Using NFTs for Product Management, Digital Certification, Trading, and Delivery in the Healthcare Supply Chain

Ahmad Musamih, Ibrar Yaqoob, *Senior Member, IEEE*, Khaled Salah, Raja Jayaraman, Mohammed Omar, and Samer Ellahham

***Abstract*—The COVID-19 pandemic caused disruption in the healthcare market, which resulted in shortages of essential healthcare products. The pandemic exacerbated the problems that already existed in the healthcare supply chain, such as poor data availability, transparency, and accessibility. Such problems necessitate the introduction of a solution that is capable of managing products, maintaining ownership, facilitating trading, and delivering products in a traceable, auditable, and trustworthy manner. In this article, we propose a nonfungible-token-based solution for the management of healthcare products, where the ownership of a product is maintained by using digital certification, the trade and delivery of healthcare products is facilitated by a smart contract, and disputes are settled by an arbitrator while keeping all related information on-chain for auditing purposes. We utilize the Interplanetary File System to store the metadata of healthcare products to avoid storing large-sized data on the blockchain. We present system diagrams and algorithms along with the implementation details. We conduct security testing to demonstrate that our solution is resilient and secure against common vulnerabilities and exploits. We compare our solution with the existing solutions to show its distinctive features and novelty. The smart contracts code is made publicly available on GitHub.**

***Index Terms*—Blockchain, data ownership, digital certification, digital ownership, Ethereum, healthcare, nonfungible tokens (NFTs), smart contracts, supply chain.**

I. Introduction

THE healthcare industry is one of the most rapidly growing industries worldwide [1]. According to the World Health Organization, the availability and accessibility of healthcare products are considered important factors in any healthcare system [2]. Moreover, having a robust healthcare system is essential for every country to improve its impact on health outcomes [3], [4]. Early in 2020, when COVID-19 became a global pandemic, it showed how important healthcare products are to our healthcare system and to the health of the public. During the pandemic, shortages of healthcare products, such as vaccines and personal protective equipment (PPE), became very severe [5]. According to a survey conducted in the United States by the National Nurses United union, 87% of the nurses reported that they had to reuse their single-use PPE [6]. In addition, 27% of the nurses reported that they were exposed to patients that were confirmed to have COVID-19 without having proper PPE [6].

Healthcare product demand forecast is essential for predicting the needs of each healthcare provider ahead of time to avoid shortages and overordering [7]. Demand forecasting in healthcare is defined as the continuing process of predicting the healthcare product that will be purchased as well as its quantity, purchaser information, delivery date, and delivery location. The decision makers in the healthcare industry depend heavily on information about demand in every stage of the healthcare supply chain, which can provide them with the necessary information to shape the future market and design a robust and proactive healthcare system [8].

Fig. 1 represents a typical flow diagram for healthcare products supply chain and future demand forecasting mechanism. First, the manufacturer produces healthcare products based on the available results from the Data Analytics Solution and historical data. Second, the transporter/3PL picks up the produced healthcare products and delivers them to the desired warehouse. Third, another transporter/3PL picks up the healthcare products from the warehouse and delivers them to the healthcare center. Fourth, the inventory management system fetches the sales orders information from healthcare centers and updates the stock levels in the Data Analytics Solution. Finally, the data analytics solution fetches sales orders from healthcare centers and performs demand forecasting analysis to predict future demand for healthcare products.

Given the complexity of the healthcare supply chain, forecasting healthcare product demand can be very challenging. One of the common challenges for healthcare product forecasting is the inability to gain access to data because the supply chain is fragmented and inefficient in data management, which is

Manuscript received 23 June 2022; revised 17 August 2022 and 26 September 2022; accepted 13 October 2022. This work was supported by the Khalifa University under Award CIRA-2019-001. Review of this manuscript was arranged by Department Editor T.-M. Choi. *(Corresponding author: Ibrar Yaqoob.)*

Ahmad Musamih, Raja Jayaraman, and Mohammed Omar are with the Department of Industrial and Systems Engineering, Khalifa University, Abu Dhabi 127788, United Arab Emirates (e-mail: ahmad.musameh@ku.ac.ae; raja.jayaraman@ku.ac.ae; mohammed.omar@ku.ac.ae).

Ibrar Yaqoob and Khaled Salah are with the Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi 127788, United Arab Emirates (e-mail: ibraryaqoob@ieee.org; khaled.salah@ku.ac.ae).

Samer Ellahham is with the Heart, Vascular and Thoracic Institute, Cleveland Clinic Abu Dhabi, 112412 Abu Dhabi, United Arab Emirates (e-mail: ellahas@clevelandclinicabudhabi.ae).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TEM.2022.3215793.

Digital Object Identifier 10.1109/TEM.2022.3215793

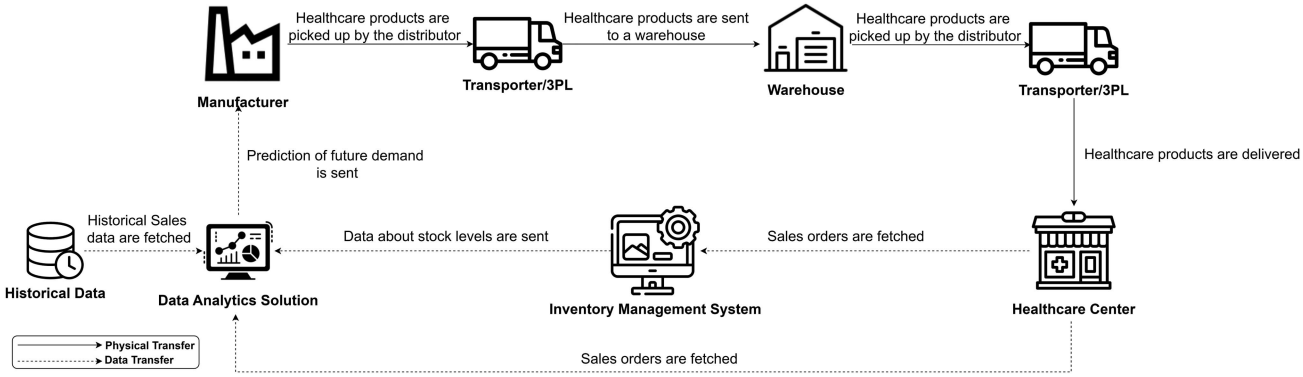


Fig. 1. Flow diagram showing a typical healthcare product supply chain [9].

essential to perform proper forecasting. Moreover, this issue is exacerbated by the lack of accountability in the healthcare supply chain, where the involved entities are not held accountable for the missing data and information [10]. For example, the Food and Drug Administration agency tried to approach 1000 manufacturers to request information about the healthcare supply chain during the COVID-19 pandemic, and only about one-third responded [5]. This shows how the healthcare supply chain is fragmented and difficult to access. Moreover, the weak links in the health products supply chain offer malicious users a backdoor which they can exploit to introduce counterfeit healthcare products into the supply chain [11]. The inaccessibility of data makes it very challenging to sustain the ownership of healthcare products and perform data analytics for demand forecasting. Therefore, a more robust solution is needed to ensure accountability, traceability, data provenance, data ownership, and data availability.

Nonfungible tokens (NFTs) have drawn a lot of attention and experienced massive growth during 2020, when their value reached around 250 million USD [12], [13]. The main industries that NFTs have disrupted are gaming, collectibles, fashion, and art [14]. Those industries have greatly benefited from the advantages of NFTs, such as ownership verification, transferability, authenticity, unlocking new revenue streams, and bringing creators from different fields into one ecosystem [15]. Inspired by these observations, we present a potential use case for NFTs in the healthcare supply chain, in which healthcare products are managed, traded, and delivered based on NFT standards. In particular, the use of NFTs in this specific application ensures that for any given healthcare product that is flowing through the supply chain at any given time, the ownership and authenticity of the product can be verified because such information is permanently stored on the blockchain and is always accessible. Moreover, the provenance of products is guaranteed since all the transactions the product is involved in are permanently stored on the blockchain and are also immutable. A successful and effective implementation of such a solution to the traditional healthcare supply chain is deemed crucial because in its current state, the only way to inspect the provenance of products is by approaching the suppliers and distributors and requesting the desired information, and there is currently no effective method to ensure that the provided information from those entities is accurate. In addition, the current approaches suffer from the issue of having information scattered across different data sources, making them susceptible to errors, hence becoming untrustworthy [16]. The main contributions of this article are as follows.

1) We propose an NFT-based solution that is decentralized, transparent, traceable, auditable, reliable, and secure to provide easy access for participants to healthcare products and offer decision makers a data repository for healthcare products to improve their demand forecasting results.
2) We integrate the decentralized storage of the Interplanetary File System (IPFS) with the NFT-based solution to ensure that the metadata of each NFT is permanently stored and to avoid storing large-sized files on the blockchain.
3) We present four phases for the proposed solution, which are minting and listing NFTs, purchasing and redeeming NFTs, delivering NFTs physical counterpart, and disputing NFTs purchases by utilizing a system architecture, sequence diagrams, entity-relationship diagram, and five algorithms.
4) We develop smart contracts to facilitate and represent the logic behind each phase of the proposed solution. We implement, test, and validate the smart contracts and design a front-end decentralized application (DApp) to interact with the smart contracts. Smart contract code is made publicly available on GitHub.[1]
5) We perform security analysis and describe how the proposed solution can be generalized and extended to fit the needs of other applications.

The rest of this article is structured as follows. Section II presents the related work focused on data ownership issues. Section III describes the design of the proposed NFT-based solution for healthcare products. Section IV describes implementation details. Section V presents testing and validation details of the smart contracts. Section VI presents the discussion and analysis of the proposed solution. Finally, Section VII concludes this article by summarizing our main findings and contributions.

[1]https://github.com/DrugTraceability/HealthcareNFTs

TABLE I
TARGETED DOMAIN AND MAIN OBJECTIVE OF THE RELATED WORK AND OUR WORK

Studies	Domain				Objectives
	Traceability	Data Management	Access Control	Data Ownership	
[17]	✓				Developing a blockchain-based solution for data traceability in blood donation supply chain
[18]	✓				Tracking drug prescriptions in the pharmaceutical supply chain
[19]	✓				Verifying the authenticity of healthcare products by tracing their origin
[20]		✓			Providing a decentralized data management system for mobile healthcare
[21]		✓			Granting healthcare data owners with data management solution for secure data sharing
[22]		✓			Providing a solution to the issue of fragmented patient's medical records
[23]			✓		Eliminating the need for a third party to access healthcare data
[24]			✓		Allowing patients to access their medical records at any time
[25]			✓		Providing patients with the ability to grant and revoke access to their medical records
[26]				✓	Proposing an NFT-based approach that grants patients ownership over their consent records
[27]				✓	Proposing an NFT-based approach that models patients' medical records as an NFT
Our Work	✓	✓	✓	✓	Designing and implementing an NFT-based solution for medical data traceability, management, sharing, and ownership

II. RELATED WORK

In this section, we present the existing blockchain-based solutions proposed for addressing healthcare issues. We classify the existing literature into four categories: traceability, data management, access control, and data ownership. Table I shows a comparison between our work and the existing works based on objectives and the targeted domain.

