Article

Supply-Blockchain Functional Prototype for Optimizing Port Operations Using Hyperledger Fabric

Bidah Alkhaldi 1,* and Alauddin Al-Omary 2

1 Department of Computer Science, College of Information Technology, University of Bahrain, Sakhir P.O. Box 32038, Bahrain
2 Department of Computer Engineering, College of Information Technology, University of Bahrain, Sakhir P.O. Box 32038, Bahrain; aalomary@uob.edu.bh
* Correspondence: 202000028@stu.uob.edu.bh

Abstract: Supply chain bottlenecks in port operations lead to significant delays and inefficiencies. Blockchain technology emerges as a viable solution, offering tamper-resistant ledgers, secure transactions, and automation capabilities. While considerable research on developing blockchain-based solutions currently exist, there is a lack of studies that specifically focus on optimizing port document management to speed up supply chain operations. In this paper, a supply-blockchain functional prototype for optimizing port operations using Hyperledger Fabric is introduced. In terms of core functionality, the prototype allows initiation of smart contract corresponding to functions such as creating and editing port-related documents, minimizing manual interventions and enhancing efficiency to reduce port congestion. Furthermore, it provides live tracking of completed events and transactions, facilitating transparency and streamlined oversight. The permissioned nature of Hyperledger Fabric ensures security and robust access controls, aligning well with sensitive port operations. Hyperledger Firefly and its connector framework was used as the middleware to facilitate blockchain integration and various functions of the prototype, while chaincode developed using Go language was used to package and deploy smart contracts. The supply-blockchain framework was used as the theoretical framework for prototype development, and agile project management was adopted to ensure timely completion. The results based on functional and performance testing demonstrate the prototype’s potential in alleviating port documentation bottlenecks and quickly delivering benefits to key stakeholders.

Keywords: blockchain; supply chain management; port decongestion; Hyperledger Fabric

1. Introduction

Port congestion persists as a significant obstacle in global supply chains, hindering the smooth movement of goods. Cumbersome document management processes, often prone to human error and exposed to security risks, are at the core of these delays [1]. Yet, shipping and port operations form the most crucial part of most supply chains, enabling the transfer of goods and materials on a global scale. Any product with a global customer base has to pass through a port in order to be shipped to the intended market. However, this critical node in a supply chain is engulfed in constant delays and slowed document processing and shipment tracking. The Center for Strategic and International Studies reports that slow manual administrative processes result in shipping delays, disrupting the entire supply chain [2]. Simple tasks like bill processing take an average of 42 days to reach the invoicing company [3]. Blockchain technology, with its emphasis on immutability, distributed ledgers, and secure data handling, presents a compelling solution to optimize these essential supply chain operations [4,5]. Several shipping companies and ports, such as Port of Koper and Abu Dhabi ports, have initiated pilot programs involving blockchain integration in shipping and port management [6]. However, most ports around the globe are yet to use this technology.
While several studies [7–10] have developed Hyperledger Fabric prototypes for various supply chains, they do not specifically address challenges in port document management, which represents a significant gap in research. This study seeks to close this gap by developing a Hyperledger-Fabric-based prototype designed to optimize document management within highly complex port operations; this would serve as guidelines for supply chain organizations looking for a simple approach to integrate blockchain into their operations, thereby contributing to supply chain optimization. Moreover, the perception of lengthy integration times and lack of support from key stakeholders are practical barriers that stand in the way of blockchain adoption [11]. The research gap identified and perceptions toward blockchain integration motivate the following research question: what combination of blockchain platform, technical stack, and essential prototype features can facilitate rapid blockchain integration in a shipping port scenario and enable key stakeholders to quickly reap its benefits?

This paper begins with a brief overview of the methodology adopted, a background section outlining key concepts, and a review of related work pertaining to the research topic. Then, the design, development, and functioning of the prototype are explained, after which the results of functional and performance tests are explained. This is followed by a discussion of the benefits to stakeholders and the extent to which the research objectives were accomplished. Finally, the paper concludes with a discussion of potential areas for future work.

