Tamper-proof Blockchain-based Contracts for the Carriage of Goods by Road

Francisco Carreira
University of Coimbra, CISUC, DEI
Coimbra, Portugal *fmbc@dei.uc.pt*

Paulo Rupino da Cunha
University of Coimbra, CISUC, DEI
Coimbra, Portugal *rupino@dei.uc.pt*

João Barata
University of Coimbra, CISUC, DEI
Coimbra, Portugal *barata@dei.uc.pt*

Jacinto Estima
University of Coimbra, CISUC, DEI
Coimbra, Portugal *estima@dei.uc.pt*

Abstract

We propose an architecture for a decentralized electronic consignment note (eCMR) management system leveraging blockchain technology. We used Design Science Research (DSR) to guide our work in cooperation with a leading logistics and transportation management solution provider. We started with a multi-vocal analysis of the literature on eCMR, regulations, and technical documentation. Then, we held several rounds of discussions with the road carriage business stakeholders to establish the core software requirements and architecture. Finally, we have implemented and tested a proof-of-concept. The resulting artifact enables logistics stakeholders to track consignment notes of interest in real-time, relying on a tamper-proof database with complete eCMR lifecycle traceability. The proposed architecture and proof-of-concept can guide the design of future decentralized eCMR services, helping to implement reliable and transparent logistics processes.

Keywords: Digital transformation, Logistics, Blockchain, DLT, eCMR

1. Introduction

Freight transport by road represented 54.3% of the total goods transport activities of the European Union (EU) and 1.2% of the total EU net turnover in 2022 [17, 18]. Nevertheless, a significant amount of information exchanged among stakeholders in this sector still relies on paper documents, like the CMR (Convention on the Contract for the International Carriage of Goods by Road) consignment note [37]. This document serves as evidence for the contract of carriage and the handling of goods by the carrier. It is applicable when the pickup and delivery locations are in two different countries, with at least one being a contracting party under the CMR convention. This consignment note contains information about the shipped goods and identification of shipping, transporting, and receiving parties. It is a control document that must travel with the goods, and its absence may lead to an administrative or criminal sanction. In 2008, the Additional Protocol to the CMR introduced the electronic consignment note (eCMR), a digital document version [36], but its use is still not widespread.

The digital transformation of supply chains is evolving globally, leveraging multiple technologies to increase efficiency, safety, security, resilience, and traceability [1], [4]. In this process, blockchain — a tamper-proof decentralized ledger — has emerged as a reliable backbone for storing and processing information [5], [35]. The research project reported in this paper is part of a larger 90-million-euro transformative venture aimed at revolutionizing Portugal's port, multimodal, and transport sector, emphasizing zero-emission operations and energy transition technologies, with a particular spotlight on enhancing efficiency and sustainability in port operations. Our research objective is as follows:

- **RO:** Develop a solution using blockchain technology to provide tamper-proof and distributed eCMR management.

The rest of this paper is structured as follows: Section 2 introduces essential concepts on eCMR, blockchain, their potential integration, and commercial eCMR platforms. Then, Section 3 presents our Design Science Research (DSR) approach [7, 8]. Section 4 outlines the proposed system, including requirements and architecture, followed by the implementation details in Section 5. Subsequently, Section 6 evaluates the solution, and Section 7 summarizes the conclusions, limitations, and future work opportunities.

2. Background

2.1. Electronic Consignment Notes

Digitalization of trade documents is not a new phenomenon. This process has been evolving over several decades, driven by the need to improve efficiency, accuracy, and speed in global trade operations. The development of Electronic Data Interchange (EDI) in 1948 [15] and its first implementation in the 1960s marked the beginning of this transformation, enabling businesses to exchange documents electronically, such as purchase orders and invoices, significantly reducing the reliance on paper-based processes [16]. Furthermore, the popularization of the Internet in the 1990s further accelerated digitalization, with the World Wide Web providing a platform for more sophisticated and accessible electronic document exchanges [6].

Digital alternatives have challenged the paper-based consignment notes in logistics and transportation, particularly the CMR. The adoption of the abovementioned additional eCMR protocol illustrates the industry's shift towards digitalization. This transition addresses the inefficiencies of paper-based processes by offering a digital alternative that promises enhanced efficiency and accessibility [33].

2.2. Blockchain Technology

Blockchain is a tamper-proof decentralized digital ledger [40] . It is revolutionizing how data is stored, and transactions are recorded and verified across a network of peers. Unlike traditional databases controlled by central authorities, blockchains distribute data, ensuring transparency, security, and immutability. It has applications across various sectors, including finance, healthcare, and supply chain management [1], [4].

The first live blockchain concept was introduced in 2008 in a whitepaper written by Satoshi Nakamoto [25], alongside the creation of Bitcoin, a peer-to-peer electronic cash system that solved the double-spending problem without needing a trusted third-party. This groundbreaking approach laid the foundation of blockchain technology and the subsequent development of cryptocurrencies and applications. Blockchain technology has several key features [40]:

- **Decentralization** removes central authorities’ control over the database, distributing it across a network of peers.
- **Transparency** ensures that all network participants can access the ledger and verify transactions.
- **Security** is heightened through cryptographic hashing and consensus mechanisms, making it nearly impossible to alter recorded data without network consensus.
- **Immutability** guarantees that once data is entered into the blockchain, it cannot be changed.

