https://doi.org/10.33271/nvngu/2024-3/142

R. Sytnyk[1],
orcid.org/0000-0001-7820-9128,
Vik. Hnatushenko*[1,2],
orcid.org/0000-0001-5304-4144

1 – Ukrainian State University of Science and Technologies, Dnipro, Ukraine
2 – Dnipro University of Technology, Dnipro, Ukraine
* Corresponding author e-mail: vvitagnat@gmail.com

DATA FLOW MANAGEMENT IN INFORMATION SYSTEMS USING BLOCKCHAIN TECHNOLOGY

Purpose. Improving the process of information transfer for critical infrastructure sectors and enterprises through new approaches to real-time tracking of goods, services, and equipment, ensuring secure and transparent data integration and auditing of data flows in information systems using blockchain technologies.

Methodology. This research moves away from traditional centralized data management systems based on SQL and no-SQL databases by implementing a decentralized, immutable system built on blockchain technology. This uses the principles of the Merle tree in a digital ledger within blockchain technology to verify data integrity and smart contracts to automate key data flow processes. By tracking goods and equipment through supply chains on the blockchain, this approach ensures product authenticity, provenance, and transparency in real time. In addition, it creates a secure and transparent audit trail for all data in the system compared to conventional centralized data management systems based on SQL and no-SQL databases.

Findings. The developed blockchain-based approach improves data security, transparency, automation, and trust in managing data flows. Compared to traditional systems, it offers unique advantages such as immutability, decentralized management, and improved traceability. But while offering numerous advantages, blockchain also faces some limitations in terms of scalability and system complexity.

Originality. Digital ledger and blockchain methods have been further developed in the context of designing information systems and data flow management systems based on blockchain algorithms in the context of Industry 4.0. This allows increasing data security, transparency, automation, and trust in data flow management.

Practical value. The proposed approach is used to design information and data flow management systems based on blockchain algorithms. This improves the quality of data flow management in industrial enterprises and critical infrastructure, as well as supply chains.

Keywords: *Data flow, information systems, blockchain, industry 4.0, supply chains, critical infrastructure*

Introduction. The implementation and development of digitalization in various industrial sectors creates new challenges in building and designing information systems for industry. Incorporating modern technologies like artificial intelligence, the Internet of Things (IoT), cloud computing, big data analytics, and similar innovations results in the generation of substantial amounts of data. The term "Industry 4.0" was created to describe the fourth industrial revolution with the integration of advanced information technologies into manufacturing and other industries. It represents a shift in the way companies design, produce, and distribute goods and services, with a focus on greater efficiency, flexibility, and customization in today's economy. Thus, Industry 4.0 is characterized by the integration of digital technologies into production processes, which leads to a significant increase in the amount of data generated and exchanged between different components of the industrial ecosystem [1].

Nowadays modern enterprises of critical infrastructure are facing new challenges related to data flow management. With the increasing use of technology and Industry 4.0 approaches, and extraordinary challenges in handling critical infrastructure, there is a need for effective data flow management to ensure the reliability of these systems. By critical infrastructure, systems are meant which are considered essential for the functioning of a society and economy. For example, water supply, heating services, transportation, electricity generation, transmission and distribution, telecommunication, etc. Critical infrastructure can consist of multiple stakeholders which participate in generation or supporting services mentioned above.

One of the main challenges which critical infrastructure enterprises are facing is the sheer volume of data that needs to be managed. With the increasing use of sensors and other monitoring devices, there is a vast amount of data being generated by these systems on a daily basis. This data needs to be collected, stored, and analyzed in real-time in order to provide insights into system performance and identify potential issues.

Also, in addition to these challenges, there is a need for effective data sharing and collaboration between different departments and organizations. Many critical infrastructure enterprises are complex systems involving multiple organizations, each with their own data management systems and protocols. There is a need for standardized protocols and interoperability between these systems to ensure seamless data flow and collaboration.

Also, it is relatable to industrial sectors, where the growth of globalization continues making supply chains more complex and lengthier due to the increasing involvement of various countries, companies, and entrepreneurs in the creation and production of new goods with the diverse range of companies and countries involved in the process. Also, each of these entities can have its own set of standards, regulations, and procedures. Their logistics processes consist of many actions and transfers between suppliers and companies in different locations. And the more of them there are, the more complex supply chain management becomes, the data flows are larger and more confusing, and the overall process becomes less transparent for suppliers, customers, and end users of products and goods [2].

These challenges make the issue of data flow management in industrial sectors and enterprises of critical infrastructure relevant and actual.