A. Traceability

Sadri et al. [17] propose a blockchain-based solution for data traceability in the blood donation supply chain. The proposed solution addresses the poor visibility issue in the blood donation supply chain, where some critical information is not stored properly. The developed smart contracts for the proposed solution are written in the Solidity language using the REMIX IDE environment. Another traceability solution is presented in [18] where Chenthara et al. propose a permissioned blockchain framework for the tracking of drug prescriptions by using Hyperledger Fabric, in addition to IPFS for decentralized data storage for large files. Moreover, smart contracts are designed to facilitate the prescription process while recording all the necessary details. Another traceability solution is proposed by Chronicled, a technology company that releases improvements to the network and develops solutions on top of it, which is called "The MediLedger Network," and was established in 2019. This solution provides its users with the ability to verify the authenticity of healthcare products by leveraging a private permissioned blockchain-based network that is run by participants in the industry [19].

B. Data Management

HealChain is a decentralized data management system for mobile healthcare. The authors leverage consortium blockchain technology to introduce HealChain. Moreover, the proposed solution is composed of three layers, which are data collection, verification, and storage. Finally, the proposed solution uses IPFS as a means for off-chain storage [20]. Another data management solution is proposed in [21], where Asad et al. present a permissioned blockchain for secured healthcare data sharing. The data owners are granted full control over their data while maintaining data integrity. Finally, the proof-of-authority (PoA) consensus algorithm is leveraged to enhance scalability and throughput. Similarly, Vardhini et al. [22] propose a blockchain-based solution that addresses the fragmented medical records issue. The goal is to provide secure access to the patient's medical records without the need for a third party. Furthermore, other stakeholders will have to request access to the medical records from the patient rather than the healthcare provider. Finally, the proposed solution is implemented by using Hyperledger Fabric, which is a permissioned blockchain.

C. Access Control

Ramyasri and Hussain [23] present a blockchain-based solution for the access control of healthcare data without the need for a central authority or a third party. The proposed solution is composed of only two phases, which are the registration of entities and accessing healthcare data. A similar solution is called MedRec, which is a decentralized record management system to handle electronic medical records (EMRs) by leveraging blockchain technology. The proposed solution is built on a proof-of-work Ethereum blockchain to incentivize participation in the network via mining rewards. The proposed solution allows patients to have control over their data and access it at any time [24]. Similarly, Younis et al. [25] propose a blockchain-based solution that enables patients to grant and revoke access permissions for their medical data. Moreover, the proposed solution uses cloud storage for sensor data storage and blockchain for access control and session logs. In addition to that, smart contracts are used to facilitate interactions between patients and the cloud storage to mitigate the risk of fraud and identity theft. Finally, the security of the proposed system is verified by using the AVISPA tool set.

D. Data Ownership

Cunningham et al. [26] propose an NFT-based system architecture that allows patients to have control over their medical records, which makes them able to grant access to their records only when necessary. The proposed architecture allows legitimate consumers to apply for consent to obtain data from medical data providers, and this process can be performed once only. NFTs are used in this architecture to bundle the consents and transmit them between data consumers and providers. This implementation eliminates the need for reliance on a trusted third party to verify the legitimacy of the consent. Another NFT-based approach that addresses the data ownership issue in healthcare is represented in [27], where the authors propose an approach in

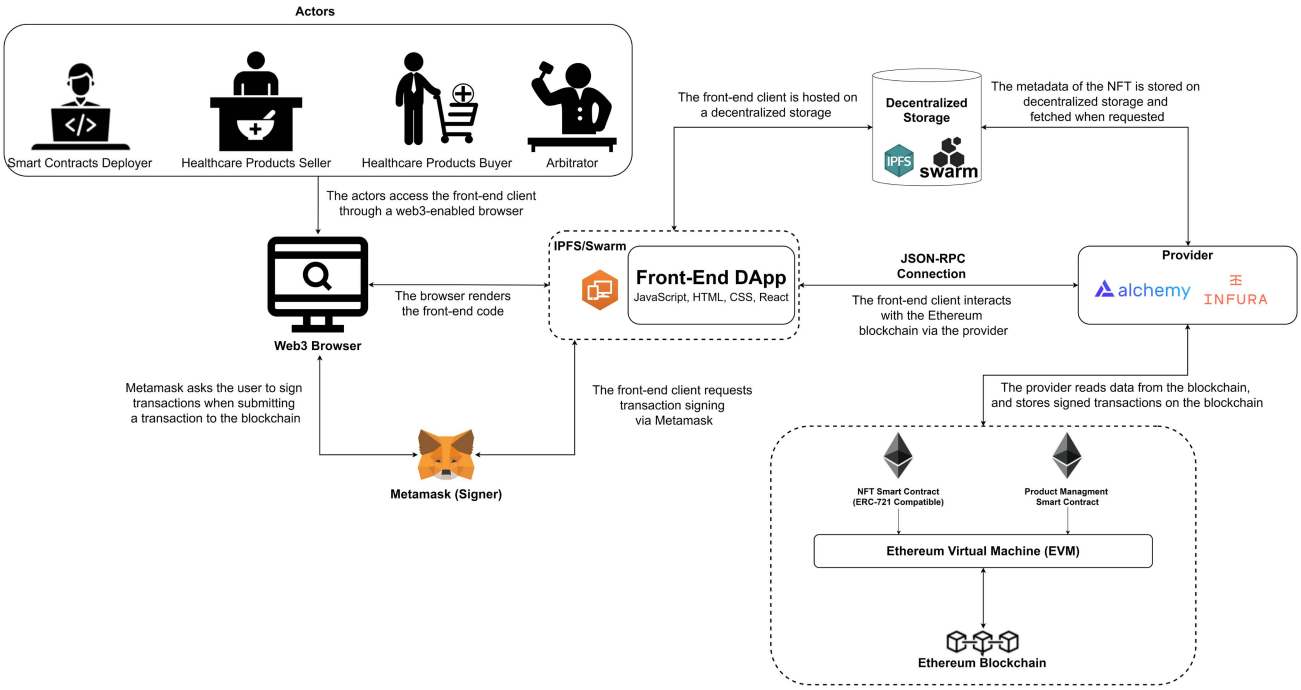


Fig. 2. High-level system architecture of the proposed NFT-based solution for healthcare products supply chain.

which the personal medical data are modeled as an NFT, and only the owner of the NFT has the right to share or trade these data with other entities. The proposed approach consists of several smart contracts that facilitate the creation and management of NFTs.

To the best of our knowledge, none of the existing work has designed and implemented an NFT-based solution to ensure traceability, data management, access control, and data ownership of healthcare products, as we did in our study. Also, we describe how our proposed solution can be generalized and extended to fit the needs of other objects in healthcare, such as medical records, as well as objects in other industries. Moreover, we fully implement the front end and back end of the proposed solution to ensure that it works as intended. Furthermore, we test and validate the functionality of our developed smart contracts. Finally, we identify the main challenges and limitations of our proposed solution.

III. Proposed Solution Details

In this section, we propose an NFT-based solution for the management of products within the healthcare supply chain. In particular, our solution aims to improve the overall coordination of the healthcare supply chain by ensuring that the ownership of healthcare products is directly controlled by the entity that owns them. Moreover, our solution provides participants with end-to-end transparency because all the details and transactions associated with a product are permanently stored on the blockchain. Furthermore, the utilization of blockchain technology provides the involved entities with an effective way to share information among each other, which eliminates the need for multiple centralized databases that usually fragment the data and make it inaccessible, and allows equitable access for all the involved entities. Finally, at any given point in time, the owner of the healthcare product can be easily identified, allowing regulators to impose accountability measures and standards.

A. Main Components of the Proposed Solution

Fig. 2 represents a high-level system architecture of the proposed solution. The system architecture is composed of three main components, which are the actors, the front-end layer, and the back-end layer. Further explanation of each component is given as follows.

1) *Actors:* The main actors in the proposed solution are the smart contracts deployer, healthcare products seller, and healthcare products buyer, and the arbitrator. Only authorized actors are allowed to participate in the NFT-based solution.
2) *Front-end layer:* The front-end layer provides actors with the required interface to perform their tasks. The front-end layer is basically a JavaScript, HTML, and CSS script that allows the actors' web3 browsers to render it into a readable and understandable format. Moreover, web3 browsers are connected to a signer, such as Metamask, which is necessary to sign transactions and relay them to the blockchain.
3) *Back-end layer:* The back-end layer is composed of the web3 provider, Ethereum blockchain, and decentralized storage. First, web3 provider allows the front-end layer to interact with the Ethereum blockchain without setting a full node locally, which can be expensive and complicated. Second, the Ethereum blockchain is the cornerstone of the back-end layer, and it hosts the NFT and the Product

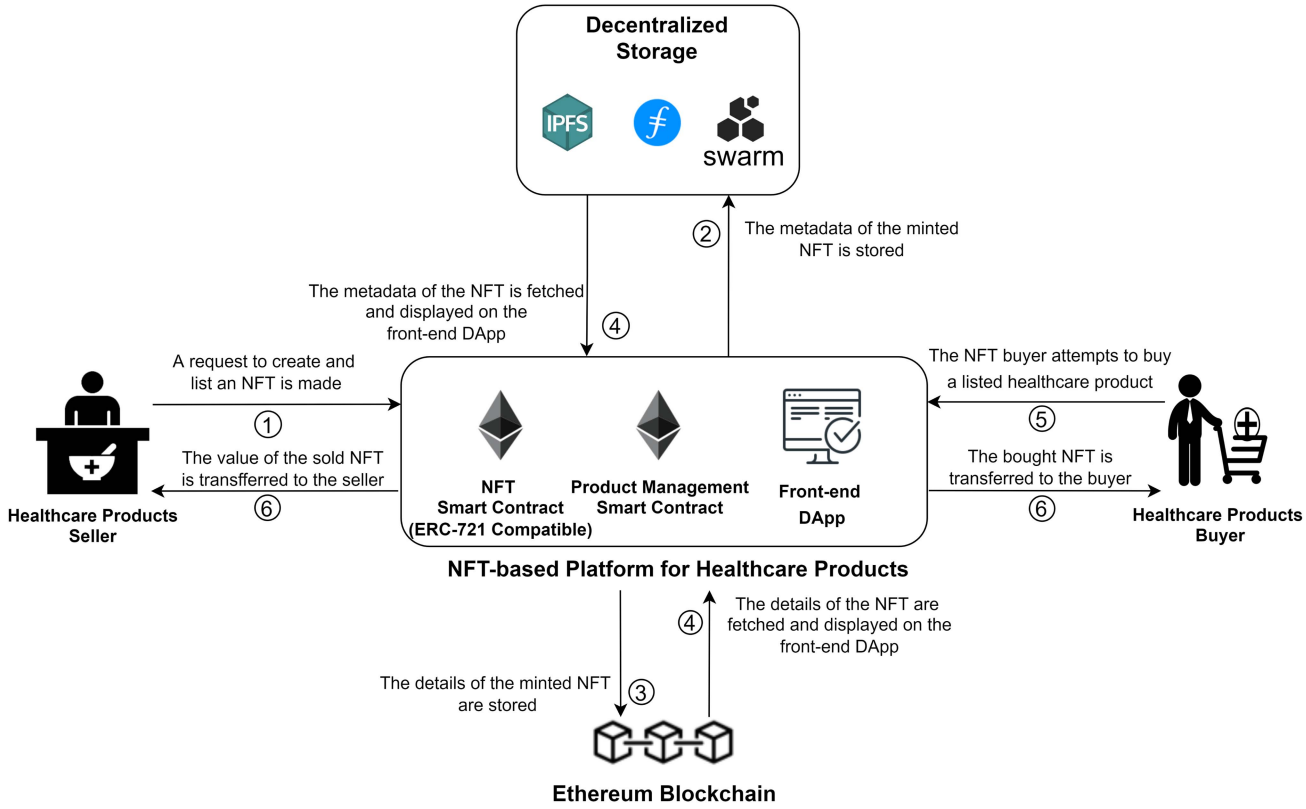


Fig. 3. Workflow of using NFTs for managing ownership and trading healthcare products.