Blockchain systems can be categorized into three main types: public, private, and consortium [40]. Public blockchains are open to anyone who wishes to join. They usually rely on consensus algorithms like Proof-of-Work (PoW) or Proof-of-Stake (PoS) for transaction verification and security. Private blockchains are restricted to specific organizations, offering more control over participants and faster transaction processing. Consortium (or hybrid) blockchains are governed by organizations combining public and private systems elements to balance control with transparency and security.

2.3. Blockchain for Managing Cargo Documents

We conducted a systematic literature review to explore the potential of blockchain technology in optimizing cargo documentation within the shipping industry following established protocols [38]. We refined our search keywords through initial exploratory searches, settling on the final search expression outlined as follows:

```
TITLE-ABS-KEY ( blockchain AND cargo AND documents ) AND ( LIMIT-TO ( LANGUAGE ,
"English" ) )
```

Using the Scopus database, in April 2024, we searched by title, abstract, and keywords in English-language documents without imposing date restrictions. This initial search yielded eleven results, categorized into six articles, four conference papers, and one book chapter. All identified works focus on maritime cargo transportation, revealing a gap in road transportation studies.

Since no relevant papers were found in Scopus, we conducted additional searches in Google Scholar, with filters applied to exclude non-peer-reviewed studies, those lacking in practical application, or built on outdated blockchain technologies. This broader search brought forth 3,700 results. We highlight three of the most relevant found in this sample. Finke et al. [19] detailed a Distributed Ledger Technology (DLT) platform designed to enhance customs processes in supply chains, highlighting potential efficiency improvements in customs clearance and overall logistics. Loklindt et al. [22] discussed the potential for blockchain in facilitating document exchanges within the shipping industry, focusing on digitalizing container shipping documents to overcome the drawbacks of traditional paper-based systems. Finally, Jović et al. [21] analyzed the impact of blockchain technology on the shipping industry, pointing out its capabilities in improving transparency, security, and efficiency, notably through projects like Maersk and IBM's TradeLens [2], [20].

Given the specific focus on blockchain and its application to cargo documents in the shipping industry — especially the gap in the literature concerning CMR documents — our search revealed the potential for new contributions in this area.

2.4. eCMR Solutions

Important grey literature exists in this domain. The United Nations Economic Commission for Europe (UNECE) makes available a document listing the pilot projects that support the digitalization of the CMR document [39], such as Collect + Go [11], TransFollow [34], and Pionira [28]. Beyond the scope of UNECE-accredited initiatives, CargoLedger [41] utilizes blockchain technology to focus on the digitalization and traceability of shipping and cargo documentation processes.

We found two additional projects still being developed. Between 2019 and 2020, the DIGINNO-Proto project [23] developed an eCMR indexing prototype for paperless international logistics, in which the source code was made available as free software, under the MIT license, from the Estonian public code repository. Later, in 2021, the Open Logistics Foundation [26], a non-profit foundation aimed at promoting digitalization in logistics and supply chain management based on open source, was established. One of their projects is digitalizing the CMR document, whose source code is available at their Git repository.

From the abovementioned market solutions and platforms, only CargoLedger uses blockchain technology. The remaining solutions do not leverage blockchain but (1) provide public API endpoints for external connection and (2) require interaction with their proprietary systems.

In summary, while blockchain technology shows promise for revolutionizing the shipping industry's documentation processes, the specific application to CMR documentation is markedly underrepresented in existing literature and current market solutions. This gap underscores an opportunity for future investigation and development in leveraging blockchain for more secure, efficient, and transparent cargo documentation practices.

3. Research Approach

Design Science Research [27] is a systematic, iterative research approach that emphasizes creating and evaluating artifacts designed to solve practical problems. This methodological framework is particularly suited to Information Systems (IS) research, where the impact of technological artifacts on organizations and society is key. Our project employs DSR to create a software artifact to address a real-world problem: providing secure and distributed eCMR management. Following the steps of Peffers et al. [27], we: (1) identified the problem in a large-scale project, justifying the value of a solution; (2) defined objectives and requirements; (3) designed and developed artifacts (e.g., architecture, proof-of-concept); (4) demonstrated them to project partners; (5) assessed utility and quality based on real case studies and validation of concepts from experts; and finally, we (6) communicated the results in a scientific publication.