The purpose of the work is to enhance information systems for critical infrastructure, enterprises, and other economy sectors via new approaches in real-time tracking of goods, services, and equipment, and providing a safe and transparent data & system integration & data flow audit in information systems using Blockchain & Digital Ledger technologies.

Related works. Many stakeholders are using a classical approach in building information systems for supply chains with usage of relational and non-relational databases, such as MySQL or MongoDB, in centralized servers, with an approach of client-side architecture. In MySQL, data is stored in tables that consist of rows and columns. Each table contains a unique set of columns that define the type of data that can be stored in each column. The structured query language (SQL) is used to create MySQL tables. MongoDB, in turn, is a non-relational database that uses a document-oriented approach to data storage. Unlike relational databases, where data is stored in the form of tables, MongoDB stores data in the form of JSON-like documents. Each document can have its own set of fields and values, and MongoDB can scale to store huge amounts of data. This data is accessed through a central server that hosts the database. The server handles requests from clients, computers that want to retrieve or modify data in the database, to retrieve and write data. This approach is called client-server architecture and is commonly used in information technologies [3].

And here it is how most information systems work: the central server hosts a database with a required structure. The client, according to client-server architecture, cannot get direct access to the database. Instead, the client makes a request to the server about retrieving/writing/deleting/editing some kind of information stored in the database on the central server. And the server decides, based on the information about the client and his request, should it be applied or not.

In the case of Industry 4.0, most entities of the supply chain have their own server with their own database. In some more rare cases, some industries can cooperate and have one common server for handling shared data in it.

But in large supply chain information systems and industrial systems, where there is a constant exchange of data between different systems and their clients for production and logistics interactions, a conventional client-server architecture may not be able to do its job efficiently.

This increased data flow creates a number of challenges related to data storage, processing and analysis, as well as data security and privacy [4]:

1. Lack of transparency and insufficient or unreliable information on the origin of products, difficulty in verifying the origin and originality of goods by the end consumer.

2. Trust, data security and privacy. As the amount of data created in Industry 4.0 increases, the risk of data leakage and cyberattacks increases. This poses a serious problem in terms of ensuring the security of the information system in the industrial ecosystem.

3. Efficiency in the interaction with other systems. Another challenge in scaling data flows in Industry 4.0 is the need for interaction between different components of the industrial ecosystem. For example, the interaction of manufacturers with suppliers, transportation companies, intermediaries, and other participants in the industrial ecosystem.

This requires the development of standardized data formats and protocols, as well as the deployment of software that works with these formats and ensures the integration of various systems and devices, which creates a seamless exchange of data between the various components of the ecosystem, enabling greater collaboration and efficiency. These challenges may prevent industry and business from fully realizing the benefits of Industry 4.0.

"Industry 4.0" is a concept of the industrial revolution based on the use of modern technologies and digital innovations in production processes. The introduction of the Industry 4.0 concept was intended to improve the competitiveness of European industry and increase productivity and product quality [5].

In order to implement information systems and improve the management of data flows that operate according to the concepts of Industry 4.0, this paper proposes to use blockchain technology.

Blockchain works as a distributed data structure which can be replicated and distributed among other network participants. The first blockchain specification was proposed along with the digital currency Bitcoin in 2008 by a person under the name or pseudonym Satoshi Nakamoto to solve the problem of centralization of finance around banks [6]. As a result of how nodes in Bitcoin (so-called miners) mutually verify agreed transactions in a centralized database, the Bitcoin blockchain identifies owners and indicates what they own (digital assets).

Currently, blockchains are used mainly in the field of decentralized finance (DeFi) in the form of cryptocurrencies and their tools in the form of exchanges, exchanges, auctions, banks, etc. There are also highly specialized studies from all over the glove on the use of blockchain in other areas of informatization besides finance. For example, the use of blockchain in the fishing industry [7].

Our previous work describes and prototypes the usage of smart-contract mechanisms in logistics and its benefits for tracking logistics actions [8].

There is also a study of the combination of blockchain with the Internet of Things for food safety monitoring, which describes the possibilities of monitoring products according to HACCP standards [9] but this work does not provide technical details on possible implementations of this system, as most of works on the topic of blockchain usage in information system, mostly covering economical & logistical questions and problems.

Proposed technique. To solve problems mentioned above, this paper proposes to adopt blockchain technology to manage data flows in supply chain information systems in Industry 4.0.