2. Methodology

This study used the supply-blockchain framework [12] developed on the basis of a modified MOHBSchain model as the theoretical foundation for the implementation of the prototype. Furthermore, the MoSCoW method was used to prioritize requirements for the prototype, ensuring essential features were implemented first [13]. Once the key requirements were identified, reviews of case studies and existing tools, as well as their technical feasibility, were conducted to identify the most suitable blockchain platform and technical stack. The next step was to begin development of the prototype to validate its viability as a solution to the practical barriers identified. The agile model of “plan, design, develop, test, deploy, and review” was duly followed during the development of the prototype to reduce the risk of delays and ensure timely completion.

3. Background

3.1. Hyperledger Fabric, Smart Contracts, and Consensus Mechanism

Hyperledger Fabric is an opensource blockchain network that prioritizes privacy and security, thereby making it well suited for addressing the challenges faced by complex supply chains such as port operations [14]. The two key components of Hyperledger Fabric are the ledger and smart contracts. A ledger stores critical data about various digital assets, their current value, and entire history of transactions. A smart contract is a digital agreement that specifies the rules necessary for a digital transaction to be considered valid. On the other hand, chaincode refers to code that packages smart contracts for deployment [15]. Digital signing of transactions is facilitated by the chaincode endorsement policy on Hyperledger Fabric. Consensus on Hyperledger Fabric is achieved by using an ordering service composed of dedicated ordering nodes. That is, transactions are sent to endorsing peers, which execute them against their individual state and generate results along with a digital signature, ensuring validity and agreement within the network. Once a transaction is considered valid by the endorsing peers, it is added to the ledger. Unlike other blockchains, chaincode’s execution on Hyperledger Fabric happens at the peer nodes, thereby separating it from the endorsing ordering nodes. This deterministic approach ensures that transactions made on Hyperledger Fabric are accurate and final [16].

The transformative potential of smart contracts in supply chain automation becomes apparent in a Hyperledger Fabric setting. These self-executing contracts minimize manual interventions and decision points, thereby enhancing the transaction integrity and overall
efficiency [14]. Smart contracts offer extensive benefits, streamlining processes ranging from regulatory approval of shipped goods to compliance verification and initiating payments based on predefined conditions. In supply chains, smart contracts can automate the creation, approval, and tracking of documents, minimizing the delays associated with manual paperwork and enhancing the overall efficiency [17]. This automation mitigates delays and cultivates a more dependable supply chain ecosystem for all stakeholders.

Furthermore, Hyperledger Fabric allows for a modular pluggable approach to a consensus mechanism, enabling the selection of a consensus mechanisms tailored to specific needs [18]. This flexibility yields several advantages. Firstly, it ensures tailored security and performance, as different mechanisms balance security, fault tolerance, and transaction speed differently. Secondly, the architecture’s adaptability allows organizations to evolve their consensus mechanisms alongside changing needs. The ability to switch consensus mechanisms aligns with evolving regulatory demands pertinent to port operations in the shipping industry. Unlike some public blockchains that use a Proof of Work (PoW) mechanism to reach a consensus and levy a mining fee, Hyperledger Fabric does not charge transaction (gas) fees to validate transactions [19]. Alternatively, peers within the network validate transactions according to a pluggable consensus algorithm, thereby making it free of transaction fees. As Hyperledger Fabric allows for a modular consensus mechanism, a suitable consensus mechanism can be chosen on the basis of project requirements [20]. Moreover, the reduced cost is integral to the sustainability of supply chain systems, as the number of daily transactions can reach hundreds of thousands depending on the scale of operations. In addition to the cost advantage, Hyperledger Fabric also facilitates reliability and transparency, which are crucial for sensitive supply chain operations.

3.2. Privacy, Transparency, and Security

The permissioned nature of Hyperledger Fabric provides essential privacy and access controls crucial for sensitive port data and regulatory compliance [17]. Unlike public blockchains, where every participant has full visibility into transactions, permissioned blockchains like Hyperledger Fabric grant access on the basis of authorized roles. While Hyperledger Fabric’s architecture prioritizes security, it also facilitates transparency to authorized users to track the status of documents in real time, fostering accountability and improving oversight [21]. Robust security protocols, in line with industry-leading cybersecurity principles, safeguard sensitive port data and mitigate risks inherent in digitalization. This controlled visibility is critical for secure port management where sensitive trade data must be protected while still allowing for streamlined transactions and oversight.