4. System Proposal

This section describes the proposed solution, called eCMR Manager, and includes its requirements and architecture. We held several meetings with key stakeholders in the road carriage of goods sector to elicit requirements. These discussions allowed us to identify distinct roles, needs, access levels, and interaction methods with the system. The identified stakeholders and their requirements are as follows:

- **System Owner:** Must be able to register employees and companies by adding their details to the smart contract. Additionally, the owner must be capable of managing these entities by setting limits on companies/employees, linking employees with specific companies, and removing employees from the system as necessary.
- **Driver:** Must be able to access the eCMR document and must be able to update the delivery status in real-time.
- **Carrier:** Must be able to create, manage and sign eCMR documents.
- **Sender:** Must be able to update and sign eCMR documents.
- **Authorities:** Must have access to a secure portal for auditing purposes, be able to review specific eCMR documents as needed, and access historical data for compliance checks and enforcement actions.
- **Receiver:** Must be able to confirm receipt of the consignment by accessing the eCMR document before delivery to review consignment details and sign it after delivery.

Table 1 shows the functional requirements prioritized according to the MoSCoW method [12].

Table 1. Project Requirements

Requirement ID	Requirement	Priority	Short Description
REQ-001	Register Company	M (Must)	Enables the registration of a company and returns a universal unique identifier (UUID) to be used in other API operations.
REQ-002	Register Employee	M	Enables the registration of an employee by its public address for use in other API operations.
REQ-003	Associate Employee w/ Company	M	Enables the association of a company and an employee, allowing the employee to submit and retrieve files only

Requirement ID	Requirement	Priority	Short Description
			on his associated company.
REQ-004	Store CMR	M	Enables the secure storage of a CMR file within the system.
REQ-005	Update CMR	M	Enables the secure update of a CMR file within the system.
REQ-006	Show CMR	M	Returns the CMR file in PDF format.
REQ-007	Grant Temporary View Access to CMR to an External Actor	M	Returns a temporary, shareable link to the CMR file in PDF format.
REQ-008	Sign CMR	M	Enables an actor to acknowledge the receipt of the cargo.
REQ-009	Search CMR	C (Could)	Enables searching targeting specific fields across the stored CMRs.
REQ-010	Retrieve CMR	M	Returns the CMR file in XML format.
REQ-011	Restrict Company	M	This operation enables the restriction of a company and its employees, restricting what they can do. The restriction level of a user is as high as the restriction level of his company.
REQ-012	Restrict Employee	M	Enables the restriction of an employee, restricting their actions.

The component diagram in Figure 1 is an overview of the system's architectural design. The purple components inside the eCMR Manager container scope (dashed rectangle) are project-developed, and the grey components are external, independent systems.

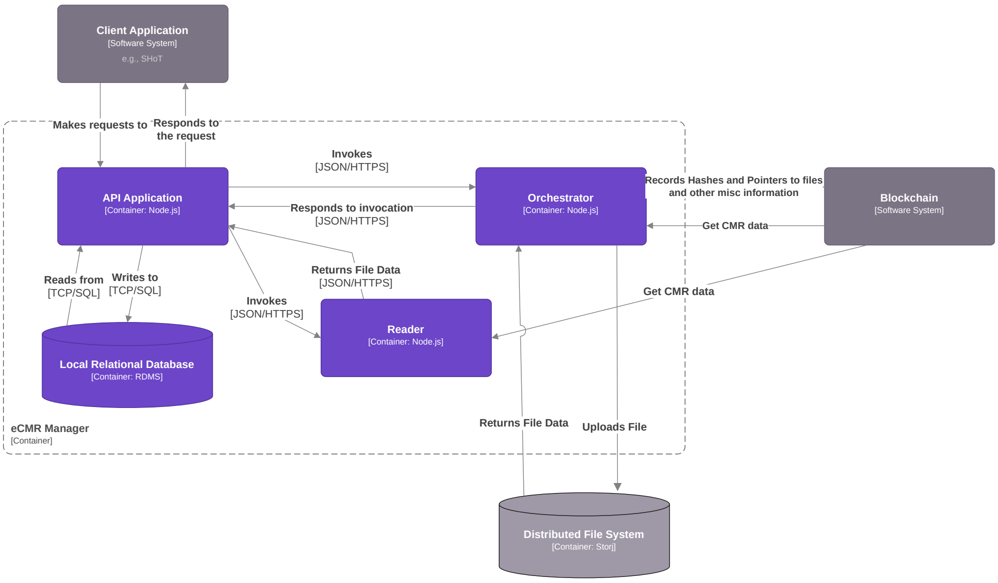


Fig. 1. Component Diagram of our system, based on the C4 model.

We utilize a Distributed File System (DFS) to store the actual CMR documents, while the blockchain records only the content identifier corresponding to these documents. It's important to note that this model of systems saving only the content identifier on the blockchain has been wildly researched and its application isn't restricted to the transport sector, and examples of its use can be found, for example, concerning document management in the education [31], medical [3], and pharmaceutical sectors [10].