Blockchain is a distributed ledger technology that provides a secure and transparent way to store and share data. Blockchain works as a continuous sequential chain of blocks (a linked list) containing information built according to certain rules. The connection between the blocks is ensured not only by numbering, but also by the fact that each block contains its own hash amount and the hash amount of the previous block. Changing any information in a block will change its hash amount. To comply with the rules for building a chain, changes to the hash amount will need to be written to the next block, which will cause changes to its own hash amount [10].

The technological basis on which the blockchain is built is Distributed Ledger Technology (DLT). This technology is a distributed database that stores transaction records in multiple network nodes rather than in a single centralized database. Each node in the network has a copy of the data, and all nodes in the network synchronize their copies of the database. DLT does not have a centralized machine that controls the database, and all nodes have the same status.

DLT uses cryptography to ensure the security and reliability of the database. Each transaction is recorded in a block of the block chain (blockchain) and contains a hash of the previous block, which ensures the continuity of the chain. In addition, DLT provides the ability to create smart contracts that are automatically executed when certain conditions are met [11].

One of the main elements of DLT is a hash function, which is used to create a unique identifier for a block of data. The hash function can be represented by the following formula

$$H(x) = y,$$

where $H(x)$ is the hash function applied to data x; y is the resulting hash value.

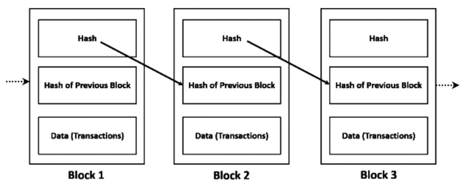

Fig. 1. Blockchain in structure

Another important element of DLT is the consensus protocol, which is used to reach agreement between network participants on the current state of the blockchain. Consensus protocols can be represented by various mathematical models, such as a decision-making or game model.

Consensus algorithms are used in blockchain technology to ensure that all nodes in the network agree on the state of the blockchain. The formula for a consensus algorithm may vary depending on the specific algorithm used, but it typically includes a set of rules that define how nodes in the network interact and agree on the state of the blockchain.

For example, the Ethereum blockchain switched from the Proof-of-Work (PoW) algorithm, which required an unecological process of "mining" blocks using mathematical calculations, to the Proof-of-Stake (PoS) algorithm.

In the PoS consensus algorithm, the probability that a node will be selected to create a new block is proportional to the amount of share it occupies in the network. This algorithm has been used in Ethereum since 2022 [12].

Formula for calculating the probability that a node will be selected as a block creator

$$P(node) = (S(node) / \sum S(i))^k,$$

where $P(node)$ is the probability that node will be chosen as a block creator; S(node) is the size of the share that node has, and $\sum S(i)$ is the total amount of the share that the network should have, the variable k is a constant that determines the level of randomness in the selection process and is usually set by the network protocol.

The Merkle Tree data structure is used to verify the integrity of transactions and data storage in the blockchain. It is named after Ralph Merkle. In a blockchain, the Merkle tree is used to ensure the integrity of the blocks that make up the blockchain. Each block contains a set of transactions, and the Merkle tree allows you to quickly check whether the transactions in the block have been changed [13].

To build a Merkle tree, each block is divided into several parts, for example, into the transactions that make up the block. Then each part is hashed using a cryptographic hash function, such as SHA-256, and the results are combined into pairs. Each pair is then hashed again, and this process is repeated until there is only one hash left, called the root hash of the Merkle tree.

The Merkle tree has many advantages in the blockchain, including quick verification of data integrity and not having to store all the data in each block. In addition, the Merkle tree can be used to verify transactions in a "light" blockchain client that does not have a full copy of the blockchain.

In Industry 4.0, the Merkle tree can be used to ensure data integrity in production systems, such as quality management systems or production process monitoring systems. It can also be used to ensure data security in supply chain management systems, for example, to authenticate documents and contracts.

This is a useful property of this data structure in handling data transactions in critical infrastructure information systems and industrial sectors.

To use a Merkle tree in a blockchain, you can use a hashing formula. Let us say we have a block of data that needs to be protected from changes and we want to use a Merkle tree for this purpose.

We start by hashing each individual data item in the block (e.g., transactions) to get their hash value. Then we combine the hash values in pairs and hash them together to get new hash values that are the roots of the subtrees of the Merkle tree.

This process is repeated until all the hash values are combined into one hash value, which is the root of the entire Merkle tree.

Mathematically, this can be represented by the following formula

$$H = H_n(H_{n-1}(H_{n-2}(...(H_2(H_1(T_a, T_b))...)T_x), T_z)),$$

where H is the final hash value of the root of the Merkle tree; H_n is the hashing function; T_{a-z} is a data element to be protected.

