A Blockchain Based Design for Maritime Transportation Systems

Yuhao Zhang[1], Jiaqi Wu[1], Haibo Yi[1,*], Weiping Deng[2], Tao Qu[3]

[1]School of Artificial Intelligence, Shenzhen Polytechnic, Shenzhen, China

[2]WeBank, Shenzhen, China

[3]Guangzhou Urban Blockchain Industry Association, Guangzhou, China

*Corresponding author

Keywords: Blockchain, Maritime Transportation Systems, GPS, transportation process

Abstract: Please describe the main issues in the supply chain. Designing a blockchain-based maritime transportation system involves the following steps: Identify the requirements and stakeholders involved in the transportation process, including shippers, freight forwarders, carriers, ports, customs, and regulatory authorities. Determine the scope of the system, including the type of cargo to be transported, the routes, and the mode of transportation (such as container ships or bulk carriers). Develop a smart contract-based system that can automate key functions, such as cargo tracking and payment processing, while ensuring transparency and security of data. Define the data structure and protocols for recording all relevant information, such as cargo ownership, shipment details, vessel schedules, and port logistics. Implement the blockchain platform, which could be a public or private network, depending on the security and accessibility needs of the system. Integrate the blockchain solution with existing transportation systems and infrastructure, such as RFID tags and GPS tracking devices. Conduct rigorous testing and validation of the system to ensure its efficiency, reliability, and compliance with legal and regulatory frameworks. Establish governance and maintenance procedures for the blockchain system, including roles and responsibilities of stakeholders, dispute resolution mechanisms, and system updates. Blockchain-based maritime transportation systems have several potential applications that can improve efficiency, security, and reliability in the industry.

1. Introduction

The development history of blockchain technology can be traced back to the early 1990s when researchers began exploring the concept of digital cash and cryptography-based systems for electronic payments. However, the modern blockchain technology that we know today was first introduced in 2008 by an individual or group of individuals using the pseudonym Satoshi Nakamoto in a whitepaper titled "Bitcoin: A Peer-to-Peer Electronic Cash System." [1]

The paper described a decentralized, public ledger system called a blockchain that could track and verify transactions without relying on a central authority like a bank or government. The transactions were verified and recorded by a network of computers called nodes, which used complex cryptographic algorithms to ensure the integrity and security of the system [2].

In January 2009, the first Bitcoin software was released, and the first Bitcoin was mined shortly thereafter. Bitcoin quickly gained popularity among computer enthusiasts and activists who valued the decentralization and anonymity that it offered. However, it also faced criticism from some quarters due to concerns about its use in illegal activities and the lack of regulation [3].

Over the years, other cryptocurrencies and blockchain-based applications have emerged, including Ethereum, Ripple, and Hyperledger Fabric. These platforms have expanded the capabilities of blockchain technology beyond simple financial transactions to include functions like smart contracts, supply chain management, and identity verification [4].

Today, blockchain technology is seen as having significant potential across a wide range of fields, from finance and insurance to healthcare and education. Its ability to provide secure,

  DOI: 10.25236/icmmct.2023.043

transparent, and decentralized record-keeping has led many experts to predict that blockchain could transform the way we conduct business and interact with each other in the years to come [5].

The blockchain network architecture is a decentralized, distributed system that uses cryptography and consensus mechanisms to maintain the integrity and security of a public ledger. Here's a detailed description of the various components of the blockchain network architecture:

Nodes: Nodes are the individual computers or devices that participate in the network. Each node stores a copy of the blockchain ledger and can verify transactions by performing cryptographic calculations.

Blockchain Ledger: The blockchain ledger is a digital record of all transactions that have occurred on the network. It is composed of blocks, each containing a set of transactions and a cryptographic hash linking it to the previous block. This creates an unbreakable chain of blocks that cannot be altered without invalidating the entire chain.

Consensus Mechanisms: Consensus mechanisms are used to ensure that all nodes on the network agree on the state of the ledger. Different consensus mechanisms, such as proof-of-work (used in Bitcoin) or proof-of-stake (used in Ethereum), require nodes to perform certain tasks, like solving complex mathematical problems or holding a certain amount of cryptocurrency, in order to add new blocks to the chain.

Smart Contracts: Smart contracts are self-executing programs that run on the blockchain network. They can be used to automate the execution of complex transactions, enforce rules and conditions, and create decentralized applications (dApps) [6].