This design decision ensures that the bulk of the data is handled by the DFS, which is optimized for large-scale data storage, while the blockchain is used to secure the integrity of the data with its immutable ledger, providing only the necessary references. The process of storing a CMR involves any Client Application (on the top of Figure 1) sending the CMR data, in XML format, to the API Application that stores a temporary request on the Local Relational Database. After signature, it is forwarded to the Orchestrator. The Orchestrator then stores the CMR data in the DFS and records the corresponding hashes and pointers in the blockchain. For retrieval, the request flows from the Client Application through the API and Reader to fetch the CMR data from the DFS and blockchain information, ensuring secure and traceable management of eCMR documents. The project-developed components shown in Figure 1 are as follows:

- **API Application:** Establishes a secure connection between the Client Application and eCMR Manager through Representational State Transfer (REST). It is used for third-party software to interact programmatically with the eCMR Manager, accepting HTTPS requests from client applications.
- **Orchestrator:** Implements the logic of the eCMR Manager, which includes CMR structure verification and document hashing and facilitates communication with the Blockchain and the DFS.
- **Local Relational Database:** Stores auxiliary data. Before transmitting any information to the blockchain, a temporary record of the user's requested operation is stored. This record is then digitally signed by the user and sent to the blockchain. After the transaction is confirmed, it is deleted.
- **Reader:** Reads stored data (e.g., an eCMR) and returns it to the requesting system through the API.

The decision to integrate a public blockchain into the supply chain digital infrastructure is driven by the need for a tamper-proof security mechanism, transparent operations, and an independent and neutral platform that can cater to the diverse stakeholders within the industry. Moreover, blockchains' public nature enables all participants in the value chain to verify transactions independently. They also offer greater resilience against hacking, such as majority and denial-of-service attacks, unlike private blockchains, which rely on fewer nodes among fewer partners. A key component, smart contracts on the blockchain, also operate within a Virtual Machine (VM). If the blockchain supports a widely recognized VM bytecode specification, like the Ethereum Virtual Machine (EVM), it allows for the avoidance of being tied to a specific vendor. As newer entities supporting the exact VM specification come into play and present themselves as more advantageous, they can adopt these new platforms without overhauling the existing smart contracts. This ensures our solution flexibility and continuity without the constraints of platform-specific limitations.

Lastly, regarding external storage, by leveraging decentralized storage solutions such as IPFS (InterPlanetary File System) [14] or similar blockchain-based storage networks, like Storj [32] or Filecoin [30], data can be stored in a manner that is separate from the onchain storage while still benefiting from the enhanced integrity and accessibility that blockchain technology provides. This approach ensures faster transaction times and lower costs. Moreover, decentralized storage maintains the core principles of blockchain — decentralization and security — by distributing data across a network of nodes, thus preventing single points of failure, and enhancing data resilience.

5. Implementation

This section describes specific technologies and implementation decisions for the development of a proof-of-concept.

5.1. Application Programming Interface

Developing an API was necessary to expose the smart contract functions and introduce a direct connection to the blockchain. This intermediary layer between a client application and the blockchain exposes endpoints through REST services via the Hypertext Transfer Protocol Secure (HTTPS) protocol. By doing so, the API ensures secure data transmission, mitigating the risks associated with data interception or eavesdropping.

The API was designed to mirror the capabilities of the developed smart contract, although it is not limited to transactional and query operations. The system not only streamlines access to smart contract functionalities but also embeds additional logic necessary for the efficient processing of these operations, thus enhancing the overall integrity and functionality of the blockchain interaction framework. Every function accepts JavaScript Object Notation (JSON) requests, making the API easily accessible.

5.2. Blockchain Network

For the scope of this project, Polygon was chosen as the blockchain network. Polygon [29] is a blockchain platform that provides scalable, secure, and instant transactions with Ethereum compatibility. It is known for its Layer 2 scaling solutions, which aim to address some of the main challenges faced by blockchain networks today, such as high fees and slow transaction speeds, without compromising on security.

For preliminary testing, we used the Amoy testnet network. A testnet is an identical copy of a blockchain network created solely to rigorously test and experiment with code and blockchain transactions. To communicate with the testnet, we used an Alchemy RPC endpoint via the ethers.js v6 library.

5.3. Smart Contracts

We wrote our smart contract in Solidity, the most adopted language by blockchain developers [9]. Structs in Solidity allow for the creation of custom types that can group several variables. To store data in our system, we specified four structs in our smart contract:

- **Employee**: Encapsulates the basic details of an employee, including a unique employee identifier and the employee’s public wallet address.
- **Company**: Contains a unique company identifier, creating an Identity and Access Management system.
- **Document**: Defines a document within the system. It includes a unique ID identifying the document and a company identifier that links the document to a specific company, establishing an association. To keep track of all the changes on a document, an array of **DocumentVersion** structs is included.
- **DocumentVersion**: Includes information on a specific version of the document, such as the version number, a file checksum to verify document integrity, an encrypted decentralized file storage pointer for accessing the document, a timestamp of the modification, and the signer's public address.

To further augment the system’s usability and accessibility, we implemented meta-transactions, a concept introduced in ERC-2771. Meta-transactions enable users to interact with smart contracts indirectly through a third party, thus bypassing the need to possess the blockchain native asset for transaction fees or to engage with the blockchain network directly. This innovation is particularly beneficial in enhancing user experience and reducing entry barriers for users who do not want to engage with the blockchain directly. Meta-transactions function by allowing a relayer to submit transactions to the blockchain on behalf of the user. Upon receiving this signed transaction, the user signs a transaction with their private key, and the relayer submits it to the network, covering the gas fees. This mechanism democratizes access to blockchain functionalities and simplifies transaction processes for end-users. Figure 2 illustrates the information flow for a meta-transaction on our system.