Transparency and security form the foundation upon which trust in digital transactions are built, particularly in the world of supply chains. Transparency, in this context, extends beyond visibility and encompasses creating an immutable audit trail of actions and decisions throughout a product’s journey [21]. A transparent record in the form of a blockchain transaction provides stakeholders with a verifiable history, reducing the likelihood of disputes arising from conflicting information [22]. Furthermore, integrating real-time data fosters agility, as early warnings about bottlenecks or unexpected delays allow stakeholders to take remedial action, minimizing disruptions throughout the supply chain. Security remains paramount in digital transactions, as it ensures the protection of personal data, trade secrets, and the integrity of records [23]. Striking a balance between security and usability is vital, which blockchain strives to deliver.

4. Related Work

4.1. Blockchain for Efficiency and Transparency in Port Operations

Research into blockchain’s potential for port operations consistently highlights the technology’s promise for enhanced efficiency and transparency. Boisdon and Antwi-Boampong [24] present a detailed exploration of how a blockchain-based system with a distributed ledger could transform port supply chain management. Conversely, Irannezhad [25] conducted a broader analysis, examining blockchain’s potential to address challenges within the logis-
tics and freight transportation sectors. Furthermore, blockchain’s immutability is crucial in optimizing record-keeping. The tamper-proof quality of the distributed ledger mitigates the chances of errors and discrepancies, establishing a dependable audit trail. Moreover, they underscore the significance of consensus mechanisms in validating transactions and expediting approval procedures. Decentralizing authority diminishes the necessity for intermediaries, potentially eradicating bottlenecks in port operations.

Irannezhad’s [25] research comprehensively evaluates blockchain’s influence on logistics. The research underscores the challenge of information asymmetries, in which stakeholders lack a unified, precise perspective of data. Blockchain’s decentralized ledger holds promise in mitigating these disparities, enhancing coordination and visibility across the supply chain. Furthermore, it highlights blockchain’s transparency and potential to refine resource allocation and diminish the delays typical in maritime logistics. Irannezhad’s [25] research, with its broader focus, offers valuable perspectives on the hurdles and prospective advantages of blockchain integration across the entirety of the logistics sector. Conversely, Boison, and Antwi-Boampong [24] emphasize a pragmatic, implementation-oriented approach, thereby outlining requirements for a custom blockchain solution for port management.

4.2. Smart Contracts for Supply Chain Automation

The potential of smart contracts to revolutionize supply chain operations through automation is conveyed by various studies. Alqarni et al. [26] conducted an explorative study illustrating how blockchain-based smart contracts can undertake routine tasks such as document approvals, compliance checks, and payment initiation. Their study underscores the potential decrease in manual intervention, streamlining processes and enabling workers to focus on higher-level responsibilities. This shift translates into a notable enhancement in overall supply chain efficiency. By securely embedding contract terms directly into the blockchain, Alqarni et al. [26] demonstrate how smart contracts can autonomously execute agreements. The decentralized nature of blockchain ensures prompt compliance checks and instant document approval once predefined conditions are met. Furthermore, the immediate triggering of payments upon fulfilling obligations promises the minimization of delays often encountered in traditional supply chain financial procedures.

A study by Turjo et al. [27] demonstrates how the inherent tamper-resistant nature of blockchain and its integrated consensus mechanisms form an optimal framework for automating diverse supply chain processes. Furthermore, Turjo et al. [27] extend their study by illustrating the utilization of a peer-to-peer encrypted system in conjunction with smart contracts, thereby fortifying security and ensuring transactional integrity. Their research demonstrates how smart contracts can facilitate processes such as product returns and customer reimbursements in case of issues, thereby nurturing trust among participants across the entire supply chain.

Despite the allure of automation, Khan et al. [28] underscore the critical limitations and raise concerns that demand a balanced approach to smart contract integration, particularly for complex agreements that require nuanced human judgment. While the study acknowledges blockchain and smart contracts’ transformative potential across sectors, their widespread adoption faces hurdles. Legal uncertainties, inconsistent standards, privacy concerns, and a lack of tolerance for errors present challenges. Moreover, the intricacies of legal contracts often demand a degree of interpretation and flexibility that smart contracts may struggle to provide [28]. This highlights the need for continuous oversight and adaptation to ensure smart contracts can effectively respond to regulatory shifts and unexpected complexities in intricate supply chains.