Wallets: Wallets are digital tools that allow users to store and manage their cryptocurrency holdings. They typically contain a public key for receiving funds and a private key for authorizing transactions.

Mining: Mining is the process by which new blocks are added to the blockchain ledger. Miners use specialized hardware to solve complex mathematical problems and earn reward tokens for adding new blocks to the chain.

APIs: Application programming interfaces (APIs) are used to enable communication between different parts of the blockchain network and external applications. APIs allow developers to build new products and services on top of the blockchain network.

Overall, the blockchain network architecture is a complex and constantly evolving system that requires a range of different components to work together seamlessly. While there are many variations on this basic architecture, the decentralized, trustless nature of the blockchain network remains a defining characteristic of the technology.

The design method for the supply chain network involves a series of steps and considerations to create an efficient and effective system that meets the needs of the organization and its customers. Here's a general outline of the design process:

Define the Objectives: The first step is to establish clear objectives for the supply chain, such as reducing costs, improving delivery times, or increasing customer satisfaction.

Analyze the Supply Chain: Next, the supply chain must be analyzed to identify potential bottlenecks, areas for improvement, and opportunities to optimize operations. This may involve mapping out the existing supply chain processes, analyzing data on inventory levels and demand patterns, and conducting market research to understand customer needs.

Develop Alternative Designs: Based on the analysis, alternative designs for the supply chain can be developed. These may include changes to the sourcing strategy, transportation modes, inventory management, and distribution channels. Multiple options should be evaluated against the established objectives.

Evaluate and Select Design: Once the alternative designs have been developed, they can be evaluated using criteria such as cost, efficiency, reliability, and flexibility. The best option(s) are selected based on how well they meet the stated objectives.

Implement and Monitor: The selected supply chain design is then implemented, with a plan for monitoring and evaluating its effectiveness over time. This may involve ongoing data analysis, performance tracking, and adjustments as needed to ensure that the supply chain is meeting its objectives.

Throughout the design process, it's important to consider factors such as risk management, sustainability, and collaboration with suppliers and partners. Effective design and implementation of the supply chain network can result in improved operational efficiency, reduced costs, enhanced customer satisfaction, and increased overall profitability.

2. Relate Work

The design method for a blockchain-based supply chain network involves using decentralized ledger technology to create a transparent and secure system for tracking goods and information as they move through the supply chain. Here's a general outline of the design process [7-15]:

Define the Objectives: The first step is to establish clear objectives for the blockchain-based supply chain, such as increased transparency, reduced fraud, or improved traceability.

Analyze the Existing Supply Chain: Next, the existing supply chain must be analyzed to identify potential inefficiencies and areas for improvement. This may involve mapping out the existing supply chain processes, identifying key stakeholders, and analyzing data on inventory levels, demand patterns, and supplier performance.

Identify Processes to Be Decentralized: Based on the analysis, the processes that can benefit from decentralization using blockchain technology are identified. These may include tracking and verifying product origin, monitoring product quality and condition, managing contracts and payments, and sharing data across the supply chain.

Develop the Blockchain Network: Once the processes have been identified, the blockchain network can be developed. This involves designing the architecture of the blockchain network, selecting appropriate consensus mechanisms, developing smart contracts to automate supply chain processes, and integrating the blockchain with existing systems.

Test and Evaluate: The blockchain-based supply chain network should be thoroughly tested and evaluated to ensure that it meets the established objectives. This may involve running pilot programs or simulations, collecting feedback from key stakeholders, and analyzing data on performance.

Implement and Monitor: The blockchain network is then implemented, with a plan for monitoring and evaluating its effectiveness over time. Ongoing evaluation and adjustment may be necessary to ensure that the system is meeting its objectives and delivering the desired benefits.

Throughout the design process, it's important to consider factors such as data privacy, security, and scalability. Effective design and implementation of a blockchain-based supply chain network can result in improved operational efficiency, reduced costs, enhanced transparency, and increased trust among stakeholders.

Public chain, alliance chain, and private chain are three different types of blockchain networks that differ in terms of their degree of openness and accessibility. Here's a detailed comparison of the three:

Public Chain: A public chain is a decentralized blockchain network that is open to anyone and everyone. Anyone can participate in the network, create transactions and smart contracts, and become a validator or miner. Transactions are public and transparent, and consensus is achieved through mechanisms like proof-of-work or proof-of-stake. Example: Bitcoin, Ethereum.