Management smart contracts. All transactions, logs, and events are permanently stored on the blockchain. Finally, decentralized storage, such as IPFS, allows actors to store the large-sized metadata of their healthcare products outside the Ethereum blockchain [28].

Fig. 3 illustrates the typical workflow of tokenizing physical assets and trading them as NFTs. First, the healthcare product seller uploads the metadata of the product to the IPFS, which returns a Uniform Resource Identifier (URI). Second, the healthcare product seller accesses the minting and listing function, which mints a new NFT through the NFT smart contract and lists it for the price that the seller sets. Third, any interested buyer will be able to view the listed products on the front-end DApp and attempt to purchase the item. Finally, if the purchase attempt is successful, the NFT is transferred to the buyer and the seller receives its value. Our proposed solution adopts a similar workflow, but it is adjusted to fit the requirements of our solution. For example, the seller can only get the value of the NFT if it is delivered to the buyer or if the time to redeem it has passed.

B. NFTs, Tokenization, and Digital Twins

NFTs are an integral component of our proposed solution. NFTs are unique cryptographic tokens that reside on the blockchain, and they cannot be replicated. Moreover, the ownership of the NFT is recorded on the blockchain and the owner has full control over the NFT [15]. Furthermore, NFTs usually represent physical assets from the real world, and this is achieved by containing references to digital files, such as images, which is referred to as a digital twin [30]. Utilizing blockchain technology to store the details of digital twins is done through a process called tokenization [31]. Any tokenized asset can be categorized as either a fungible token or NFT. In the former, multiple copies of the same asset with the exact same attributes can exist at the same time. Therefore, they can be considered interchangeable and, hence, called fungible. In the latter, only one asset with a unique set of attributes can exist at the same time and cannot be replicated; therefore, it is called an NFT. In our solution, NFTs are coded on the Ethereum blockchain, which standardizes the use of NFTs through a standard called ERC-721 (Ethereum Request for Comment), which was introduced based on EIP-721 (Ethereum Improvement Proposal) [33]. This standard provides the needed functionality to track and transfer NFTs.

C. Main Phases and Interactions of the Proposed Solution

The proposed solution comprises four main phases, namely, minting and listing NFTs, purchasing and redeeming NFTs, delivering NFTs physical counterpart, and disputing NFTs purchases. Each phase is described further as follows.

1) Minting and Listing NFT Phase: Fig. 4 illustrates all the interactions among the actors and the solution components during the minting and listing phase. First, the healthcare product seller uploads the image of the product to the IPFS, which returns a URI for the image. Then, the metadata for the product is uploaded to the IPFS, which includes information, such as the

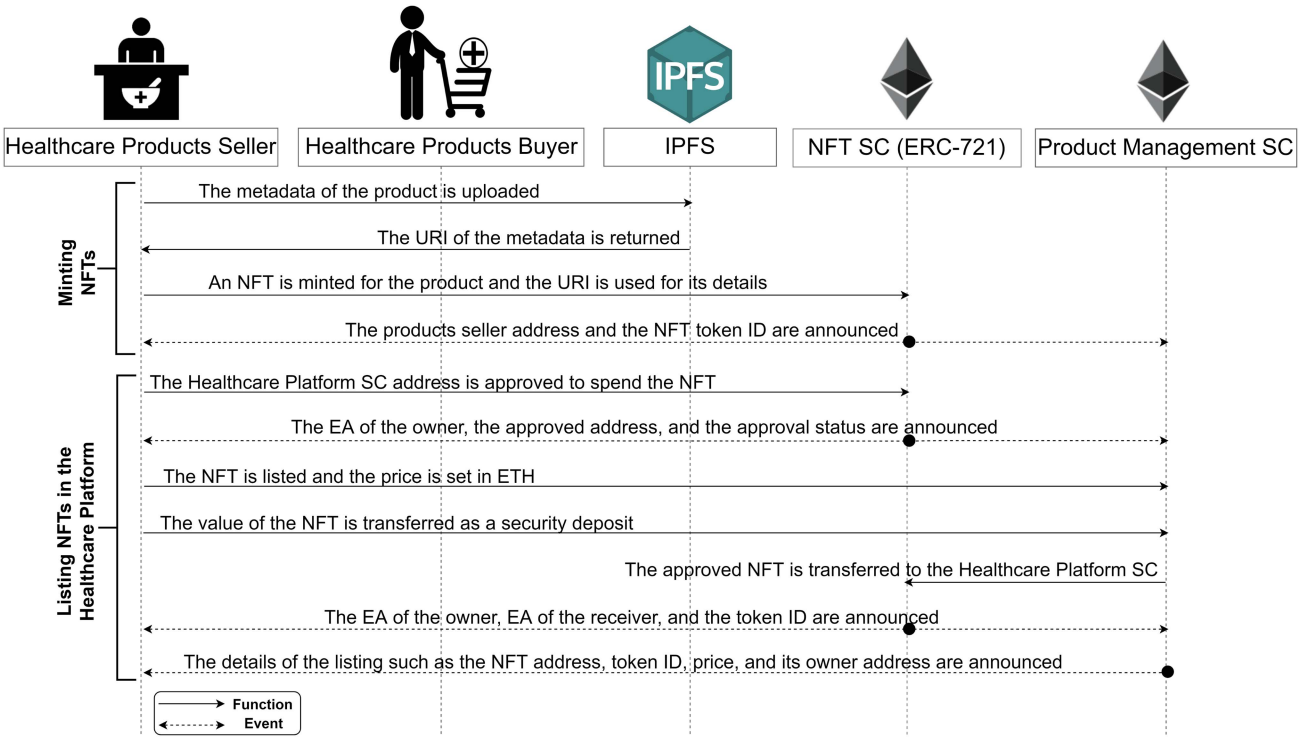


Fig. 4. Sequence diagram illustrating the deployment of smart contracts and minting and listing NFTs.

image URI, name, description, and price. Second, the healthcare product seller interacts with the NFT smart contract to mint the product as an NFT, which is done by providing the minting function with the product's URI. Third, the healthcare product seller interacts with the Healthcare Product smart contract to list the NFT and make it visible to other users. To list an item, the healthcare product seller needs to interact with the NFT smart contract to approve the address Product Management smart contract to use the healthcare product NFT. Then, the listing function is executed, which requires the seller to transfer the value of the NFT to the Product Management smart contract as a security deposit in case of a dispute between the buyer and seller later on. Moreover, the Product Management smart contract will transfer the NFT from the seller to its own address. Finally, an event is emitted with all the details of the listed NFT, such as the NFT address, token ID, price, and the seller's address.

2) Purchasing and Redeeming NFT Phase: Fig. 5 illustrates all the interactions during the NFT purchase and redemption phase. First, the healthcare product buyer will be able to view the listed healthcare products through the DApp. Second, the buyer will attempt to purchase a listed product by triggering the purchase function in the Product Management smart contract. If the buyer has a sufficient amount of Ether in his account to cover the cost of the listed NFT [29], the ownership of the NFT is transferred to the buyer and an event is emitted with the details of the purchase process. However, if the buyer has an insufficient amount of Ether in his account, the transaction will be reverted and rejected. Third, the buyer will have an option to redeem the NFT for its physical counterpart throughout a predetermined redemption period. If the buyer decides to redeem the NFT during the redemption period, the address of the Product Management smart contract must be approved again to spend the NFT because it has a new owner. Then, the redemption function is executed where the buyer is required to transfer the value of the NFT to the Product Management smart contract as a security deposit in case a dispute happens between the buyer and the seller, and finally, an event is emitted with all the necessary details of the redemption process. If the buyer tries to trigger the redemption process outside the redemption time window or without having sufficient Ether for the security deposit, the transaction will be reverted.

3) Delivering NFT Physical Counterpart Phase: Fig. 6 illustrates all the interactions during the delivery process of a redeemed NFT. In this phase, it is assumed that the healthcare product seller is responsible for delivering the healthcare product, and it is also assumed that once the NFT is redeemed, the healthcare product seller will have to deliver the product within a specific delivery time window. First, if the delivery time window is still open, the seller will start the delivery process in which the state of the NFT is updated to "EnRoute," which means that the product is out for delivery, and an event is emitted with the details of the product that is being delivered, which is fetched from the NFT metadata that is stored on the blockchain. Second, once the seller arrives at the location of the buyer, the buyer will be requested to sign a message off-chain confirming the reception of the product with the private key of the Ethereum address used in the Product Management smart contract. Once the message is signed, the seller will store the message and signature on the blockchain via the Product Management smart contract. Finally, if the signature of the buyer is valid, the seller will be able to