Thus, the Merkle tree allows you to effectively protect large amounts of data, ensuring the integrity of their storage and transmission. In a blockchain, the Merkle tree is used to ensure the integrity of blocks and confirm the correctness of transaction hashes, which is an important element of system security.

To authorize and further protect transactions in the blockchain, each transaction is additionally protected by a digital signature [14].

A digital signature is a mathematical method used to authenticate the identity of the sender of a message or transaction in DLT. Digital signatures are based on public key cryptography, created using a private key, and verified using a public key

$$Signature = sign(Hash(Data), PrivateKey),$$

where $Hash(Data)$ is the encrypted transaction data, *PrivateKey* is the private key of the user creating the transaction, sign() is the signature function according to the cryptographic algorithm used, Signature is the digital signature.

In systems with a large number of peers, it is important to maintain trust between each participant in the business process in order to organize high-quality management of data flows between enterprises in the Industry 4.0 ecosystem.

Experiment. In this paper, it is proposed to use smart contracts as a server for processing and storing enterprise data in the Industry 4.0 ecosystem. In this approach, all the logic of storing and managing data flows is performed directly on the blockchain network. The information system for monitoring the movement of goods in this paper is based on the specifications of the Ethereum blockchain network and the Ethereum Virtual Machine (EVM). In addition, the developed system can be implemented to work inside most existing blockchains, such as NEAR, Cardano, and others that support the execution of smart contracts inside their blockchains and finite state machines. Ethereum is the only decentralized blockchain-based virtual machine platform based on the operation of smart contracts. This means that Ethereum is a blockchain network with the ability to program the behavior of transactions using smart contract code [15].

Each smart contract in the Ethereum network is assigned a 20-byte address, which is its unique identifier. To write contracts in the Ethereum blockchain virtual machine, the Solidity programming language is used. This means that smart contracts can be used as a server for an information system to process and manipulate data.

Smart-contract proposed in this paper for handling supply

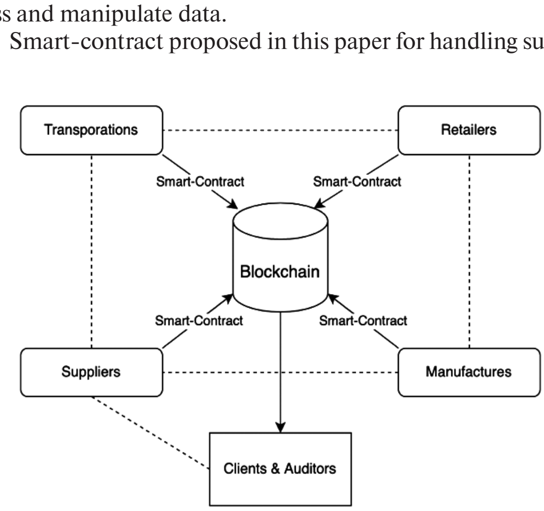

Fig. 2. Communication between entities in industrial information system based on Blockchain

chains in information systems has the following interface structure:

```
interface SupplyChain {
   struct Order {
   uint orderId;
   string productName;
   uint quantity;
   address supplier;
   address manufacturer;
   address transporter;
   address distributor;
   string deliveryAddress;
   string status; //With possible values: "ordered", "in_progress", "delivered", "received"
   string trackingNumber;
   uint256 deliveryTimestamp;
   }
   function addOrder(uint orderId, string memory productName, uint quantity, address manufacturer, string memory deliveryAddress) external;
   function updateProductionStatus(uint orderId, string memory status) external;
   function generateDeliveryOrder(uint orderId, address transporter, string memory trackingNumber) external;
   function updateDeliveryStatus(uint orderId, string memory status, uint256 deliveryTimestamp) external;
   function notifyDelivery(uint orderId, address distributor) external;
   function confirmReceipt(uint orderId, string memory status, uint256 deliveryTimestamp) external;
   function updateInventory(string memory productName, uint quantity) external view returns (uint);
   function getOrderStatus(uint orderId) external view returns (string memory);
}
```

It defines a structure for Order – model for a supply chain, which is processed within the information system.

Each role in the supply chain has their own functions in smart-contract, which should be secured from access from other roles by the implementation of the interface.

List of functions for each role:

Supplier: addOrder(uint orderid, string memory productName, uint quantity, address manufacturer, string memory deliveryAddress): This function adds a new order to the supply chain with the specified order ID, product name, quantity, manufacturer address, and delivery address.

Manufacturer: updateProductionStatus(uint orderId, string memory status): This function updates the production status of the order with the specified order ID to the specified status ("ordered", "in_progress", or "delivered").