4.3. Capabilities and Challenges of Blockchain-Based Port Management

Ravi et al. [17] emphasize the crucial roles of privacy and security in blockchain-driven port management systems. Their research scrutinizes how permissioned blockchains tackle these key issues by illustrating how Hyperledger Fabric bolsters privacy via access controls. In contrast to public blockchains that expose all transactions, permissioned systems limit
access solely to authorized users, safeguarding sensitive port data. Using encryption in Hyperledger Fabric helps to protect sensitive port data by scrambling data during storage and transmission, making it illegible to unauthorized entities [17].

In contrast, Al-Farsi et al. [29] directed their attention to the persistent risks and essential mitigation tactics in blockchain-powered systems. The authors pinpoint hacking and phishing as enduring cybersecurity menaces capable of jeopardizing the integrity of blockchain-driven port operations. Al-Farsi et al. [29] assert that security protocols, regular audits, and consistent vulnerability patching are indispensable for safeguarding the security of systems utilizing blockchain technology. Their research emphasizes that the dynamic landscape of cyber threats necessitates ongoing vigilance from port authorities and stakeholders embracing blockchain.

4.4. Existing Hyperledger Fabric Prototypes for Supply Chains

Research leveraging Hyperledger Fabric for supply chain management has gained significant traction, offering enhanced security and transparency compared to traditional centralized systems. Hijazi et al. [7] developed a prototype combing a Building Information Modeling (BIM) system with blockchain using Hyperledger Fabric to improve trust and transparency in the context of construction supply chains. As unreliable data exchange is a major problem in construction supply chains, the prototype used JavaScript to develop smart contracts that would facilitate a unified reliable source of supply chain data delivery. A virtual scenario involving cladding attributes during the project handover stage was used to test the prototype, demonstrating the feasibility of the approach. The results underscore the potential of blockchain technology to enhance data reliability, traceability, and accountability in construction supply chains.

Chen et al. [8] focused on integrating blockchain technology specifically within the tea supply chain. Their Hyperledger Fabric prototype aimed to improve traceability and address the challenges of authenticity and trust. By incorporating an Elliptic Curve Digital Signature Algorithm (ECDSA) and Interplanetary File System (IPFS), the solution ensured secure identity confirmation and efficient data storage. While effective in improving traceability within the tea industry, its direct applicability to other supply chain segments or industries remains a potential limitation. Further research could explore its adaptability and scalability in different supply chain contexts.

Marchese and Tomarchio [9] introduced a Hyperledger-Fabric-based prototype aiming to bolster traceability within agrifood supply chains. Their study underscores the necessity of an integrated information system to ensure the quality and safety of food products. The prototype’s decentralized approach to managing traceability information emphasizes blockchain’s superiority over conventional systems, ensuring data authenticity sans a central authority. However, its narrow focus on the agrifood sector might curtail its immediate applicability across other supply chain segments.

The prototype developed in this research offers distinct advancements over existing works that have developed blockchain-based supply chain solutions. Table 1 presents a feature comparison between the prototype developed in this study with other developed blockchain-based supply chain solutions. In addition to facilitating quick and robust integration, the Hyperledger Firefly stack also provides access to intuitive graphical user interfaces through Swagger UI and Explorer UI to deliver a better end-user experience. While other works have focused on specific supply chains pertaining to various industries, none of them have specifically focused on optimizing shipping port operations, thereby making this research’s findings novel and valuable.
Table 1. Feature comparison table.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Construction supply chain</td>
<td>General supply chain</td>
<td>Tea supply chain</td>
<td>Agriculture and food supply chain</td>
</tr>
<tr>
<td>Technical Stack</td>
<td>Hyperledger Firefly, Go, JSON-based data structures</td>
<td>JavaScript, REST API</td>
<td>Python, Solidity</td>
<td>ECDS Algorithm, Interplanetary file system</td>
<td>Kubernetes, Docker, MongoDB</td>
</tr>
<tr>
<td>Features</td>
<td>Document creation, editing, retrieval, real-time blockchain transaction tracking</td>
<td>BIM integration, smart contracts for secure data delivery in construction supply chain</td>
<td>P2P with smart contracts, secure transactions and refunds</td>
<td>Product traceability, anticounterfeit</td>
<td>Agrifood traceability, quality control automation</td>
</tr>
<tr>
<td>Performance Testing</td>
<td>Hyperledger Caliper, Prometheus, Grafana</td>
<td>Virtual business scenario with external validation</td>
<td>Not specified</td>
<td>Hyperledger Caliper</td>
<td>Not specified</td>
</tr>
</tbody>
</table>