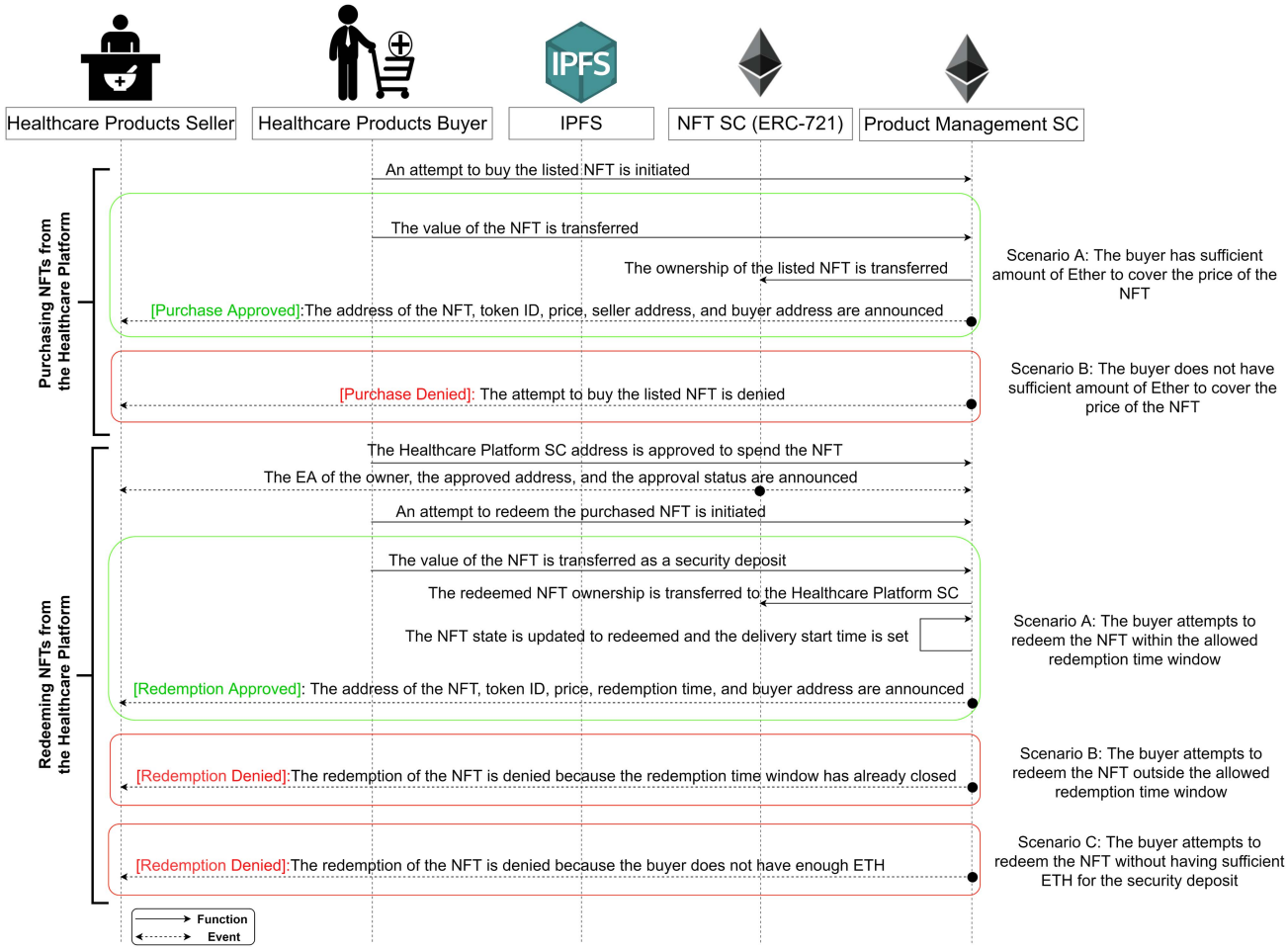


Fig. 5. Sequence diagram illustrating the process of purchasing and redeeming NFTs.

prove delivery of the product, end the delivery process, and claim the value of the NFT, which also includes the security deposit.

4) Dispute Settlement Phase: Fig. 7 illustrates all the interactions during the dispute settlement phase. The healthcare product buyer can open a dispute for any of the redeemed products as long as the redemption time window is not closed. First, the buyer will upload evidence supporting the dispute on the IPFS, which returns a URI for the metadata. Then, the seller will trigger the dispute function in the Product Management smart contract and use the URI as an input for the function. Once the function is successfully triggered, the state of the NFT is updated to "Disputed" and an event is emitted with the details of the opened dispute. Second, the seller will have a chance to challenge the dispute. The challenge is done by uploading the supporting document to the IPFS, which returns a URI for the metadata. Then, the seller will trigger the dispute challenge function in which the URI of the metadata is taken as an input, the state of the NFT is updated to "Challenged," and an event is emitted with the details of the dispute challenge. Finally, an arbitrator will make the final decision based on the provided documents, and it is assumed that there are three potential outcomes. The first decision would be to choose the seller as the winner of the dispute if the buyer rejected receiving the product for no valid reason, and this decision results in twice the value of the NFT being transferred to the seller, which comprises the security deposit of the buyer and the original value of the NFT; the NFT being transferred back to the seller; the NFT attributes being reset; and emitting an event that has the details of the dispute decision. The second decision would be to choose the buyer as the winner if the received product is proven to be damaged or expired, and this decision results in twice the value of the NFT being transferred to the buyer, the NFT being burned since it no longer represents a usable product, and an event is emitted with the details of the final decision. The third decision would be to choose the buyer as a winner if the seller failed to deliver the item on time. This decision results in twice the value of the NFT being transferred to the buyer, resetting the attributes of the NFT. The NFT is then transferred back to the seller, and an event is emitted with the details of the final decision. It should be noted that an event where a buyer denies receiving the product is excluded because the seller will hand over the product only if the buyer signs a message with a private key that proves the reception of the item, and the message and signature are both stored on-chain. Therefore, this accusation can be easily spotted and invalidated.

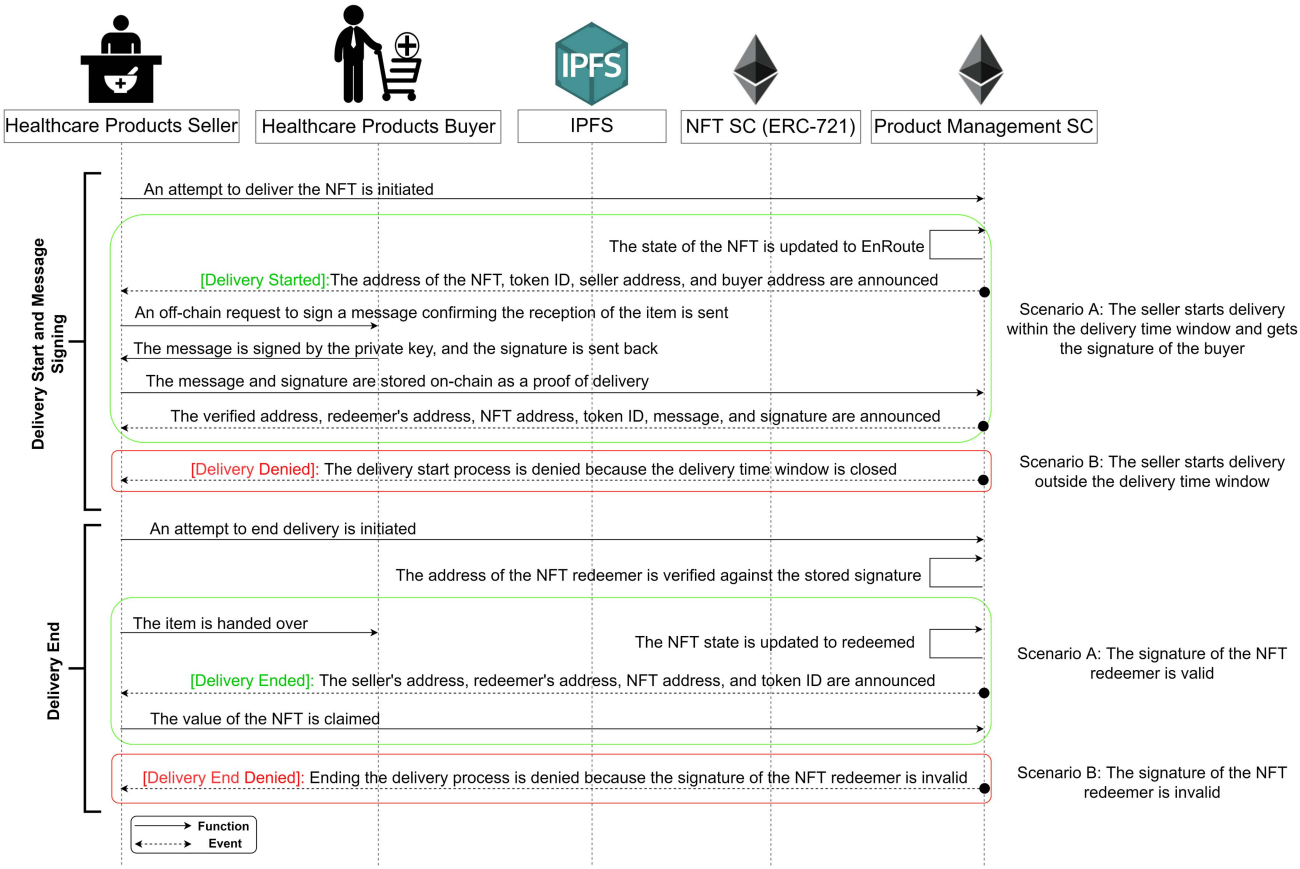


Fig. 6. Sequence diagram illustrating the process of delivering the physical counterpart of the NFT.

IV. Implementation Details

In this section, the implementation details of the proposed solution are described. The proposed solution consists of two main smart contracts: the NFT smart contract and the Product Management smart contract. The smart contracts are written in Solidity, which is a high-level language for implementing smart contracts [32], and compiled by Hardhat, which is a development environment to compile and deploy smart contracts. Finally, the front-end interface is developed by React.js, which is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications.

A. Smart Contract Description

In this subsection, the written smart contracts are described in detail. First, the NFT smart contract is written based on Ethereumś ERC-721 standard, a standard interface for NFTs that allows the implementation of a standard API for NFTs within smart contracts [33]. This smart contract facilitates the "mint," "approve," and "transfer" functions of NFTs while recording all the necessary details in the form of events.

Second, the Product Management smart contract is developed to govern the main phases of the proposed solution with the aid of the NFT smart contract. This smart contract comprises several functions. First, the minting and listing phase is facilitated by the "mint," "approve," and "transfer" functions from the NFT smart contract, and the "makeitem" function in the Product Management smart contract. Second, the purchase and redemption phase is facilitated by the "purchaseitem" and "RedeemNFT" functions. Third, the delivery phase is facilitated by the "startDelivery," "storeSignatures," "isValidSignature," and "ProofOfDelivery" functions. Finally, the dispute settlement phase is facilitated by the "OpenDispute," "ChallengeDispute," and "DisputeFinalDecision" functions. The full description of each function is made publicly available in the smart contract code on GitHub.[2]

Fig. 8 illustrates the entity-relationship diagram between the actors and the smart contracts. First, the healthcare products sellers interact with both the NFT smart contract and the Product Management smart contract. The relationship with the former is n to 1 because there can be multiple sellers within the proposed solution. However, there can only be one NFT smart contract with a unique address. Similarly, the relationship with the latter is also n to 1 because multiple sellers can interact with the same Product Management smart contract. On the other hand, the healthcare products buyers will have an n-to-1 relationship with the NFT smart contract because when an NFT is sold it will still be linked to the same NFT smart contract address, and the only thing that changes is the ownership. Similarly, the relationship between the buyers and the Product Management

