application of blockchain technology", Procedia Computer Science, 160, 266-271.

[15] Khan, M.A., Salah, K. (2018). "IoT security: Review, blockchain solutions, and open challenges", Future Generation Computer Systems, 82, 395-411.

[16] Li, F., Xiaohui, H. (2016). "25 Analysis of a dynamic inventory and transportation integrated optimization model of an online pharmaceutical supply chain based on a shared savings contract", Journal of Investigative Medicine, 64, A9.

[17] Lin, I.C., Liao, T.C. (2017). "A survey of blockchain security issues and challenges", IJ Network Security, 19(5), 653-659.

[18] Liotine, M., Ginocchio, D. (2020). "The supply blockchain: integrating blockchain technology within supply chain operations", Technology in Supply Chain Management and Logistics: Current Practice and Future Applications, 57-89.

[19] Liu, H., Sun, R., Zhao, G. (2018). "A method of logistics information security based on blockchain technology", 3rd Joint International Information Technology, Mechanical and Electronic Engineering Conference (JIMEC), Atlantis Highlights in Engineering, 3. 200-204.

[20] Luu, L., Chu, D.H., Olickel, H., Saxena, P., Hobor, A. (2016). "Making smart contracts smarter", In CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 254-269.

[21] Mendez Mena, D.M. (2021). "Blockchain-based security framework for the internet of things and home networks". Purdue University Graduate School. Thesis.

[22] Mendling, J., Weber, I., Van Der Aalst, W., Vom Brocke, J., Cabanillas, C., Daniel, F., Debois, S., Di Ciccio, C., Dumas, M., Dustdar, S. (2018). "Blockchains for business process management-challenges and opportunities", ACM Transactions on Management Information Systems, 9, 1-16.

[23] Monrat, A.A., Schel´en, O., Andersson, K. (2019). "A survey of blockchain from the perspectives of applications, challenges, and opportunities", IEEE Access, 7, 117134-117151.

[24] Montes, J.M., Ramirez, C.E., Gutierrez, M.C., Larios, V.M. (2019). "Smart contracts for supply chain applicable to smart cities daily operations", in 2019 IEEE International Smart Cities Conference (ISC2), 565-570.

[25] Nofer, M., Hinz, O., Muntermann, J., Roßnagel, H. (2014). “The economic impact of privacy violations and security breaches”, Business & Information Systems Engineering, 6(6), 339-348.

[26] Pal, A., Tiwari, C.K., Haldar, N. (2021). “Blockchain for business management: applications, challenges, and potentials”, Journal of High Technology Management Research, 32, 100414.

[27] Peters, G., Vishnia, G. (2016). “Overview of emerging blockchain architectures and platforms for electronic trading exchanges”, Available SSRN 2867344.

[28] Petersen, M., Hackius, N., Von See, B. (2018). “Mapping the sea of opportunities: Blockchain in supply chain and logistics”, IT-Information Technology, 60 (5-6), 263-271.

[29] Pettersson, J., Edstorm, R. (2016). “Safer smart contracts through type-driven development: Using dependent and polymorphic types for safer development of smart contracts”. Master’s Thesis in Computer Science.

[30] Saxena, S., Bhushan, B., Ahad, M.A. (2021). “Blockchain-based solutions to secure IoT: Background, integration trends and a way forward”, Journal of Network and Computer Applications, 181, 103050.

[31] Staples, M., Chen, S., Falamaki, S., Ponomarev, A., Rimba, P., Tran, A.B., Weber, I., Xu, X., Zhu, J. (2017). “Risks and opportunities for systems using blockchain and smart contracts”, Data61, CSIRO, Sydney.

[32] Tijan, E., Aksentijević, S., Ivanić, K., Jardas, M. (2019). “Blockchain technology implementation in logistics”, Sustainability, 11(4), 1185.

[33] Tucci, L. (2023). “What is business process management: An in-depth BPM guide”. TechTarget. Accessible at: https://www.techtarget.com/searchcio/definition/business-process-management#:~:text=BPM%20is%20a%20broad%20discipline,and%20lean%20management%20to%20Agile.

[34] Van Der Aalst, W.M. (2013). “Business process management: a comprehensive survey”, ISRN Software Engineering, 12, 1-37.

[35] Van Der Aalst, W. (2016). “Process mining: data science in action”, Springer.

[36] Wang, M., Qian, C., Li, X., Shi, S., Chen, S. (2020). “Collaborative validation of public-key certificates for IoT by distributed caching”, IEEE/ACM Transactions on Networking, 29, 92-105.

[37] Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., Mendling, J. (2016). “Untrusted business process monitoring and execution using blockchain”, In: La Rosa, M., Loos, P., Pastor, O. (eds) Business Process Management. BPM 2016. Lecture Notes in Computer Science, vol 9850. Springer, Cham.

[38] “Welcome to Remix Documentation”. [Online]. Available at: https://remix-ide.readthedocs.io/en/latest/

[39] Zheng, Z., Xie, S., Dai, H.N., Chen, W., Chen, X., Weng, J., Imran, M. (2020). “An overview on smart contracts: Challenges, advances and platforms”. Future Generation Computer Systems, 105, 475-491.