4.5. Blockchain-as-a-Service (BaaS)

Blockchain-as-a-Service (BaaS) can help simplify the process of integrating blockchain technology into supply chain domains like port management. BaaS providers use a cloud-based approach to facilitate swift development, hosting, and supervision of blockchain infrastructure. This approach reduces the entry threshold for entities seeking to harness blockchain advantages [30]. Entities within the shipping port ecosystem, such as regulatory bodies, shipping lines, and port authorities, can utilize BaaS to concentrate on crafting bespoke smart contracts tailored to optimize desired processes. Research underscores the merits of BaaS for supply chains, citing cost-effectiveness, expedited deployment, and adaptability [30–32]. However, prudent evaluation is necessary to consider deploying BaaS for port management, particularly concerning vendor lock-in and data sovereignty concerns.

5. Prototype Design

5.1. Use Case Scenario

The case scenario that the prototype strives to addresses is port congestion. While blockchain integration in supply chains allows for better communication between suppliers and producers and reduces human error, it also facilitates automation using smart contracts to reduce lengthy wait times. The nature of transactions in ports requires strong alignment with transparency and security. Goods shipped to and from ports must be well-documented as they can involve tariffs and quality requirements imposed by the importing country. Since these restrictions encompass economic- and health-related concerns, the documents must be free from tampering. While processing digital transactions for ports, blockchain technology allows for the immutability of records through its security features and transparency through its public ledger. This helps to addresses port requirements, such as approval of entry of goods, certification of the contents of parcels, and proof of payment of taxes, among others.

This study developed a functional prototype by applying the supply-blockchain framework, as shown in Figure 1, to the use case scenario discussed above. This framework [12] is based on a modified MOHBSchain model, and only core components of the framework were applied to the development process. Furthermore, the MoSCoW method, as shown in Figure 2, was applied to the case scenario to prioritize the prototype’s requirements. This prototype was implemented using Hyperledger Fabric and a stack called Hyperledger
Firefly, which provides access to tools such as the Hyperledger Firefly API and simplifies the process of connecting to blockchain networks [33]. This allows for digital signing of transactions, controlled access to authorized personnel, and intervention of end users to fix anomalies, thereby helping to speed up operations and reducing port congestion.

Figure 1. Supply-blockchain framework.

<table>
<thead>
<tr>
<th>External devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFID</td>
</tr>
<tr>
<td>Cloud storage</td>
</tr>
<tr>
<td>Devices and other networks</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Must Have</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyperledger Fabric and Firefly functional integration</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Should Have</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functions to create, edit, and retrieve smart contracts</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Could Have</th>
</tr>
</thead>
<tbody>
<tr>
<td>User interface to keep track of changes to smart contracts</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Will Not Have</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced features such as AI forecasting and IoT integration</td>
</tr>
</tbody>
</table>

Figure 2. MoSCoW prioritization of prototype functionality.
In a shipping port scenario, transparency not only encompasses the availability of transaction details but also the accessibility of said information through front-end applications for increased visibility. This allows for real-time viewing of any uploaded documents and their approval status. In theory, transparency builds trust and accountability among authorities and service providers, as any error is audited and can be tracked. Moreover, security goes hand in hand with transparency as transactions made by an organization would only be viewable to those given permission, despite having a public ledger for all transactions. The segmentation also allows for any personal data such as customer info to be protected.