Advantages:

Decentralized and transparent

No permission required to join

Robust security due to large number of nodes and validators

Disadvantages:

Slow transaction processing speed

High energy consumption due to proof-of-work consensus

Limited scalability due to constraints on block size and transaction throughput

Alliance Chain: An alliance chain is a semi-decentralized blockchain network that is formed by a group of organizations that collaborate to achieve shared goals. The network is permissioned, meaning that only authorized parties can participate as validators or nodes. Example: Hyperledger Fabric, R3 Corda.

Advantages:

Faster transaction processing speed compared to public chains

Greater flexibility in terms of governance and consensus mechanisms

Increased privacy and confidentiality for sensitive data

Disadvantages:

Less transparent compared to public chains

Requires trust among participating organizations

May be less secure than public chains due to a smaller number of nodes and validators

Private Chain: A private chain is a centralized blockchain network that is used within a single organization or consortium of organizations. The network is permissioned, and access is restricted to approved users and entities. Example: Quorum.

Advantages:

Faster transaction processing speed compared to public chains

Complete control over network governance and consensus mechanisms

Enhanced privacy and confidentiality for sensitive data

Disadvantages:

Limited scalability due to the centralized nature of the network

Requires trust among participating entities

May be less secure than public chains due to a smaller number of nodes and validators

In summary, as shown in Table 1, public chains offer the highest degree of openness and transparency but can be slow and energy-intensive. Alliance chains provide a compromise between openness and control, while private chains offer complete control over governance and privacy but are limited in scalability and require trust among participants. The choice of blockchain network type depends on the specific goals and requirements of the organization or consortium implementing it.

Table 1 Comparing of public chain, alliance chain, and private chain

Attribute	Public Chain	Alliance Chain	Private Chain
Access	Open to anyone	Permissioned	Permissioned
Governance	Decentralized	Semi-decentralized	Centralized
Consensus Mechanism	Proof-of-work, proof-of-stake	Various (e.g. practical byzantine fault tolerance)	Various (e.g. raft consensus)
Transaction Processing Speed	Slow	Faster than public chains	Faster than public chains
Scalability	Limited	More scalable than public chains, but still limited	Limited
Privacy	Public and transparent	Can offer increased privacy compared to public chains	Can offer enhanced privacy and confidentiality
Security	Robust due to large number of nodes and validators	Less robust than public chains	May be less secure than public chains
Use Cases	Cryptocurrencies, decentralized applications	Supply chain management, finance, healthcare	Enterprise resource planning, internal record keeping

3. System Design

Building a blockchain involves several steps, which can vary depending on the specific requirements and objectives of the project. Here is a detailed description of the process:

Define Objectives: The first step is to establish clear objectives for the blockchain project, such as creating a decentralized network for financial transactions, enabling smart contracts, or building a new cryptocurrency.

Choose the Right Blockchain Platform: Select a suitable blockchain platform that can support the desired functionality and features of the network. Ethereum, Hyperledger Fabric, Corda, Quorum, and EOS are some of the popular platforms that can be used.

Determine Design Parameters: Decide on the design parameters of the blockchain, such as the consensus mechanism, block size, transaction throughput, and token economics. These design choices will impact the performance, security, and usability of the network.

Develop the Network Architecture: Develop the architecture of the blockchain network, including the node structure, communication protocols, data storage mechanisms, and APIs. This involves crafting the technical specifications and blueprints for the blockchain network.

Build the Codebase: Write the codebase for the blockchain network, using programming languages like Solidity, Golang, Java, or Python. This involves developing the smart contracts, infrastructure components, and user interfaces for the network.

Test and Evaluate: Conduct testing and evaluation of the blockchain network to ensure it meets the established objectives and design parameters. This may involve running simulations, stress testing, and security audits to identify vulnerabilities and areas for improvement.

Implement and Deploy: Once the testing phase is complete, deploy the blockchain network on the desired platform and integrate it with other applications and systems as needed. This involves setting up nodes, launching smart contracts, and configuring the network parameters.

Table 2 The process of building a blockchain

Step	Description
1. Define Objectives	Establish clear objectives for the blockchain project.
2. Choose the Right Blockchain Platform	Select a suitable blockchain platform that can support the desired functionality and features of the network.
3. Determine Design Parameters	Decide on the design parameters of the blockchain, such as consensus mechanism, block size, transaction throughput, and token economics.
4. Develop the Network Architecture	Design the architecture of the blockchain network, including node structure, communication protocols, data storage mechanisms, and APIs.
5. Build the Codebase	Write the codebase for the blockchain network, using programming languages like Solidity, Golang, Java, or Python.
6. Test and Evaluate	Conduct testing and evaluation of the blockchain network to ensure it meets the established objectives and design parameters.
7. Implement and Deploy	Deploy the blockchain network on the desired platform and integrate it with other applications and systems as needed.
8. Monitor and Maintain	Monitor the functioning of the blockchain network to ensure its stability, security, and performance over time.