Transporter: generateDeliveryOrder(uint orderId, address transporter, string memory trackingNumber): This function generates a delivery order for the order with the specified order ID, assigning the specified transporter and tracking number.

updateDeliveryStatus(uint orderid, string memory status, uint256 deliveryTimestamp): This function updates the delivery status of the order with the specified order ID to the specified status ("in_progress" or "delivered"), and sets the delivery timestamp.

Distributor: notifyDelivery(uint orderid, address distributor): This function notifies the distributor of the delivery of the order with the specified order ID.

confirmReceipt(uint orderid, string memory status, uint256 deliveryTimestamp): This function confirms receipt of the order with the specified order ID, setting the receipt status ("received" or "rejected") and the receipt timestamp.

Inventory Management: updateinventory(string memory productName, uint quantity): This function updates the inventory level for the specified product name by adding the specified quantity. It returns the updated inventory level.

getOrderStatus(uint orderid): This function gets the current status of the order with the specified order ID.

The proposed smart contract for handling supply chains in information systems utilizes a decentralized and blockchain network to automate and streamline the order fulfillment process from supplier to distributor. Here's a detailed explanation of how the system works:

1. *Order Creation:* The supplier initiates the order placement by calling the addOrder() function, specifying the order ID, product name, quantity, manufacturer address, and delivery address. This information is stored immutably on the blockchain.

2. *Production Status Updates:* The manufacturer updates the order's production status to either "ordered," "in_progress," or "delivered" by calling the updateProductionStatus() function. This ensures real-time visibility into the production process.

3. *Delivery Order Generation:* Once the order is ready for shipment, the transporter generates a delivery order using the generateDeliveryOrder() function, assigning themselves and generating a tracking number. This information is also recorded on the blockchain.

4. *Delivery Status Updates:* As the order progresses through the delivery process, the transporter updates its status to "in_progress" or "delivered" by calling the updateDeliveryStatus() function. The delivery timestamp is also recorded.

5. *Notification and Receipt Confirmation*: The transporter notifies the distributor of the order's arrival by calling the notifyDelivery() function, providing the order ID and their address. The distributor then confirms receipt and sets the receipt status ("received" or "rejected") using the confirmReceipt() function, along with the receipt timestamp.

6. *Inventory Management:* The distributor updates the inventory level for the delivered product by calling the updateInventory() function, adding the received quantity. The updated inventory level is returned.

7. *Order Status Tracking:* Anyone in the supply chain can retrieve the current status of an order by calling the getOrderStatus() function, providing the order ID. This provides real-time transparency and visibility across the supply chain.

A simulated supply chain environment will be created to replicate the real-world process of placing an order, manufacturing the product, delivering it to the distributor, and receiving it. The scenario will include multiple participants representing different roles in the supply chain, such as suppliers, manufacturers, transporters, and distributors.

The smart contract will be deployed on a local blockchain network. The smart contract will be configured to interact with the participants through APIs calls from Ganache development tool for interaction with local blockchains via Solidity programming language. Designed system is compared with a classical approach with MySQL database on centralized backend server API.

Results. The experiment was conducted using a simulated supply chain environment with blockchain and MySQL database with the same API as in Solidity smart-contract.

By automating these processes and maintaining an immutable record on the blockchain, the smart contract and system offer several benefits:

1. *Enhanced Supply Chain Visibility.* The tamper-proof and transparent nature of the blockchain ensured that all participants in the supply chain had access to real-time updates on order status. This improved visibility enabled stakeholders to make informed decisions, optimize logistics, and resolve issues more effectively.

2. *Reduced Fraud Risk*. The immutability of blockchain records made it difficult for malicious actors to tamper with or edit order information. This reduced the risk of fraud and errors, such as unauthorized order modifications or duplicate orders.

3. *Improved Traceability*. The system provided a complete and traceable record of an order's journey from supplier to distributor. This enhanced traceability enabled stakeholders to

track the movement of goods, identify potential bottlenecks or delays, and resolve issues proactively.

4. *Increased Collaboration.* The shared and transparent nature of the blockchain information fostered trust and collaboration among supply chain participants. This improved collaboration enabled stakeholders to work together more effectively to optimize the supply chain process.

Despite its potential benefits, the proposed smart contract and system for handling supply chains in information systems also have some limitations and challenges that need to be addressed:

1. *Scalability.* Blockchain networks have limited transaction processing capacity, which could become a bottleneck for large-scale supply chains with high order volumes. This can lead to delays and congestion, potentially negating the efficiency gains from automation.