In terms of the functionality considered in using the MoSCoW method, the integration of Hyperledger Fabric with Firefly is a fundamental requirement, while the ability to invoke functions for creating and editing smart contracts is also essential to facilitate its basic functionality, as shown in Figure 2. Invoking a smart contract does not require any programming knowledge from end users, as the invoke function automatically triggers the chaincode corresponding to the creation or editing of a smart contract, as illustrated in Figure 3. Having a user interface that keeps track of changes made to the smart contract is a useful feature that was initially considered and eventually realized using the Explore UI.

![Architecture detailing prototype functions and users.](image-url)

**Figure 3.** Architecture detailing prototype functions and users.

### 5.2. Architecture

The architecture depicts the connection between technical stack, central blockchain, and smart contracts. Figure 4 presents the Hyperledger Firefly architecture. The architecture is composed of a Hyperledger Firefly stack, which is essentially a supernode that comprises APIs and other tools. The FabConnect API allows for integration with public or enterprise blockchains, enabling the prototype to work with existing systems and ensuring transparent transaction management. It also acts as a hub connecting the Hyperledger Fabric blockchain with other system components. While chaincode was developed using the Go language, all transactions are processed within the Hyperledger Fabric framework using channel-based privacy for permissioned access. The encrypted data are transferred to Hyperledger Firefly for further handling by other APIs, in addition to providing access to interfaces...
such as Swagger UI for end users to interact with the prototype and Explorer UI for transaction tracking.

![Hyperledger Firefly architecture](image)

Figure 4. Hyperledger Firefly architecture.

6. Prototype Development and Deployment

A linear process was followed during prototype development to realize the basic functions required to exhibit its potential to optimize document management in congested ports. The five main steps involved in the prototype development process include the following: (1) setting up the environment and technical stack, (2) connecting to the central blockchain, (3) creation of smart contracts, (4) automation of verification processes, and (5) processing of transactions. These steps encompass the development of the prototype architecture that will connect with the blockchain and handle transactions, as well as the user interface needed to obtain inputs from end users to carry out functions. The following diagram represents the process involved in prototype design and development. Figure 5 presents the prototype development and deployment process.

The first step was to install the Windows Subsystem for Linux (WSL) to allow Hyperledger Firefly to be installed on a Windows system, followed by configuring it to work with the Docker Desktop application. Following this, Hyperledger Firefly was installed locally and the stack was initialized, thereby starting the process of connecting with the central blockchain. The specific consensus mechanism used in this prototype is Raft orderer, which was chosen for its simplicity, resilience, and fault tolerance [16]. A Raft-based consensus algorithm involves electing a leader node responsible for ordering transactions. A transaction is considered valid only when a consensus is reached among the leader node and follower nodes, thereby ensuring consistency [16]. Furthermore, the leader node replicates the transaction log on various follower nodes to facilitate crash fault tolerance.

The chaincode (see Appendix A) for deploying smart contracts was developed using the Go language by customizing basic templates provided by the Hyperledger Foundation to fit the case scenario [34]. Furthermore, a Firefly interface document was created using JSON-based data structures, which enables Hyperledger Firefly to comprehend and interact with the chaincode developed [35]. The next step was to package chaincode via the command line interface to automate the verification process. This is followed by deployment of the chaincode, which allows blockchain transactions to be processed.

Upon successful deployment, a contract interface was created using the Sandbox UI provided by Hyperledger Firefly. Smart contract functions can generate events that Hyperledger Firefly listens to, enabling notifications to users or triggering of HTTP requests for further action. Once the contract interface is registered, it provides access to the REST API and associated user interfaces. Swagger UI allows end users to invoke various functions of the prototype, such as creating, editing, and retrieving port-related documents, are implemented. Furthermore, Explorer UI provides a snapshot of transactions carried out.
In terms of logistics, there are two main modes of transportation that need to be taken into consideration, namely, container ships and delivery vehicles. For decongesting ports, the main focus is on container ships, as these have a greater impact on the volume of goods entering and being stored in ports, which is what this prototype strives to address. Data sets used by the prototype are categorized into the following: chassis number, comment, container number, export booking, reference client, reuse date, seal number, and shipping line [36]. These data sets were used in the smart contract to allow automation of transactions. Data sets such as discharge port, loading port, export SCAC, import SCAC, and SCAC code help in determining the responsibility of shipping lines to transfer, load, and/or unload parcels that have arrived or will leave the designated ports. The designation of an SCAC code for service providers helps monitor the efficiency rate of each service provider to identify current gaps and import and export bottlenecks.