Monitor and Maintain: Monitor the functioning of the blockchain network to ensure its stability, security, and performance over time. Regular maintenance, upgrades, and bug fixes may be necessary to keep the network running smoothly.

As shown in Table 2, throughout the process of building a blockchain, it's important to consider factors such as security, scalability, interoperability, and user experience. Effective design and implementation of a blockchain network can offer numerous benefits, including decentralization, transparency, immutability, and automation of processes.

The deployment and invocation process of a smart contract involves several steps, which can vary depending on the blockchain platform being used. Here's a general description of the process:

Deployment Process:

Write the Smart Contract: First, write the smart contract code using a programming language supported by the blockchain platform (e.g. Solidity for Ethereum). The smart contract should define the rules and logic that govern its execution.

Compile the Smart Contract: Compile the smart contract code into bytecode that can be executed on the blockchain platform. This generates an output file that contains the compiled code.

Deploy the Smart Contract: Deploy the smart contract to the blockchain platform, either through a command-line interface or a web-based interface. This involves sending a transaction that includes the bytecode of the smart contract to the blockchain network.

Confirm Deployment: Wait for the transaction to be confirmed by the network, which typically involves multiple nodes validating the transaction and including it in a new block.

Get Smart Contract Address: Once the transaction is confirmed, the smart contract will be assigned a unique address on the blockchain. This address can be used to interact with the smart contract.

Invocation Process:

Connect to the Blockchain Network: To invoke a smart contract, connect to the blockchain network using a suitable client application, such as Metamask for Ethereum.

Call the Smart Contract Function: Identify the function you want to call within the smart contract and specify any required inputs. Send a transaction to the smart contract with these inputs.

Confirm Transaction: Wait for the transaction to be confirmed by the network, typically involving multiple nodes validating the transaction and including it in a new block.

View Results: Once the transaction is confirmed, the results of the smart contract function call will be available. These results can include data generated by the function or changes to the state of the blockchain.

Throughout the process of deploying and invoking a smart contract, it's important to consider factors such as data privacy, security, and scalability. Effective deployment and invocation of smart contracts can facilitate automation of various processes on the blockchain platform and create a more efficient and transparent system.

4. System Result

As show in Table 3, implementing a blockchain-based supply chain using Python programming involves several steps, which we can describe as follows:

Choose a Suitable Blockchain Platform: The first step is to choose a suitable blockchain platform that supports the desired functionality and features of the supply chain. Ethereum is a popular option due to its support for smart contracts and decentralized applications.

Set Up the Development Environment: Set up a development environment with the required tools and software for building blockchain applications using Python. This may include installing packages such as Web3.py, Solidity compiler, Ganache, Truffle, etc.

Define the Smart Contract: Define the smart contract that will govern the supply chain transactions and interactions. Write Solidity code to define the rules and logic of the smart contract, including functions to handle product tracking, inventory management, payments, etc.

Compile and Deploy the Smart Contract: Compile the smart contract code into bytecode and deploy it onto the blockchain using a tool like Truffle or Remix. This will generate a contract address that can be used to interact with the smart contract.

Build the User Interface: Build a user interface using Python frameworks such as Flask or Django, which will allow supply chain participants to interact with the smart contract. This interface may include features such as product tracking, inventory management, and payment processing.

Connect to the Blockchain Network: Establish a connection to the blockchain network using the Web3.py library, which provides a Python interface for interacting with the blockchain platform. This will enable the user interface to communicate with the smart contract and retrieve data from the blockchain.

Test and Deploy: Test the application thoroughly to ensure it meets the established objectives and design parameters. Once testing is complete, deploy the application on the desired blockchain network and integrate it with other applications and systems as needed.

Throughout the process of implementing a blockchain-based supply chain using Python programming, it's important to consider factors such as security, scalability, interoperability, and user experience. Effective implementation of a blockchain-based supply chain can offer numerous benefits, including increased transparency, improved traceability, enhanced security, and reduced costs.