[2]https://github.com/DrugTraceability/HealthcareNFTs

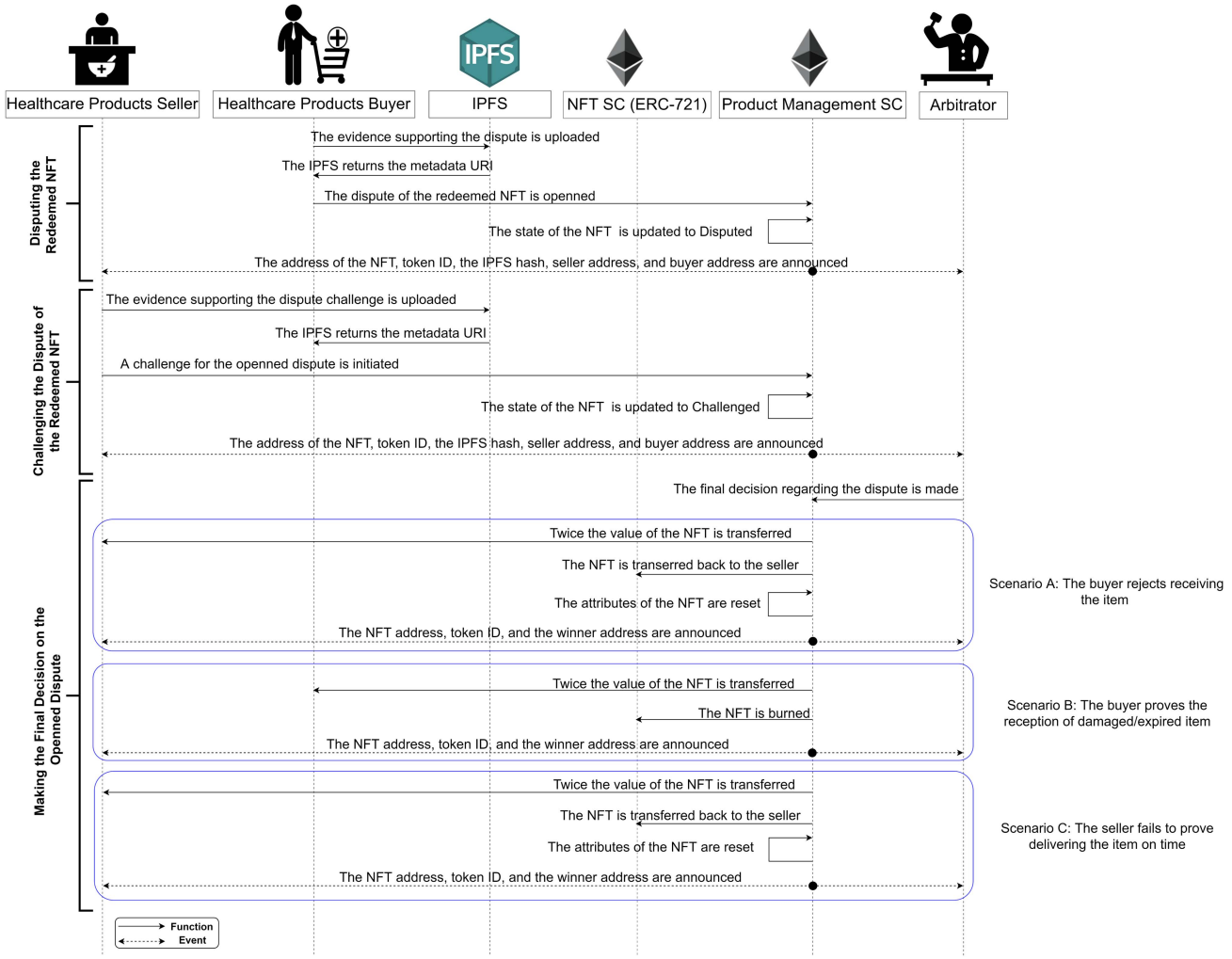


Fig. 7. Sequence diagram illustrating the process of disputing the redeemed NFTs.

smart contract is n to 1 because multiple buyers can interact with the same Product Management smart contract. Finally, in our solution, it is assumed that NFTs are only minted through our NFT smart contract. Therefore, the relationship between the NFT smart contract and Product Management smart contract is one to one. However, if the smart contract accepts NFTs minted from different NFT smart contracts, then the relationship between the Product Management smart contract and the NFT smart contract will be 1 to n.

B. Algorithms

The designed smart contracts are logic-based, and to further explain the logic behind each smart contract, we develop detailed algorithms, which are explained below.

Algorithm 1 represents the main steps of the NFT minting process, which consists of six main steps and one requirement. First, the entity that is interested in minting a new NFT that represents a healthcare product uploads the metadata of the product to the IPFS, which returns a unique immutable hash; then, the healthcare product specifies the price, name, and description of the product, which are then stringified with the IPFS hash to produce a unique URI for the product. Second, if the URI is valid, the token count within the NFT smart contract is increased by one. Third, the new token count is assigned to the newly minted NFT. Fourth, the produced URI is attached to the NFT. Fifth, the Ethereum address of the entity that minted the NFT is assigned as the owner of it. Finally, an event is emitted declaring the key details of the minting process, such as the address of the NFT owner and the token ID. However, if the provided URI to the minting function is invalid, the minting process is declined, and an error is shown to the user.

Algorithm 2 represents the NFT listing process, which consists of seven steps and three requirements. First, the entity that is interested in listing an NFT triggers the listing function within the Product Management smart contract. Second, the listing function checks for three requirements before executing. It requires the function caller to be the current owner of the NFT, the price of the NFT is greater than zero, and a value equivalent to the price of the NFT is transferred to the smart contract during the execution of the function as security deposit. Third, if all the requirements are fulfilled, the function caller approves the Product Management smart contract to handle the listed NFT, and an event is emitted with the details of

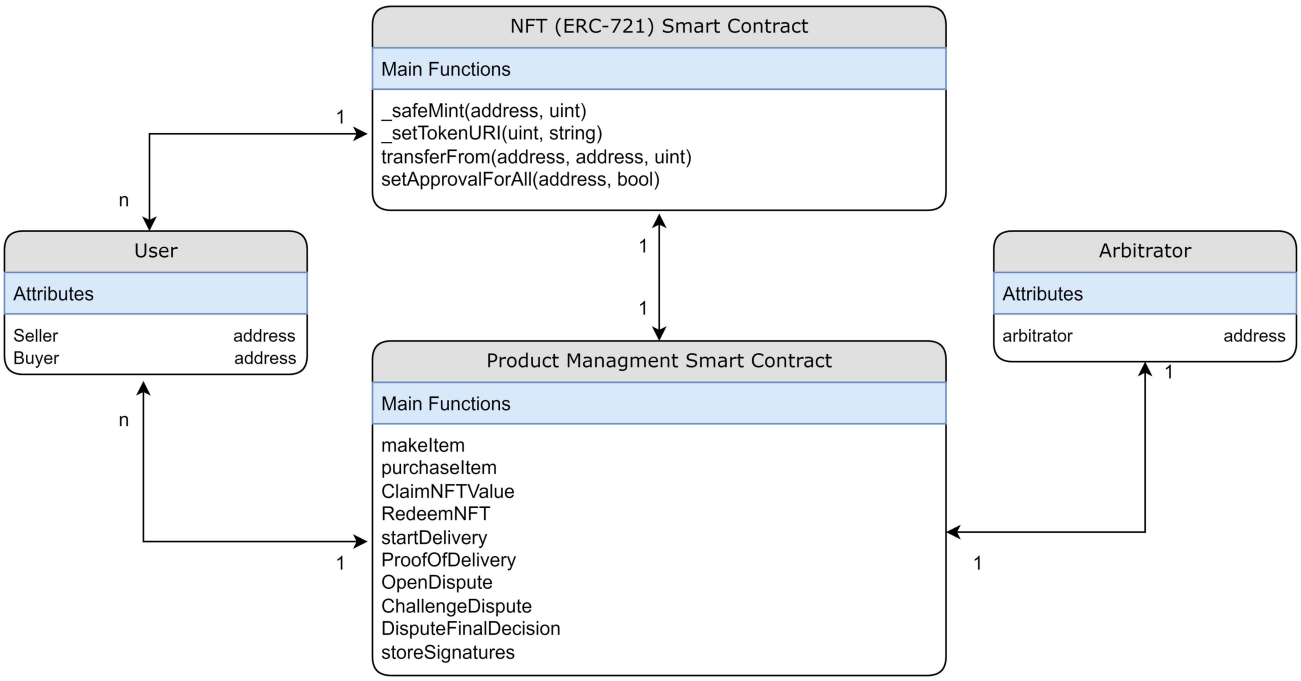


Fig. 8. Entity-relationship diagram depicting the interactions between the users and the smart contracts.

the approval process. Fourth, the product count within the Product Management smart contract is increased by one. Fifth, the security deposit and NFT are transferred to the Product Management smart contract. Sixth, the NFT value is set to the price determined by the seller. Finally, an event is emitted with the details of the listed NFT, which can be filtered and picked up by the front-end to display it for potential buyers. The emitted event includes details such as the NFT address, item ID, token ID, price, and healthcare product seller Ethereum address.

Algorithm 3 represents the NFT purchase process, and it consists of five main steps and three requirements. First, an interested buyer attempts to execute the NFT purchase function. Second, the NFT purchase function checks for three main requirements. It requires the buyer to have enough Ether to cover the price of the NFT, specify a valid NFT token ID, and choose an NFT that is unsold. Third, if all the requirements are fulfilled, the value of the NFT is transferred from the buyer to the Product Management smart contract. Fourth, the status of the NFT is changed from unsold to sold. Fifth, selling time is set to the time of purchase completion. Sixth, the ownership of the NFT is transferred to the buyer. Finally, an event is emitted declaring the details of the NFT purchase, which includes information such as the token ID, NFT address, NFT price, seller address, and the buyer address.

Algorithm 4 represents the NFT redemption process, and it consists of four main steps and four requirements. First, if the current owner of the NFT decides to redeem the NFT for its physical counterpart, the redemption function will check for three main requirements. The first requirement is that the redeemer of the NFT must be the current owner of the NFT. The second requirement is that the redemption function is executed during the valid redemption period. The third requirement is that the NFT has not already been redeemed. The fourth requirement is that the redeemer of the NFT has enough Ether to cover the security deposit. Second, if all the requirements are fulfilled, the NFT is transferred to the Product Management smart contract. Third, the value of the NFT is transferred to the Product Management smart contract as a security deposit. Fourth, the delivery start time is set to the time of the redemption function execution, as the seller will be required to deliver the physical counterpart of the NFT to the buyer. Finally, an event is emitted with the details of the redemption process, such as the redeemer address, seller address, and the NFT address and token ID.

Algorithm 5 represents the details of the delivery process of the redeemed NFT. The delivery process consists of three main phases, namely, stating the delivery, message signing and verification, and proof of delivery. First, the healthcare product seller will have to start the delivery process before the end of the delivery duration, which is a predetermined period of time during which the healthcare product seller has to deliver the product to its buyer; otherwise, the transaction will be reverted. Second, the NFT state is set to "EnRoute," which is an indicator that the product is out for delivery, and an event is emitted with further details, such as the seller address, buyer address, NFT ID, and delivery start time. Third, when the seller arrives at the location of the buyer who redeemed the NFT, a message confirming the reception of the product must be signed by the buyer. The message signing and verification process is illustrated in Algorithm 6, where the message is first hashed using the Keccak256 hashing algorithm [34]. Then, the message hash is encrypted by the private key of the buyer, which produces a unique hash. Next, the signature is sent to the seller, who interacts with the Product Management smart contract to store the message and signature of the buyer as proof of delivery.

Algorithm 1: Healthcare Product NFT Minting.