Data sets can also pertain to practical aspects of supply chain operations. For instance, the comment data set may include information related to items such as perishable, fragile, or contain harmful chemicals. To support forecasting of expected volume of goods entering and leaving ports, data sets such as shipping line, reuse date, export booking date, and import delivery date provide reasonable timelines with respect to arrival and departure.
timelines for shipping containers to and from designated ports. These data sets directly impact port decongestion as the flow of parcels within ports can be directly monitored, allowing for coordination to improve efficiency.

The container number and seal number correspond to a package’s identifying code and security code, which is necessary for real-time location tracking and delivery status. The chassis number corresponds to the container ship on which the package is being shipped, which serves as the identifying number for each vessel. Finally, the reference client data set allows users to know the service provider responsible for delivery. These data sets ensure accountability and transparency within the blockchain network, particularly for end users to track and confirm receipt of a package.

7. Results

7.1. Functional Testing

In terms of functionality, the prototype allows end users to create, update, delete, and retrieve documents related to port operations while successfully updating these transactions on to the blockchain. In addition to streamlining processes susceptible to bottlenecks, this system can facilitate the management of documents pertaining to regulatory approvals, shipment manifests, and inspection reports. In terms of design and usability, the Swagger UI provides a user-friendly interface to authorized end users to engage with the prototype, triggering actions via smart contracts and automatically registering transactions to the blockchain. The chaincode deployed enables automation of document workflows, lowering reliance on manual processes and minimizing the risk of human error. Furthermore, the Explorer UI enables transparent monitoring of transactions, augmenting transparency and accountability. The diagram shown in Figure 6 displays the processes involved in invoking a smart contract corresponding to the creation of a new document.

Figure 6. Smart contract—invoke function.

When a smart contract function is invoked by the user through the Swagger user interface, it is forwarded to the blockchain through FabConnect and relevant functions are performed as per user inputs. Figure 7 shows the create document function accessed through Swagger UI.
Following this, the response is evaluated and a consensus is reached by Hyperledger Fabric, while it is recorded and stored by Hyperledger Firefly on the central blockchain ledger. However, in the event of a user entering incorrect data outside of the acceptable parameters (e.g., incorrect container number), the smart contract’s invoke function fails. The Explorer user interface, as shown in Figure 8, allows for monitoring of the blockchain’s transactions, as it displays both successful and failed attempts.

7.2. Performance Testing

Performance testing of the prototype was carried out using a combination of tools including Hyperledger Caliper, Prometheus, and Grafana. Hyperledger Caliper was first set up to run performance tests to measure the latency and transaction throughput, as shown in Figure 9. Following this, the performance data generated by Caliper during the testing was extracted using a monitoring tool called Prometheus, which helps to gather metrics using an HTTP pull model. These extracted data were then sent via a push gateway to an analytics application called Grafana, which helps to generate graphs of the real-time data. This approach to performance testing ensures that authentic performance data are gathered and represented visually.
The report indicates a high success rate for the transaction, as zero failed transactions were recorded. Furthermore, a throughput of 223.4 transactions per second with an average latency of 0.01 s indicates fairly good performance. The performance testing was conducted on a Core i5 system with 7.4 GB of usable memory. The testing process involved the creation of assets, which refers to writing new objects to the contract, which corresponds to peaks in memory and CPU consumption. The results are shown in Figure 10 and indicate that prototype consumed 1–2.5% of CPU resources. Furthermore, it also shows that the prototype utilized between 115 and 135 MB of memory during the tests as presented in Figure 11.

During the performance testing, a scenario involving two users simultaneously using the prototype was simulated to evaluate the total number of transactions submitted versus the number of successful transactions. This test indicated a peak throughput of 130 trans-
actions per second, with a high success rate of almost all transactions being successfully finished as shown in Figure 12.