2. *Cost.* Implementing and maintaining a blockchain-based supply chain system can be expensive, due to the cost of infrastructure, transaction fees, and ongoing development and maintenance. This can make it less attractive for smaller businesses or those with tight budgets.

3. *Complexity.* Integrating a blockchain-based system into existing supply chain infrastructure can be complex and time-consuming. This requires careful planning, coordination among stakeholders, and adaptation of existing processes to accommodate the new technology.

A brief summary of the results of the experiment with blockchain and MySQL database with the same API as in Solidity smart-contract can be seen on Table.

The smart contract for managing data flows in Industry 4.0 works by keeping track of the various stages of the data transactions, from the initial order placement with the supplier, to the delivery of the finished product to the distributor. It has several functions that allow the different parties involved in the supply chain to interact with it. The supplier can add information about the order, such as the quantity and expected delivery date, and the manufacturer can update the status of the production process and confirm when the product is ready for delivery. Similarly, the transporter can update the status of the delivery process and provide tracking information, while the distributor can confirm receipt of the product and update the inventory levels.

The smart contract also has various automated functions that are triggered by certain events. For example, when the manufacturer confirms that the product is ready for delivery, the smart contract automatically generates a delivery order and sends it to the transporter. The smart contract uses various mechanisms to ensure the security and integrity of the data and transactions. For example, it uses digital signatures to ensure that only authorized parties can make changes to the data, and it uses encryption to protect the data from unauthorized access.

Table

Comparison of blockchain-based and classical database-based information systems

Feature	Blockchain-based system	MySQL-based system
Centralization	Decentralized	Centralized
Transparency	Highly transparent	Limited transparency
Security	Highly secure and resistant to fraud	Susceptible to security breaches
Traceability	Provides a complete and traceable record of an order's journey	Limited traceability
Scalability	Some scalability issues	Scalable to handle large volumes of transactions, but centralized
Cost	Higher initial setup costs	Low cost
Complexity	More complex to implement	Easier to implement and maintain

If we look into the traditional centralized back-end server system with a client-server architecture and a central database, it requires separate tables and logic for all the actions. Meanwhile, the full access to the database is possible for the server administration, which creates trust issues and makes it possible to manipulate data. It reduces the security, transparency and trust for all stakeholders of the information systems for industrial sectors and critical infrastructure enterprises.

It can be partly solved by holding different information systems with own databases administrated by every stakeholder of the chain within the integrated system and they share required data between each other in order to make their work done. This approach creates additional complexity and more complicated data flow management within the system in the industry. But these issues, trust, transparency, and data security, are fixed by the blockchain-based information system, but this also creates some new types of complexity and scalability due to immutable nature of blockchain, both transactions and smart-contracts, and needing of consensus mechanism for writing transaction. But these issues should be considered and analyzed in every case-scenario for information system separately.

As for scalability, scalability is a critical issue for blockchain-based supply chain systems. As supply chains become more complex and involve more participants, the need for scalable solutions becomes increasingly important.

There are a number of scalability challenges to scaling blockchain-based supply chain systems. These challenges include:

1. *Transaction throughput:* Blockchains have a limited transaction throughput, which means that they can only process a certain number of transactions per second. This can become a bottleneck for large-scale supply chains with high order volumes.

2. *Consensus mechanisms:* Blockchains rely on consensus mechanisms to validate transactions and maintain the integrity of the ledger. These mechanisms can be computationally intensive, which can limit the scalability of the system.

3. *Storage capacity:* Blockchains store all transactions and data on the blockchain, which can require a significant amount of storage space. This can be a challenge for large-scale supply chains with a lot of historical data.

Mentioned challenges above need to be addressed to fully realize the potential of blockchain technology in supply chain management. Continued research and development are needed to improve scalability, reduce costs, simplify integration.

Discussions. In the work an analysis and comparison of a blockchain-based information system with a traditional information system for industrial sectors and critical infrastructure enterprises, based on centralized server with a database and application interface to it was conducted.

The proposed smart contract and system for handling supply chains in information systems has the potential to revolutionize supply chain management by significantly improving efficiency, transparency, and security. However, it is important to recognize the challenges that need to be addressed before this technology can be widely adopted, such as scalability, cost, complexity, regulatory compliance, security, and data reliability.

Based on the findings, the following recommendations are made for further enhancements and potential areas for wider adoption of blockchain-based solutions in supply chain management:

1. *Invest in infrastructure:* Enhance blockchain network capacity to handle larger transaction volumes and accommodate more participants.