Input: *caller*, *nftOwnerAddress*, *metadata*, *tokenURI*, *ProductManagementSCAddess*, *Price*, *itemCount*, *tokenCount*

caller is the Ethereum address of the function caller
metadata is the uploaded NFT data to the IPFS
tokenURI is a variable representing the unique URI produced by IPFS for the uploaded *metadata* of the NFT
ProductManagementSCAddess is the smart contract address of the Product Management smart contract
Price is a variable representing the price of the NFT in Wei
tokenCount is a variable used to count the total number of NFTs listed in the NFT smart contract
initialization;
Upload *metadata* to the IPFS
Return *tokenURI*
if *tokenURI is valid* **then**
    Increase *tokenCount* by 1
    Mint an NFT with *tokenCount* as its token ID
    Set the *tokenURI* for the minted NFT
    Assign *Caller* as the owner of the NFT
    Emit an event declaring the address of the NFT owner and token ID
else
end
Revert contract state and show an error.
/* NFT minting is complete */

Algorithm 2: Healthcare Product NFT Listing.

Input: *caller*, *nftOwnerAddress*, *ProductManagementSCAddess*, *Price*, *itemCount*, *SecurityDeposit*, *msg.value*

caller is the Ethereum address of the function caller
ProductManagementSCAddess is the smart contract address of the Product Management smart contract
tokenID is a variable representing the unique ID of the NFT within the NFT smart contract
Price is a variable representing the price of the NFT in Wei
itemCount is a variable used to count the total number of NFTs listed in the Product Management smart contract
SecurityDeposit An amount that is transferred to the Product Management smart contract to ensure that the seller delivers the physical counterpart of the NFT when it is redeemed
value is the amount of Ether transferred when a function is called
initialization;
if *caller* == *nftOwnerAddress* **then**
    Approve the *ProductManagementSCAddess* to use the NFT identified by the *tokenID*
    Emit an event declaring the details of the approval process
else
end
Revert contract state and show an error.
/* NFT approval is complete */
if $Price > 0 \wedge value > SecurityDeposit$ **then**
    Increase *itemCount* by 1 and assign it to the NFT
    Set the NFT value to *Price*
    Transfer*value* to the *ProductManagementSCAddess*
    Transfer NFT to the *ProductManagementSCAddess*
    Emit an event declaring the details of the listed NFT
else
end
Revert contract state and show an error.
/* NFT listing is complete */

However, the function that stores the message and signature performs a validity check to ensure that the signature belongs to the buyer. The verification process is done by applying the same hashing algorithm (Keccak256) to the message; then, the signature is decrypted by the buyer's public key. If both hashes are equivalent, the signature is considered valid and, hence, is stored on-chain; otherwise, the transaction is reverted. Fourth, the healthcare product seller will be able to trigger the proof of delivery function in the Product Management smart contract, which updates the NFT state to "Delivered," and an event is emitted with the details of the delivery proof. Finally, the value of the NFT and the security deposit can be claimed by the seller.

The NFT buyer will have the option to open a dispute in case of an issue during the delivery process. Algorithm 7 illustrates the process of opening a dispute. First, the NFT buyer will have to upload evidence of the encountered issue during the delivery process on the IPFS. Second, the IPFS will produce a unique URI for the uploaded metadata. Finally, the NFT buyer will trigger a function in the Product Management smart contract that allows opening a dispute. This function requires that the NFT is not already successfully delivered, the caller of the function is the owner of the NFT, and the redemption time window is still opened. If all the conditions are satisfied, the IPFS hash of the metadata will be stored on-chain, the state of the NFT is updated to "Disputed," and an event is emitted with the details of the opened dispute, such as the seller Ethereum address, NFT address and token ID, NFT owner Ethereum address, and the IPFS hash.

The seller of the disputed NFT will have the option to challenge the dispute. Algorithm 8 illustrates the details of challenging a dispute. First, the NFT seller will upload the metadata of the evidence supporting the dispute challenge to the IPFS, and the IPFS will return a unique URI for the uploaded metadata. Second, challenging a dispute requires the caller of the corresponding function to be the seller of the NFT, the NFT state is disputed, and the time of challenging the dispute is within the redemption period. Finally, if all the conditions are satisfied, the IPFS hash is stored on-chain, the NFT state is updated to

Algorithm 3: Healthcare Product NFT Purchase.

Input: *caller*, *nftID*, *value*, *ProductManagementSCAddess*, *currenttime*

caller is the Ethereum address of the function caller
value is the amount of Ether transferred when a function is called
Price is the price of the NFT
nftCount is a variable used to count the total number of NFTs listed in the Product Management smart contract
nftID is the NFT ID within the Product Management smart contract
nft.sold is a boolean variable that shows if the NFT has already been sold or not
SellingTime is a variable that stores the time at which the NFT is sold
initialization;
if $value == Price \wedge (nftID \geq 0 \wedge nftID \leq nftCount) \wedge nft.sold == false$ **then**
    Transfer NFT *Price* to the *ProductManagementSCAddess*
    Set the *nft.sold* boolean to true
    Set the *SellingTime* to *currenttime*
    Transfer ownership of the NFT to *caller*
    Emit an event declaring the details of the NFT purchase
else
    Revert contract state and show an error.
end
/* NFT purchase is complete */

Algorithm 4: Healthcare Product NFT Redemption.

Input: *caller*, *nftID*, *value*, *ProductManagementSCAddess*, *currenttime*

caller is the Ethereum address of the function caller
value is the amount of Ether transferred when a function is called
currenttime is the time at which a function is executed
Price is the price of the NFT
nftOwnerAddress is the Ethereum address of the current owner of the NFT
ProductManagementSCAddess is the smart contract address of the Product Management smart contract
SellingTime is a variable that stores the time at which the NFT is sold
RedemptionPeriod is a period during which the NFT can be redeemed
initialization;
if $caller == nftOwnerAddress \wedge currenttime <= SellingTime + RedemptionPeriod \wedge NFTnotalreadyredeemed \wedge value >= SecurityDeposit$ **then**
    Set the *DeliveryStartTime* to the *currenttime*
    Transfer the NFT to the *ProductManagementSCAddess*
    Transfer the NFT *Price* to the *ProductManagementSCAddess* as a security deposit
    Emit an event declaring the details of the NFT redemption
else
    Revert contract state and show an error.
end
/* NFT Redemption process is initiated */

"Challenged," and an event is emitted with the details of the dispute challenge, such as the seller Ethereum address, NFT address and token ID, NFT owner Ethereum address, and the IPFS hash.

Algorithm 9 describes how a dispute is settled. The arbitrator will inspect the provided evidence for the opened dispute, as well as the challenge of the dispute. There are three potential outcomes for a dispute. In the first case, if the provided evidence proves that the buyer rejected receiving the product for no valid reason, then the arbitrator will trigger the dispute final decision function and declare the winner of the dispute. In this particular case, twice the value of the NFT is transferred to the seller, the NFT is transferred back to its seller, the attributes of the NFT are reset, and an event is emitted with the details of the final decision. In the second case, if the provided evidence proves that the delivered product is damaged or expired, then the arbitrator will declare the buyer as the winner and trigger the dispute final decision function accordingly. Twice the value of the NFT is transferred to the buyer, the NFT is burned as it is no longer usable, and an event is emitted with the details of the final decision. Finally, in the third case, if the provided evidence proves that the seller did not deliver the product at all, then the arbitrator will choose the buyer as the winner and trigger the dispute final decision function accordingly. Twice the value of the NFT is transferred to the buyer, the NFT is transferred back to the buyer, the attributes of the NFT are reset, and an event is emitted with the details of the final decision.

Fig. 9. Front-end view of the healthcare product NFT creation and listing.

C. Front-End View

The last stage of implementing the proposed solution is developing the front-end application, which allows the users to interact with the smart contracts and the blockchain in an easy and seamless way.

Algorithm 5: Healthcare Product Delivery.

Input: *caller*, *nftID*, *healthcareproductbuyer*, *healthcareproductseller*, *nftState*, *DeliveryStartTime*, *DeliveryDuration*, *Message*, *Signature*, *currenttime*

caller is the Ethereum address of the function caller
currenttime is the time at which a function is executed
Message is a text message that the NFT buyer has to sign to confirm receiving its physical counterpart
Signature is the generated outcome from signing a message with a private key
initialization;
if *caller* == *healthcareproductseller* ∧ *currenttime* ≤ *(DeliveryStartTime* + *DeliveryDuration)* **then**
    Update *nftState* to *EnRoute*
    Emit an event declaring the details of the delivery start
else
    Revert contract state and show an error.
end
/* Healthcare product delivery has started */
Request *healthcareproductbuyer* to sign a *message* and verify *signature* as illustrated in 6
if *caller* == *healthcareproductseller* ∧ *currenttime* ≤ *(DeliveryStartTime* + *DeliveryDuration)* ∧ *(hash*1 == *hash*2*)* **then**
    Update *nftState* to delivered
    Emit an event declaring the details of delivery completion
else
    Revert contract state and show an error.
end
/* Healthcare product delivery is complete */

Algorithm 6: Message Signing and Verification.

Input: *caller*, *healthcareproductseller*, *healthcareproductbuyer*, *Message*, *Signature*

caller is the Ethereum address of the function caller
Message is a text message that the NFT buyer has to sign to confirm receiving its physical counterpart
Signature is the generated outcome from signing a message with a private key
initialization;
Apply keccak256(*Message*) to produce *hash*1
Encrypt *hash*1 with the *healthcareproductbuyer* private key to produce a unique *Signature*
Send the *Signature* and *message* to the *healthcareproductseller*
if *caller* == *healthcareproductseller* **then**
    Apply keccak256(*Message*) to produce *hash*1
    Decrypt the *Signature* using the *healthcareproductbuyer* public key to produce *hash*2
    if *hash*1 == *hash*2 **then**
        Set the *signature* of *healthcareproductbuyer* as valid **Emit** an event declaring the details of the stored signature
    else
        Revert contract state and show an error.
    end
else
    Revert contract state and show an error.
end
/* Message signing and verification is complete */

Fig. 9 illustrates the front-end page for tokenizing products and listing them. The user needs to upload an image of the product, name it, describe it, and specify the price in Ether. The front-end will automatically parse these inputs into their respective types and utilize them to mint, approve, and list NFTs. Finally, Metamask is used as a signer for the transactions as well as a provider to relay transactions to the blockchain.

Fig. 10 shows how the products that are converted into NFTs are presented on the home page of the DApp. Each listed NFT will be displayed based on its metadata that is uploaded to the IPFS, where its image, name, description, and price are displayed on the front-end to make it easy for the user to view NFTs. Finally, the user can purchase NFTs by clicking on the buy button and then signing the transaction using Metamask to confirm it.

Fig. 11 shows the "My Products" page via the front-end DApp. This page allows the NFT seller to view all the currently listed NFTs, as well as the previously sold NFTs. This page makes

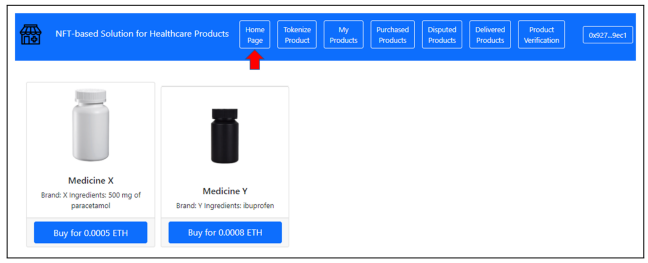


Fig. 10. Front-end view of the listed healthcare products.

it easy for the user to track the status of the NFTs without manually fetching information from the blockchain, which can be a complicated process. Also, any actively listed NFT that has not been sold yet can be taken off the list using the smart contract for Product Management.