In the smart contract (on the right of Figure 2), cryptographic signatures created off-chain allow the extraction of public addresses directly from the signature itself, as they are used to prove ownership of an address without exposing its private key. Implementing EIP-712, a standardized method for structuring and signing human-readable and machine-verifiable data is key here, as it facilitates the creation of typed, structured data for signatures. Additionally, our smart contract incorporates a proprietary signature system that bypasses the necessity for external wallets, i.e., MetaMask [42]. However, it remains compatible with external wallets, leveraging the structured data capabilities of EIP-712. It is also possible to sign transactions with an external software by requesting the request's structure and data via a GET request with the provided request’s unique identifier.

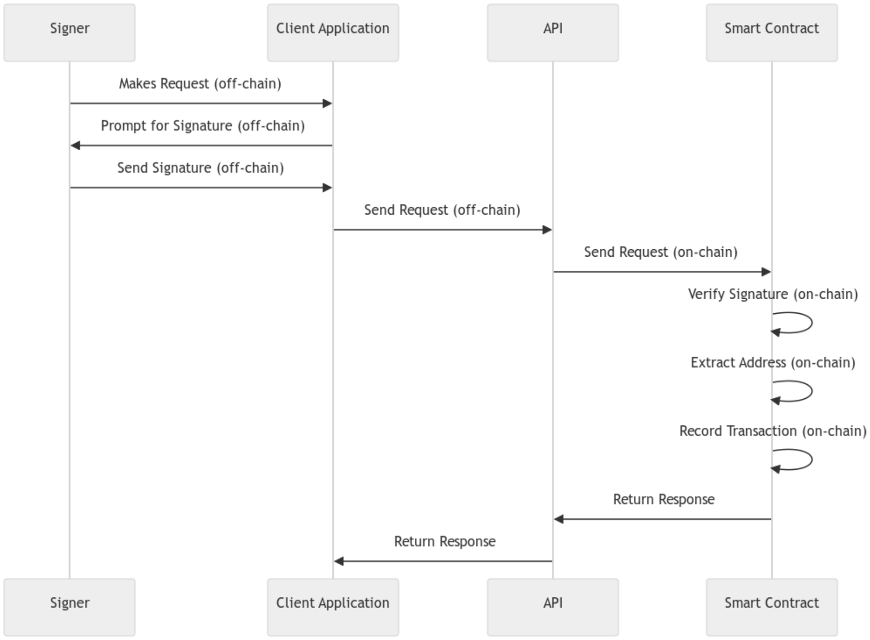

Signer
Client Application
API
Smart Contract
Makes Request (off-chain)
Prompt for Signature (off-chain)
Send Signature (off-chain)
Send Request (off-chain)
Send Request (on-chain)
Verify Signature (on-chain)
Extract Address (on-chain)
Record Transaction (on-chain)
Return Response
Return Response
Signer
Client Application
API
Smart Contract

Fig. 2. ERC-2771 transaction information flow.

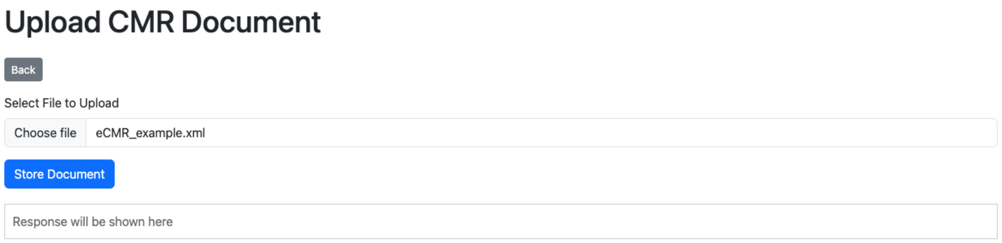
5.4. External Storage

Considering the need for compliance with the General Data Protection Regulation (GDPR) in Europe, particularly the "right to be forgotten", we explored alternative decentralized storage solutions that can accommodate these legal requirements effectively. Traditional blockchain storage, which involves storing data directly on-chain, is not efficient for handling large-scale data due to slow transaction times and high costs. Moreover, platforms that leverage IPFS, like Filecoin, do not guarantee that data can be fully erased, potentially conflicting with GDPR mandates.

We opted for Storj [32], a decentralized storage system that addressed all the abovementioned concerns. Unlike traditional blockchain or other decentralized storage solutions, Storj utilizes a decentralized network of independent nodes to store data offchain. This approach not only accelerates transaction times but also significantly reduces costs by minimizing the load on the blockchain.

5.5. Proof-of-Concept Testing

We developed a front-end web application for testing and discussion with experts. It has a Graphical User Interface (GUI) to simulate visual functionalities for users to interact with our solution, serving the users with API responses. For the sake of simplicity, we present two examples of pages in the developed web application, which are the CMR document upload, shown in Figure 3, and the transaction signing, displayed in Figure 4.