2. *Develop standardized protocols:* Establish industry-wide standards for blockchain-based supply chain solutions to ensure interoperability and compatibility.

3. *Address regulatory concerns:* Work with regulators to develop clear guidelines for the use of blockchain technology in supply chain management.

4. *Educate stakeholders:* Raise awareness among supply chain participants about the benefits and potential risks of blockchain technology.

5. *Develop standardized protocols:* Establish industry-wide standards for blockchain-based supply chain solutions to ensure interoperability and compatibility.

By implementing these recommendations, organizations can further leverage the power of blockchain technology to transform their supply chains into more efficient, transparent, and secure operations.

Continued research and development are needed to address these challenges and make blockchain-based supply chain solutions more practical and cost-effective. As the technology matures and becomes more widely adopted, we can expect to see a significant transformation in the way supply chains operate, leading to improved efficiency, transparency, and security for all stakeholders.

Conclusions. As a result of the work, information systems for critical infrastructure sectors and enterprises were enhanced via new approaches in real-time tracking of goods, services and equipment, and providing a safe and transparent data & system integration & data flow audit in information systems using Blockchain & Digital Ledger technologies. With Blockchain and Digital Ledger unique properties of immutability, transparency, and decentralized governance offer a transformative approach to enhancing data security, tracking goods and services, and facilitating secure data integration and auditing.

Advantages of using proposed approach consist of the following aspects:

1. *Data Security*. In terms of data flow management, blockchain-based systems are inherently more secure than traditional systems. The use of a distributed ledger and consensus algorithms can prevent unauthorized access, tampering, and data breaches.

2. *Transparency*. Blockchain-based systems provide greater transparency and traceability in the terms of data flow management. Each transaction is recorded on the blockchain, enabling all parties to see the entire history of the transaction and have full control over analyzing suspicious activity.

3. *Automation*. Smart contracts can automate many processes in data flow inside of information systems reducing the need for manual intervention and increasing efficiency.

4. *Trust*. In terms of data flow management, blockchain-based systems can increase trust between parties in the supply chain, as all parties have access to the same information and can verify the authenticity of data.

5. *Decentralization*. Systems based on blockchain algorithms are decentralized, what means that no single party has control them over the network. This can help to prevent centralization and monopolies in the data flow management.

But while blockchain technology has many potential advantages for data flow management in Industry 4.0, there are also several disadvantages to consider in comparison with classic back-end systems used in industry, business, and critical infrastructure. The main disadvantages include:

1. *Scalability*. Blockchain technology is limited in terms of its scalability and ability to handle large volumes of data in terms of data flow management. The process of validating transactions through consensus mechanisms can slow down the system and make it less efficient than traditional information systems.

2. *Complexity*. Blockchain technology can be complex and difficult to understand and implement, especially for organizations without a deep understanding of distributed systems, cryptography, and decentralized networks. Also, it could be difficult to upgrade software based on blockchain due to immutable nature of blockchains.

This work is useful for designing information systems and data flow management systems based on the use of blockchain algorithms.

References.

1. Ghobakhlo, M. (2020). Industry 4.0, digitization, and opportunities for sustainability. *Journal of cleaner production, 252,* 119869. https://doi.org/10.1016/j.jclepro.2019.119869.

2. Okeagu, Ch. N. (2021). Principles of supply chain management in the time of crisis. *Best Practice & Research Clinical Anaesthesiology, 35*(3), 369-376. https://doi.org/10.1016/j.bpa.2020.11.007.

3. Győrödi, C., Győrödi, R., Pecherle, G., & Olah, A. (2015). A comparative study: MongoDB vs. MySQL. In *2015 13th International Conference on Engineering of Modern Electric Systems (EMES)* (pp. 1-6). IEEE. https://doi.org/10.1109/EMES.2015.7158433.

4. Dutta, P. (2020). Blockchain technology in supply chain operations: Applications, challenges and research opportunities. *Transportation research part e: Logistics and transportation review, 142,* 102067. https://doi.org/10.1016/j.tre.2020.102067.

5. Lasi, H., Fettke, P., Kemper, H.G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business & information systems engineering, 6*, 239-242. https://doi.org/10.1007/s12599-014-0334-4.

6. Nakamoto, S. (n.d.). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Retrieved from https://bitcoin.org/bitcoin.pdf.

7. Mathisen, M. (2018). The Application of Blockchain Technology in Norwegian Fish Supply Chains. A Case Study. *Norwegian University of Science and Technology.* Retrieved from https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2561323.