Table 3 The performance of a blockchain-based supply chain

Attribute	Description
Transparency	Enhanced transparency across the supply chain through real-time tracking and sharing of data.
Traceability	Improved traceability of goods and materials, with the ability to track them from origin to final destination.
Efficiency	Increased operational efficiency through automation of processes and reduction of manual errors.
Security	Enhanced security through use of cryptographic techniques and decentralized consensus mechanisms.
Cost Reduction	Potential cost savings due to reduced need for intermediaries, streamlined processes, and improved inventory management.
Trust	Increased trust among stakeholders in the supply chain, as the blockchain network provides immutable records and tamper-resistant transactions.
Flexibility	Greater flexibility in terms of governance and collaboration among participants in the supply chain.
Scalability	Limited scalability due to the constraints of current blockchain technology, though advancements are being made to address this issue.

5. Conclusion

Blockchain technology has the potential to revolutionize the maritime transportation industry by enabling secure, transparent, and efficient processes. Some of the key application prospects of blockchain-based maritime transportation systems are:

Supply Chain Management: Blockchain technology can be used to create a tamper-proof record of every step in the supply chain, from the origin of a shipment to its final destination. This can help to reduce fraud, increase transparency, and improve the efficiency of the supply chain.

Cargo Tracking: By using blockchain-based systems, shippers can track their cargo in real-time, reducing the risk of loss or theft. This can also help shipping companies optimize their operations and provide better customer service.

Vessel Maintenance: Ship owners and operators can use blockchain-based systems to manage vessel maintenance, fuel consumption, and other aspects of vessel operations. This can improve safety, reduce downtime, and increase profitability.

Trade Finance: Blockchain technology can be used to streamline trade finance processes, such as letter of credit issuance and documentation management. This can reduce the time and cost associated with financing international trade transactions.

Decentralized Marketplaces: Blockchain technology can enable the creation of decentralized marketplaces, where buyers and sellers can transact directly without intermediaries. This can reduce transaction costs, increase transparency, and improve access to markets.

Overall, the application prospects of blockchain-based maritime transportation systems are vast and varied, ranging from supply chain management to vessel operations, trade finance, and beyond. By leveraging the power of blockchain technology, the maritime transportation industry can become more efficient, transparent, and secure, leading to increased profitability and growth.

Acknowledgment

We acknowledge the funding support from 2022 Guangdong Provincial Ordinary University Characteristic Innovation Projects (No. 2022KTSCX308).

References

[1] Haughton, Joy, Sikorski, et al. Blockchain technology in the chemical industry: Machine-to-machine electricity market. Applied energy, 2017.

[2] Aafaf, Ouaddah, Anas, et al. FairAccess: a new Blockchain-based access control framework for the Internet of Things. Security and Communication Networks, 2017, 9 (18).

[3] Cachin C, Vukoli M . Blockchain Consensus Protocols in the Wild (Keynote Talk). 2017.

[4] Huckle S, Bhattacharya R, White M , et al. Internet of Things, Blockchain and Shared Economy Applications. Procedia Computer Science, 2016, 98: 461-466.

[5] Underwood S. Blockchain beyond Bitcoin. Communications of the ACM, 2016, 59 (11): 15-17.

[6] Yuan Y, Wang F Y. Blockchain: The State of the Art and Future Trends. Acta Automatica Sinica, 2016.

[7] Catalini C, Gans J S. Some Simple Economics of the Blockchain. SSRN Electronic Journal, 2016.

[8] Yermack D. Corporate Governance and Blockchains. Social Science Electronic Publishing, 2015, 21 (1): 7-31.

[9] Iansiti M, Lakhani K R . The Truth About Blockchain:. Harvard business review, 2017, 95 (1): 118-127.

[10] Marc P . Blockchain Technology: Principles and Applications. Post-Print, 2016.

[11] Gervais A, Karame G O, K Wüst, et al. On the Security and Performance of Proof of Work Blockchains. ACM, 2016.

[12] Tapscott D. Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World. 2016.

[13] Eyal I, Gencer A E, Sirer E G, et al. Bitcoin-NG: A Scalable Blockchain Protocol. USENIX Association, 2015.

[14] Zyskind G, Zekrifa D M S, Alex P, et al. Decentralizing Privacy: Using Blockchain to Protect Personal Data// IEEE Security & Privacy Workshops. IEEE, 2015.

[15] Kosba A, Miller A, Shi E, et al. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts// 2016 IEEE Symposium on Security and Privacy (SP). IEEE, 2016.