Fig. 12 illustrates how a purchased NFT appears on the "Purchased Products" page. This page makes it easy for the NFT owner to view the contents of the NFT, exchange the NFT for its physical counterpart, or start a dispute if there is a problem with the product.

Fig. 13 shows the delivery page on the DApp. Once the NFT buyer triggers the redemption function, the NFT will appear

Algorithm 7: Opening a Dispute.

Input: *caller*, *healthcareproductbuyer*, *nftState*, *RedemptionPeriod*, *SellingTime*, *currenttime*

caller is the Ethereum address of the function caller
RedemptionPeriod is the allowed duration to redeem the NFT
currenttime is the time at which a function is executed
initialization;
Upload *metadata* to the IPFS
Return *IPFShash*
if *caller* == *healthcareproductbuyer* ∧ *nftState* ≠ *Delivered* ∧ *(currenttime* ≤ *SellingTime* + *RedemptionPeriod)* **then**
    Store the *IPFShash* of the metadata of the dispute on-chain
    Update *nftState* to Disputed
    Emit an event declaring the details of the opened dispute
else
    Revert contract state and show an error.
end
/* The healthcare product NFT dispute is openned */

Algorithm 8: Challenging a Dispute.

Input: *caller*, *healthcareproductseller*, *nftState*, *RedemptionPeriod*, *currenttime*

caller is the Ethereum address of the function caller
RedemptionPeriod is the allowed duration to redeem the NFT
currenttime is the time at which a function is executed
initialization;
Upload *metadata* to the IPFS
Return *IPFShash*
if *caller* == *healthcareproductseller* ∧ *nftState* == *Disputed* ∧ *currenttime* ≤ *(SellingTime* + *RedemptionDuration)* **then**
    Store the *IPFShash* of the metadata of the dispute challenge on-chain
    Update *nftState* to challenged
    Emit an event declaring the details of the dispute challenge
else
    Revert contract state and show an error.
end
/* The healthcare product NFT dispute is challenged */

on the delivery page of the seller, who is required to deliver it within the delivery time window, which is opened as soon as the buyer redeems the NFT. On this page, the NFT seller can start the delivery process, store the confirmation message and the

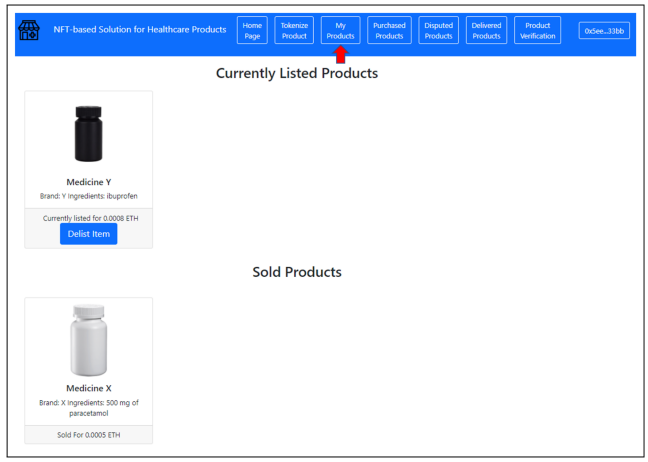


Fig. 11. Front-end view for the currently listed and sold NFTs.

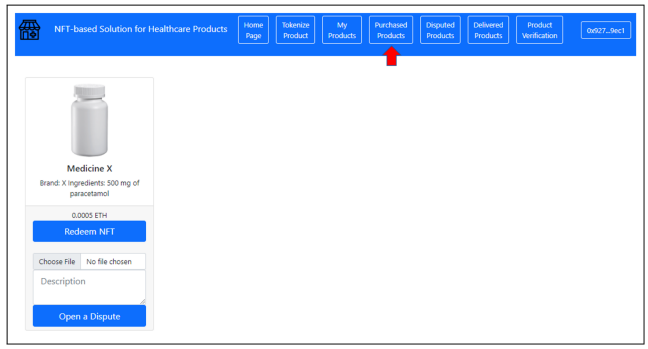


Fig. 12. Front-end DApp view of the purchased NFTs.

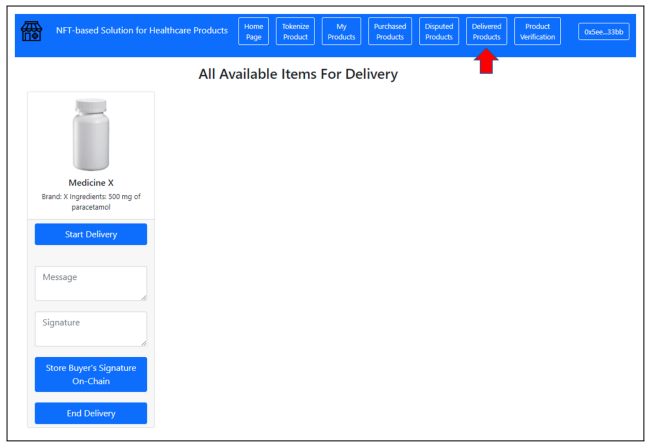


Fig. 13. Front-end view of the delivery page.

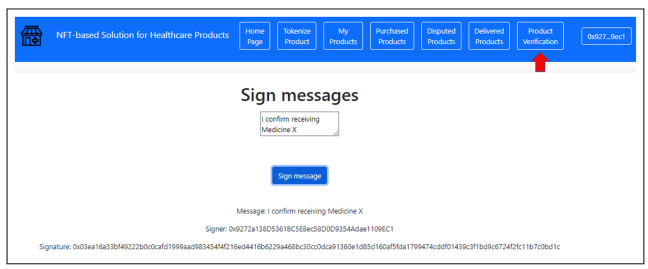


Fig. 14. Front-end view of the message signing page.

Algorithm 9: Dispute Settlement.

Input: *caller*, *nftID*, *Price*, *Arbitrator*, *winner*

caller is the Ethereum address of the function caller
Arbitrator is the Ethereum address of the entity responsible for solving the dispute
winner is the Ethereum address of the winner from the dispute
initialization;
if *caller* == *Arbitrator* ∧ (*nftState* == *Disputed* || *nftState* == *Challenged*) **then**
  if *Decision* == *BuyerRejectedReceivingtheNFT* **then**
    Set the *healthcareproductseller* as the *winner* **Transfer** twice the NFT *Price* to the *healthcareproductseller*
    Transfer the healthcare product NFT to the seller
    Reset the attributes of the NFT to their initial values
    Emit an event declaring the details of the final decision of the dispute
  else
  end
  if *Decision* == *BuyerReceivedaDamaged/ExpiredProduct* **then**
    Set the *healthcareproductbuyer* as the *winner*
    Transfer twice the NFT *Price* to the *healthcareproductbuyer*
    Burn the healthcare product NFT
    Emit an event declaring the details of the final decision for the dispute
  else
  end
  if *Decision* == *SellerFailedtoDelivertheProductonTime* **then**
    Set the *healthcareproductbuyer* as the *winner*
    Transfer twice the NFT *Price* to the *healthcareproductseller*
    Transfer the healthcare product NFT back to the seller
    Reset the attributes of the NFT to their initial values
    Emit an event declaring the details of the final decision for the dispute
  else
  end
else
  Revert contract state and show an error.
end
/* The healthcare product NFT dispute is settled */

signature that are obtained from the buyer as shown in Fig. 14, and end the delivery process if all the conditions are satisfied.

Fig. 15 shows the "Disputed Products" page, which provides the arbitrator with the details of any active disputes. Once the arbitrator makes the final decision, the reasoning behind the decision is inserted, and the winner is picked accordingly. In addition, a MultiSig wallet, which requires multiple signatures to sign transactions, can be used for the decision-making process to ensure that no single entity has full control over such decisions.

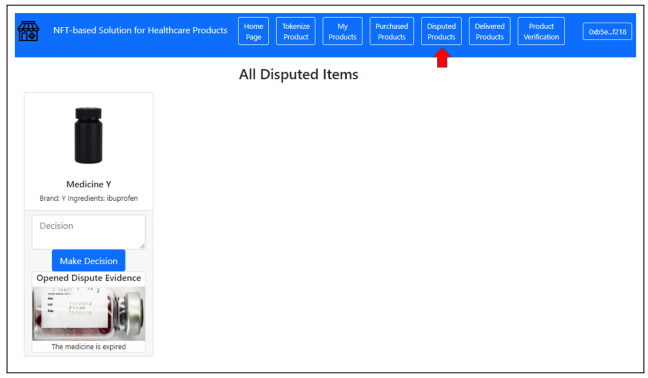


Fig. 15. Front-end view of the disputes page.

TABLE II
ETHEREUM ADDRESS OF EACH PARTICIPANT AND SMART CONTRACT IN THE PROPOSED SOLUTION

Participant/Smart contract	Ethereum Address
Healthcare Product Seller	0x5eE248A1C87aAc059fC4038B594b8122426033Bb
Healthcare Product Buyer	0x9272a138D5361BC5E8ec58D0D9354Adae1109EC1
Arbitrator	0xb5Ee4B7b2366425c71ABd97096079550CF4dF218
NFT Smart Contract	0x8a1487fA507084Cae331fEAFCF375147781dAe8b
Product Management smart contract	0x81FAeaAbBE36540f77640F04c683cf6202d4cc76

TABLE III
HASH ADDRESS OF THE EXECUTED FUNCTIONS IN THE SMART CONTRACTS

Function/Interaction	Transaction Hash
Mint NFT	0x4c96d537a8e83ce59b0bdb0b133447158b5d9f6823556d16463b714c472c3c43
List NFT	0xdca80b170d6a2ac3d917cbdb2106d946f09f1b7dbea953f8a2a58a36da5d9994
Purchase NFT	0xcb89482c865f4702ec73ad8b42b9816a76f7e3c18ca016b59cf66dca9687a5b8
Redeem NFT	0x0713857551512a829d920a18cb8d675e4b5753dd6c9c1ed00f5426f3e299168f
Start Delivery	0x23b53238feb80ecfafe9b6492a669b52468d4043614273e4b991392afac8260b
Store Signature	0xc6cdd54d96cdd3e005b92caeabe3c355999f214a65c2da2c53e94108f25c41bd
Proof of Delivery	0xb44a7bd67abf1a50def84f3537f5419e5a0e9414ee418f65cd7404a0ae57f6c4
Open Dispute	0x25c249294462228e8233899a68320be608ff691e98f32207de3cca9897f66e67
Dispute Settlement	0x2239d782b2315943cb7d0ac105c8baebe9a41ba1de6aaf7df3e34889f411e9b7

V. TESTING AND VALIDATION

In this section, the functionality of the smart contracts is tested and validated. The smart contracts are written in Solidity Language, compiled using Hardhat environment, and finally deployed and tested on an Ethereum Testnet called Rinkeby. Moreover, the front-end DApp is used to interact with the smart contracts and blockchain. Table II provides a list of all participants and smart contracts and their corresponding Ethereum addresses. Table III provides all the produced transaction hashes, which can be viewed on a block explorer, such as Etherscan.