Upload CMR Document

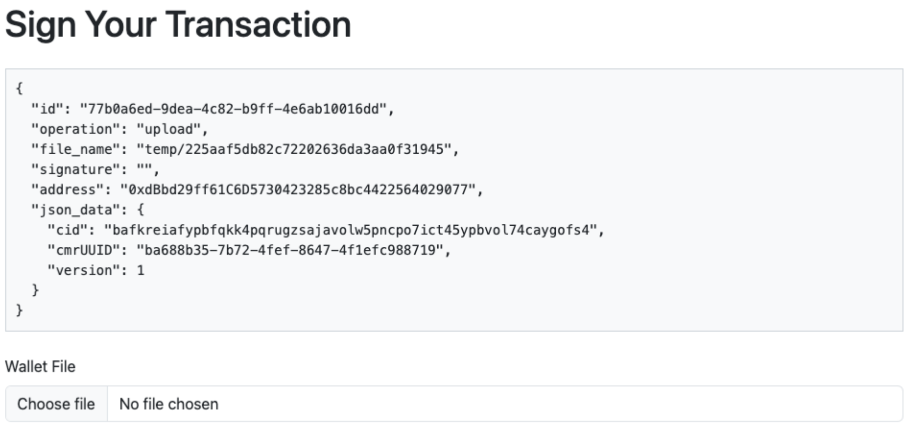
Select File to Upload

eCMR_example.xml

Response will be shown here

Fig. 3. CMR Document Upload Page

The CMR document upload page (Figure 3) serves as the gateway for the user to upload a CMR document. After uploading a document by clicking on the "Store Document" button, a popup is shown to the user to verify the information and to sign the transaction. The remaining responses, such as error messages, are shown in a separate HTML container.



Sign Your Transaction

```
{
  "id": "77b0a6ed-9dea-4c82-b9ff-4e6ab10016dd",
  "operation": "upload",
  "file_name": "temp/225aaf5db82c72202636da3aa0f31945",
  "signature": "",
  "address": "0xdBbd29ff61C6D5730423285c8bc4422564029077",
  "json_data": {
    "cid": "bafkreiafypbfqkk4pqrugzsajavolw5pncpo7ict45ypbvol74caygofs4",
    "cmrUUID": "ba688b35-7b72-4fef-8647-4f1efc988719",
    "version": 1
  }
}
```

Wallet File

Fig. 4. Transaction Signature Popup Page

As previously discussed in Section 5.3, we developed a proprietary signature system that bypasses the necessity for external wallet managers. Figure 4 presents a popup with information about the file or operation that the user is executing and requests an encrypted wallet file and the passphrase to decrypt it. The decryption of the user's wallet and the signature of the transaction are made entirely offline and on-browser.

6. Evaluation and Discussion

Client applications access our proof-of-concept by calling its REST API. Every operation on the API ultimately involves at least one interaction with the smart contract. We conducted performance tests simulating real users making requests through a client application, using Postman as the API client platform to execute these requests. Table 2 presents the minimum, average, and maximum duration times, in seconds, for the listed and implemented requirements, with each request being executed ten times. Indented entries refer to parts of a request, such as temporarily storing information of a "Store CMR" operation and then submitting the stored information to the blockchain. This is due to the need to divide the operation into two parts to accommodate user input between them.

Table 2. API requests response times, in seconds.

Request	**Min.**	**Avg.**	**Max.**
1) Create Employee	0,94	1,50	2,31
2) Create Company	0,90	1,21	1,61
3) Associate Employee with Company	1,70	2,42	3,12
4) Store/Update CMR	8,86	9,47	10,07
Temporary Store/Update Request	0,36	0,60	0,96
Submit Store/Update Request	8,50	8,88	9,11
5) CMR View Request	8,79	9,63	10,15
Temporary Document View Request	0,47	0,79	1,04
Submit Document View Request	8,32	8,83	9,11
6) CMR Sign Request	8,13	9,03	9,74
Temporary Sign Request	0,51	0,98	1,32
Submit Sign Request	7,62	8,05	8,42
7) Retrieve CMR	0,72	1,16	1,60
8) Restrict Company	1,00	1,69	2,34
9) Restrict Employee	0,94	1,35	1,85

Despite being reasonable within the context of blockchain-based systems, the results from two-part operations seem to fall out of the scope of the commonly accepted 2-second threshold for user waiting times in web systems [24]. This is because operations 4), 5), and 6) wait for the transaction to be mined before showing a response to the user, guaranteeing that the transaction is fully submitted and final on the blockchain. For comparison, we also tested the response times for operation 4) without this confirmation, as shown in Figure 5.