![Image of Figure 12: Submitted vs. finished transactions performance graph.](image)

**Figure 12.** Submitted vs. finished transactions performance graph.

### 8. Discussion

The decision to develop the prototype based on the Hyperledger Fabric network and the Hyperledger Firefly technical stack was instrumental in realizing blockchain integration within a short span of time, without compromising the key benefits of security, efficiency, immutability, and transparency. The results of the performance testing indicate relatively good performance measures in terms of throughput and efficient resource utilization. It is hoped that this efficient approach will encourage supply chain organizations to integrate blockchain into their operations, dispelling fears of long integration times and delayed realization of benefits, thereby gaining the support of key stakeholders [11]. Furthermore, this approach serves as a secure alternative to BaaS providers for supply chain organizations, as it keeps operations in house and addresses data sovereignty concerns [32]. While the core functions implemented, such as creating, editing, deleting, and retrieving documents, may seem simple on the surface, they can be applied to a number of port operations, such as shipment manifests, inspection reports, and regulatory approvals, to optimize and automate a wide range of processes.

The features included in this prototype target port decongestion by considering all critical data sets involved, thereby improving efficiency in processes related to document processing. Furthermore, the prototype has the potential to provide both direct and indirect benefits to various stakeholders involved in supply chains, thus allowing for a varying degree of benefits to be enjoyed starting from producers up to consumers [25]. More specifically, one of the key stakeholders that can potentially benefit from these efforts is regulators, as it allows for the automation of repetitive tasks. Bottlenecks in ports often result from either a high volume of documents or special issues encountered in time-consuming cases. Since approval can be automated on the basis of a predetermined checklist, it can free up regulators and allow them to focus on more complex cases that require manual intervention. Shipping companies also directly benefit from blockchain implementation in ports, as it speeds up timely delivery and improves transparency. Apart from this, another benefit to shipping companies is the reduction in costs resulting from schedule disruptions and increased transit times.

### 9. Conclusions and Future Work

This study demonstrates Hyperledger Fabric’s potential to facilitate blockchain integration to alleviate inefficiencies in port document management. Hyperledger Fabric’s permissioned nature along with Firefly’s seamless integration facilitated creation of a solution meeting the security, transparency, and access control needs of various stakeholders.
While chaincode developed using Go language was used to deploy smart contracts, Raft orderer consensus mechanism was deployed to facilitate resilience and fault tolerance necessary to carry out uninterrupted port operations. Hyperledger Fabric’s modular architecture offers significant flexibility for future adaptations and feature expansions.

Based on functional and performance testing, the prototype shows promising results in streamlining port document processes. However, in the context of real-world, high-volume port operations, it is vital to recognize that blockchains have inherent limitations in terms of scalability. Future research can explore potential optimizations, like off-chain data storage or alternative consensus mechanisms, to strike a balance between efficiency, security and scalability. Future improvements to the prototype can potentially facilitate automation of approval processes based on predefined rules and compliance checklists, while Internet of Things (IoT) integration can facilitate real-time location tracking. While this research serves as a foundation showcasing feasibility of a Hyperledger-Fabric-based solution to address challenges in congested ports, future efforts can explore incorporating cutting-edge blockchain technologies, such as Nonfungible Token (NFT) schemas, which can help improve traceability and ownership verification.

**Author Contributions:** Conceptualization of research, B.A. and A.A.-O.; methodology, B.A. and A.A.-O.; formal analysis, B.A. and A.A.-O.; investigation, B.A. and A.A.-O.; resources, B.A. and A.A.-O.; data curation, B.A. and A.A.-O.; validation, B.A. and A.A.-O.; writing—original draft preparation, B.A.; writing—review and editing, B.A. and A.A.-O.; visualization, B.A.; project administration, A.A.-O. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable for this article.

**Acknowledgments:** We would like to thank our families, friends, and the University of Bahrain for their long-standing continued support.

**Conflicts of Interest:** The authors declare no conflicts of interest.

**Appendix A**

![Figure A1. Chaincode.](image-url)
References


8. Chen, C.L.; Zhan, W.B.; Huang, D.C.; Liu, L.C.; Deng, Y.Y.; Kuo, C.G. Hyperledger Fabric-Based Tea Supply Chain Production Data Traceable Scheme. Sustainability 2023, 15, 13738. [CrossRef]


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.