8. Sytnyk, R., Hnatushenko, Vik., & Hnatushenko, V. (2022). Decentralized Information System for Supply Chain Management Using Blockchain. *IntelITSIS 2022: Proc. of the 3rd Intern. Workshop on Intelligent Information Technologies & Systems of Information Security, March 23–25, 2022.* Khmelnytskyi, Ukraine, 3156, 587-598. Retrieved from https://ceur-ws.org/Vol-3156/paper45.pdf.

9. Tian, F. (2016). An agri-food supply chain traceability system for China based on RFID & blockchain technology. *WU Vienna University of Economics and Business.* https://doi.org/10.1109/ICSSSM.2016.7538424.

10. Yaga, D. (2019). Blockchain technology overview. *arXiv preprint arXiv:1906.11078.* https://doi.org/10.48550/arXiv.1906.11078.

11. El Ioini, N., & Pahl, C. (2018). A review of distributed ledger technologies. *On the Move to Meaningful Internet Systems. OTM 2018 Conferences.* https://doi.org/10.1007/978-3-030-02671-4_16.

12. Kapengut, E., & Mizrach, B. (2023). An event study of the ethereum transition to proof-of-stake. *Commodities, 2*(2), 96-110. https://doi.org/10.3390/commodities2020006.

13. Dhumwad, S., Sukhadeve, M., Naik, C., Manjunath, K.N., & Prabhu, S. (2017). A peer to peer money transfer using SHA256 and Merkle tree. *23 Annual International Conference in Advanced Computing and Communications (ADCOM).* IEEE, 2017. https://doi.org/10.1109/ADCOM.2017.00013.

14. Fang, W. (2020). Digital signature scheme for information non-repudiation in blockchain: a state of the art review. *Journal on Wireless Communications and Networking, 56*(2020). https://doi.org/10.1186/s13638-020-01665-w.

15. Dannen (2017). *Introducing Ethereum and solidity*, (1). Berkeley: Apress. https://doi.org/10.1007/978-1-4842-2535-6.

Управління потоками даних в інформаційних системах із використанням технології блокчейн

Р. С. Ситник[1], *Вік. В. Гнатушенко*[*1,2]

1 – Український державний університет науки і технологій, м. Дніпро, Україна

2 – Національний технічний університет «Дніпровська політехніка», м. Дніпро, Україна

[*] Автор-кореспондент e-mail: vvitagnat@gmail.com

Мета. Удосконалення процесу передачі інформації для секторів і підприємств критичної інфраструктури завдяки новим підходам до відстеження товарів, послуг та обладнання в режимі реального часу, забезпеченню безпечної та прозорої інтеграції даних й аудиту потоків даних в інформаційних системах із використанням технологій блокчейн.

Методика. Це дослідження відходить від традиційних централізованих систем управління даними на основі баз даних SQL і no-SQL, упроваджуючи децентралізовану, незмінну систему, побудовану на технології блокчейн. При цьому використовуються принципи дерева Мерля в цифровому реєстрі в рамках технології блокчейн для перевірки цілісності даних і смарт-контрактів для автоматизації ключових процесів потоку даних. Відстежуючи товари та обладнання через ланцюги поставок на блокчейні, цей підхід забезпечує автентичність продукції, її походження та прозорість у режимі реального часу. Крім того, він створює безпечний і прозорий аудиторський слід для всіх даних у системі порівняно зі звичайними централізованими системами управління даними на основі баз даних SQL і no-SQL.

Результати. Розроблений підхід на основі блокчейну підвищує безпеку даних, прозорість, автоматизацію й довіру до управління потоками даних. Порівняно із традиційними системами, він дає унікальні переваги, такі як незмінність, децентралізоване управління й покращена відстежуваність. Але, пропонуючи численні переваги, блокчейн також стикається з деякими обмеженнями щодо масштабованості та складності системи.

Наукова новизна. Методи цифрового реєстру та блокчейну набули подальшого розвитку в контексті проєктування інформаційних систем і систем управління потоками даних, побудованих на алгоритмах блокчейну в умовах «Індустрії 4.0». Це дозволяє підвищити безпеку даних, прозорість, автоматизацію й довіру до управління потоками даних.

Практична значимість. Запропонований підхід використовується задля проєктування інформаційних систем і систем управління потоками даних, побудованих на основі алгоритмів блокчейну. Це підвищує якість управління потоками даних на підприємствах промисловості та критичної інфраструктури, а також ланцюгами поставок.

Ключові слова: *потік даних, інформаційні системи, блокчейн, індустрія 4.0, ланцюги поставок, критична інфраструктура*

The manuscript was submitted 05.01.24.