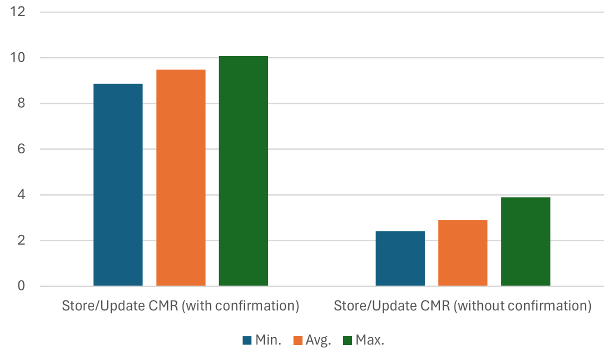


Fig. 5. Comparison between operations with and without confirmation, in seconds.

A consistent decrease in the request duration is evident in Figure 5, where the operation presented on the left, on average, is 3.26x times slower. However, a transaction only achieves finality once it is mined and included in a block, at which point it is considered irreversible. This helps prevent fraudulent activities by ensuring the transaction cannot be altered; hence, there is a need to confirm. The remaining presented response times are within acceptable limits, offering users a responsive experience. It is also worth mentioning that upon analyzing the only available reading request, the average duration of retrieving a CMR file is 1,16 seconds. This guarantees a fast interaction between our solution and outside actors (e.g., authorities).

Throughout the development of our eCMR system, we conducted four evaluations involving project partners, industry experts, and logistics stakeholders. We also held eight bi-weekly meetings, to gather insights and feedback, which served as the driving force behind redefining several system operations. Additionally, our developed concept and architecture are being adapted by other project partners, who are currently developing an electronic Bill of Lading (eBL) solution in a related work package. The insights gained from the eCMR system created synergies with the eBL system development, ensuring that similar challenges related to regulatory compliance and stakeholder acceptance are addressed.

7. Conclusion

This paper presents the design and prototype deployment of a blockchain-based eCMR management system, analyzing stakeholder requirements, architecture, and assessing efficiency and suitability for participating logistic companies. Powered by a public blockchain system, our solution ensures access control, secure upload, and retrieval of CMR documents via integration with off-chain external storage and electronic document signature.

Important limitations must also be stated. Following the framework proposed by [13], first, concerning Input Knowledge and Technology, blockchain is still nascent in CMR implementations, and there is a lack of similar studies in the literature, requiring us to expand our review to adjacent fields. Second (research process), we obtained encouraging feedback from experts participating in the project, but the evaluation was more focused on efficiency. Third, the resulting artifact is still at the laboratory stage and only addresses a part (file management) of eCMR deployment challenges. Finally, (4) design knowledge does not consider a comparison with non-blockchain alternatives, and the transferability is restricted to eCMR solutions requiring a third-party API for their integrated logistics solution.

Deploying the solution on a Polygon mainnet is the first step of our future work plan. This deployment will allow real-world validation and performance assessments under typical operational conditions. Additionally, to ensure scalability, the solution should undergo extensive testing with a larger sample of users to identify potential issues in diverse real-world scenarios and fully understand user interactions and system response time. Finally, the solution should be tailored to comply with electronic signature regulations across all European countries. Future research may also compare the social (e.g., market acceptance) and technical (e.g., performance, security, usability) aspects of blockchain and non-blockchain-based solutions for eCMR advances.

Acknowledgments

This research was funded by Project “Agenda Mobilizadora Sines Nexus”. ref. No. 7113, supported by the Recovery and Resilience Plan (PRR) and by the European Funds Next Generation EU, following Notice No. 02/C05-i01/2022, Component 5 - Capitalization and Business Innovation - Mobilizing Agendas for Business Innovation. It was also funded through national funds by FCT - Fundação para a Ciência e a Tecnologia, I.P., in the framework of the Project UIDB/00326/2020 and UIDP/00326/2020.

References

1. Ângelo, A., Barata, J., Da Cunha, P.R., Almeida, V.: Digital transformation in the pharmaceutical compounds supply chain: Design of a service ecosystem with e-labeling. Lect. Notes Bus. Inf. Process. 299 307–323 (2017)
2. A.P. Moller, Maersk: Maersk and IBM to discontinue TradeLens, a blockchain-enabled global trade platform.
3. Babu, E.S., Yadav, B.V.R.N., Nikhath, A.K., Nayak, S.R., Alnumay, W.: MediBlocks: secure exchanging of electronic health records (EHRs) using trust-based blockchain network with privacy concerns. Clust. Comput. 26 (4), 2217–2244 (2023)
4. Barata, J., Cunha, P.R. da: Mobile Supply Chain Management: Moving Where? Proc. 13th Eur. Mediterr. Middle East. Conf. Inf. Syst. EMCIS. (2016)
5. Bavassano, G., Ferrari, C., Tei, A.: Blockchain: How shipping industry is dealing with the ultimate technological leap. Res. Transp. Bus. Manag. 34 100428–100428 (2020)
6. Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H.F., Secret, A.: The World-Wide Web. Commun. ACM. 37 (8), 76–82 (1994)
7. vom Brocke, J., Fettke, P., Gau, M., Houy, C., Morana, S.: Tool-Support for Design Science Research: Design Principles and Instantiation. SSRN Electron. J. (2017)
8. vom Brocke, J., Hevner, A., Maedche, A.: Introduction to Design Science Research. 1–13 (2020)
9. Chainlink: Top 6 Smart Contract Languages in 2024.
10. Chauhan, G., Patel, B., Prajapati, N., Raj, S., Gadre, S., Patel, S.: Pharmaceutical supply chain management system using Blockchain and IoT technology. In: Intelligent Green Communication Network for Internet of Things. pp. 87–96. (2023)
11. Collect + Go: Collect + Go, https://collectgo.eu/en/home-en/, Accessed: June 17, 2024, (2021)
12. Craddock, A. (Management consultant), DSDM Consortium: The DSDM Agile Project Framework. 176–176
13. Cunha, P.R. da, Soja, P., Themistocleous, M.: Blockchain for development: a guiding framework. Inf. Technol. Dev. 27 (3), 417–438 (2021)
14. Daniel, E., Tschorsch, F.: IPFS and Friends: A Qualitative Comparison of Next Generation Peer-to-Peer Data Networks. IEEE Commun. Surv. Tutor. 24 (1), 31–52 (2022)
15. Desmarais, N., Paul, S.K., Simmonds, A.: Chaos-XML and Its Potential for ECommerce. Grain. 11 (2), 34–34 (1999)
16. Emmelhainz, M.A.: Edi: A Total Management Guide. (1993)

17. European Commission, Directorate-General for Mobility and Transport: EU transport in figures – Statistical pocketbook 2023. Publications Office of the European Union (2023)
18. Eurostat: Enterprises by detailed NACE Rev.2 activity and special aggregates [dataset]. https://doi.org/10.2908/SBS_OVW_ACT. (2023)
19. Finke, C., Graf, L., Guthahn, S., Leidecker, L., Masoldt, A., Schumann, M.: Design and Implementation of a Distributed Ledger Technology Platform to Support Customs Processes within Supply Chains. ACIS 2023 Proc. (2023)
20. Jovanovic, M., Kostić, N., Sebastian, I.M., Sedej, T.: Managing a blockchain-based platform ecosystem for industry-wide adoption: The case of TradeLens. (2022)
21. Jović, M., Filipović, M., Tijan, E., Jardas, M.: A Review of Blockchain Technology Implementation in Shipping Industry. Pomorstvo. 33 (2), 140–148 (2019)
22. Loklindt, C., Moeller, M.-P., Kinra, A.: How Blockchain Could Be Implemented for Exchanging Documentation in the Shipping Industry. (2018)
23. Ministry of Economic Affairs and Communications of Estonia: DIGINNO-PROTO, *DIGINNO BSR*, https://www.diginnobsr.eu/diginno-proto, Accessed: June 17, 2024
24. Nah, F.F.H.: A study on tolerable waiting time: how long are Web users willing to wait? Behav. Inf. Technol. 23 (3), 153–163 (2004)
25. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System, https://bitcoin.org/bitcoin.pdf, Accessed: April 09, 2024
26. Open Logistics Foundation: Open Logistics Foundation develops European eCMR open-source standard, https://openlogisticsfoundation.org/open-logistics-foundation-develops-european-ecmr-open-source-standard/, Accessed: April 10, 2024
27. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. J. Manag. Inf. Syst. 24 (3), 45–77 (2007)
28. Pionira NV: Pionira, https://www.pionira.be/en/, Accessed: April 09, 2024
29. Polygon Labs: Polygon, https://polygon.technology/, Accessed: April 15, 2024
30. Protocol Labs: Filecoin: A Decentralized Storage Network. (2017)
31. Rani, P., Sachan, R.K., Kukreja, S.: Educert-chain: a secure and notarized educational certificate authentication and verification system using permissioned blockchain. Clust. Comput. (2024)
32. Storj Labs, Inc.: Storj: A Decentralized Cloud Storage Network Framework. (2018)
33. Tomicová, J., Poliak, M., Zhuravleva, N.A.: Impact of using e-CMR on neutralization of consignment note. Transp. Res. Procedia. 55 110–117 (2021)
34. TransFollow: Uniting Supply Chains, https://www.transfollow.org/, Accessed: April 09, 2024
35. Underwood, S.: Blockchain beyond Bitcoin. Commun. ACM. 59 15–17 (2016)
36. United Nations: Additional Protocol to the Convention on the Contract for the International Carriage of Goods by Road (CMR) concerning the Electronic Consignment Note. (2008)
37. United Nations: Convention on the Contract for the International Carriage of Goods by Road (CMR), https://unece.org/DAM/trans/conventn/cmr_e.pdf, (1956)
38. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. pp. 1–10. Association for Computing Machinery, New York, NY, USA (2014)
39. Working Party on Road Transport: Operations of future eCMR system Comparison with services provided by Pilot projects. (2021)
40. Yaga, D., Mell, P., Roby, N., Scarfone, K.: Blockchain technology overview. National Institute of Standards and Technology (2018)
41. CargoLedger, https://cargoledger.nl/, Accessed: April 10, 2024
42. MetaMask, https://metamask.io/, Accessed: April 10, 2024