A. Minting and Listing an NFT

The first process that is tested is the minting process. Fig. 16 shows the details of the emitted event after successfully minting a new NFT. The address that initiated the minting process and the token ID of the minted NFT are announced. The second process that is tested is the listing process in the Product Management

Fig. 16. Resulting events after minting a new NFT.

Fig. 17. Resulting events after listing an NFT in the Product Management smart contract.

smart contract. This process involves three transactions, namely, approval, list NFT, and transfer NFT. Fig. 17 provides the details of the emitted events after each transaction. First, the "ApprovalForAll" event is needed to announce that the Product Management smart contract is allowed to use the minted NFT on behalf of the participant that minted it. Second, the "Transfer" event is emitted after the transferring the NFT from its owner to the Product Management smart contract, and it is an internal transaction that is executed within the listing function. Finally, the "Offered" event provides details of the listed NFT, such as item ID within the Product Management smart contract, the NFT address, token ID within the NFT smart contract, NFT price, and the address of the NFT seller. Further details of the minting and listing processes can be inspected via their transactions hashes in Table III.

B. Purchasing and Redeeming an NFT

Once an NFT is listed successfully, any buyer that is interested can initiate the "Purchase" function. If the purchase transaction goes through, the buyer can redeem the purchased NFT for its physical within a given redemption time window. Fig. 18 provides the details of the events that are emitted after a successful purchase and redemption. The event "Bought" provides information for the purchased NFT, such as the item ID of the NFT within the Product Management smart contract, the purchased NFT address, token ID of the NFT within the NFT smart contract, the price of the NFT, and the addresses of both the buyer and seller. The event "RedemptionRequest" emits information for the redemption request, such as the item

Fig. 18. Resulting events after purchasing and redeeming an NFT.

Fig. 19. Emitted events after successfully delivering a redeemed NFT.

ID of the NFT within the Product Management smart contract, redeemed NFT address, token ID of the NFT within the NFT smart contract, the address of the buyer, and the redemption time. Further details of the purchase and redemption processes can be viewed by the provided transaction hashes in Table III.

C. Successfully Delivering a Redeemed NFT

If an NFT buyer decides to redeem the NFT, the seller will be responsible for delivering its physical counterpart within a predetermined time window. Fig. 19 provides details of the emitted events after successfully delivering a healthcare product. First, the NFT seller will initiate the delivery process by running the "startDelivery" function in the Product Management smart contract, which emits an event that declares the start of the delivery process and provides information, such as the seller Ethereum address, NFT address, NFT token ID, buyer Ethereum address, and the delivery start time. Second, when the seller

Fig. 20. Resulting events after opening and settling a dispute.

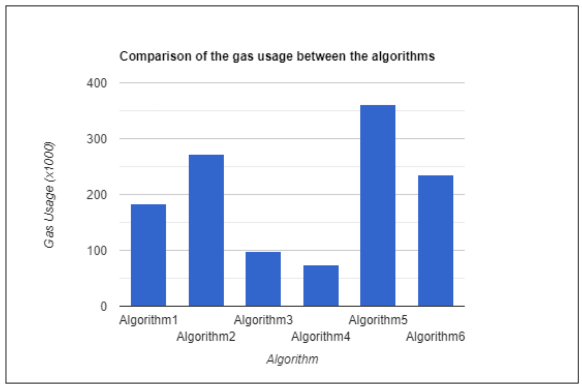


Fig. 21. Comparison of the gas usage between the algorithms.

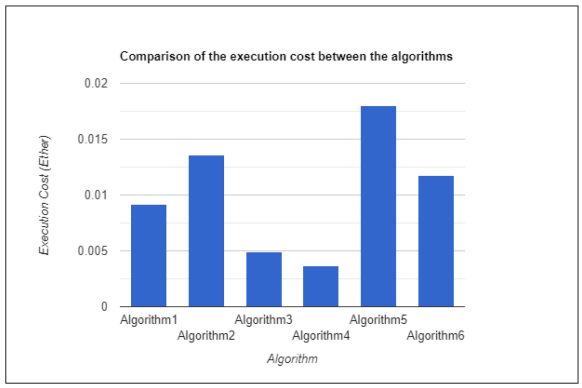


Fig. 22. Comparison of the execution cost between the algorithms.

arrives to the location of the buyer, the latter will be requested to sign a message confirming the reception of the product with a private key. This signature will be obtained off-chain, and it can be used on-chain (or off-chain) to verify that it belongs to the buyer. Next, the seller will store the message and signature of the buyer on-chain. The event called "SingatureStorage" emits the details of this process, such as the verified Ethereum address, buyer's Ethereum address, NFT address, NFT token ID, signed message, and signature. Finally, the seller will be able to execute the "ProofOfDelivery" function, which requires the stored signature to be valid. This function emits an event called "DeliveryProof," which provides details, such as the seller and buyer Ethereum addresses, NFT address, NFT token ID, and delivery time. Further details of the delivery process can be viewed by the provided transaction hashes in Table III.

D. Settling an Opened Dispute

For this particular test, it is assumed that the buyer receives an expired item and opens a dispute accordingly. Fig. 20 provides details of the emitted events after successfully settling the dispute. First, the buyer will have to upload the metadata of the expired item to the IPFS, which returns a unique URI. Then, the buyer trigger the "OpenDispute" function in the Product Management smart contract, which requires the IPFS hash and item ID as inputs. Once executed successfully, an event called "DisputeOpenned" is emitted, which provides details, such as the seller and buyer Ethereum addresses, the NFT address, token ID, and the IPFS hash. Second, the arbitrator will be able to view the details of the opened dispute by the provided IPFS hash and make a decision accordingly. Finally, the arbitrator's final decision would be that the buyer indeed received an expired item and an event is emitted with the details of the final decision, such as the winner address, NFT address, NFT token ID, and the final decision. Further details of the dispute settlement process can be viewed by the provided transaction hashes in Table III.

VI. Discussion

In this section, we analyze the performance of our solution, as well as different security aspects for the developed smart contracts. Moreover, we compare our solution with the existing solutions. Furthermore, we show how our solution can be generalized and extended to other applications. Finally, we describe the main advantages, challenges, and limitations of our proposed solution.

A. Performance Analysis

The performance of our proposed solution is analyzed in terms of the amount of gas used to execute functions and the cost that is incurred. Table IV summarizes the involved functions in each algorithm, the gas used to successfully execute those functions, and their cost in Ether if they are executed on Ethereum's Mainnet. Fig. 21 better compares the gas usage associated with each algorithm. It can be clearly seen that Algorithm 5 is the most costly algorithm, and the reason is that it requires the storage of the recipient's signature to confirm the delivery process. Overall, the gas usage of executing the algorithms is considered high, and using a public blockchain like Ethereum's Mainnet will require spending Ether, which can be very costly and volatile, hence making the experience for the users unpleasant and inconsistent. Moreover, Fig. 22 compares the cost of executing each algorithm in Ether. The Minting and Listing algorithms are considered prerequisites for the remaining algorithms, and since both of them are relatively costly, it would not be feasible to implement our solution on a public blockchain. This analysis shows that our solution is more suitable for a private blockchain than a public blockchain.

TABLE IV
QUANTITATIVE ANALYSIS OF THE PERFORMANCE OF THE IMPLEMENTED ALGORITHMS

Algorithm	Involved Functions	Gas Usage	Cost to Execute (Ether)	
			Public Blockchain	Private Blockchain
Healthcare Product NFT Minting	Minting the NFT	183,260	0.009163	None
Healthcare Product NFT Listing	NFT Approval NFT Transfer NFT Listing	271,601	0.01358	None
Healthcare Product NFT Purchase	NFT Transfer NFT Purchase	97,596	0.004879	None
Healthcare Product NFT Redemption	NFT Transfer NFT Redemption	73,530	0.003676	None
Healthcare Product Delivery	Start Delivery Signature Storage End Delivery	360,913	0.01804	None
Dispute Settlement	Open Dispute Challenge Dispute Dispute Settlement	234,793	0.01173	None

Fig. 23. Summary of the detected vulnerabilities by MythX.

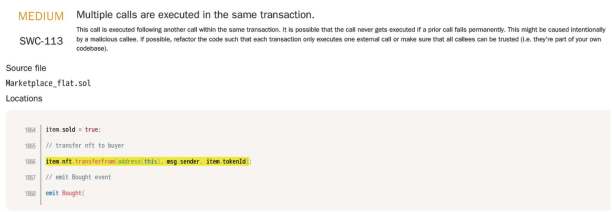


Fig. 24. Moderate vulnerability found in the Product Management smart contract.

B. Security Analysis

We conduct security analysis to check if our smart contracts are susceptible to the most common vulnerabilities and exploits of EVM-based smart contracts .

Smart contracts are an integral part of any blockchain-based system, and if they are not coded properly, they become susceptible to various vulnerabilities that can lead to the failure of the whole system. MythX security tool is developed by ConsenSys company and used to conduct security analysis for our smart contracts. MythX allows developers to submit their code to an API that sends their code to multiple microservices in parallel that detect vulnerabilities in the smart contract, and their severity ranges from low to high [35].

MythX covers various vulnerabilities, such as integer overflow and underflow, unauthorized controls, ERC standards violations, and poor practices in solidity coding. The full list of the covered vulnerabilities is available in [36]. Fig. 23 shows the summary of the found vulnerabilities by MythX in our smart contracts. The only vulnerability that is found is shown in Fig. 24; however, in this particular case, this vulnerability cannot be exploited because the function requirements ensure that both calls within the function cannot be manipulated. The full reports are made available on GitHub.[3]

C. Comparison With the Existing Solutions

Table V compares our solution with existing blockchain-based healthcare solutions and other NFT-enabled solutions for different applications. Sadri et al. [17] propose a blockchain-based solution to solve the issue of poor visibility in the blood donation supply chain. The authors in [18], [19], [20], [22], and [24] provide a solution for healthcare data management, where users can access their records at any time without needing a central authority. Asad et al. [21] provide a solution for healthcare product authentication where users are able to verify the origin of the products. Finally, Ramyasri and Hussain [23] and Younis et al. [25] provide a solution for healthcare record access control where users can decide who accesses their records. Our solution represents healthcare products in the form of NFTs that allows users to trade them easily and monetize them, and this is the main difference between our solution and all the aforementioned solutions. Moreover, our solution provides a better solution for data sustainability, management, and organization. Our proposed solution is different from all the aforementioned solutions because it tokenizes healthcare products in the form of an NFT, which provides a standard method for organizing the details of data since it uses the ERC-721 standard. Moreover, the tokenization of healthcare products allows users to trade and monetize their products easily. In addition, NFTs make tracking products much easier because the holder of the NFT can be easily identified through the NFT smart contract. Moreover, the structure of the NFT smart contract along with the emitted events provides much more meaning full data to analyze and interpret since they are structured in a standard way. Finally, our solution allows its users to redeem NFTs for their physical counterparts, and provides the seller with a mechanism that verifies the delivery process, and in the case of a dispute, there is a dispute settlement mechanism as well.

[3]https://github.com/DrugTraceability/HealthcareNFTs

TABLE V
COMPARISON BETWEEN OUR SOLUTION AND THE EXISTING SOLUTIONS

	Application Type	Blockchain Platform	Mode of Operation	Consensus Algorithm	Off-Chain Storage	Traceability	Items Tokenization (NFTs)	Items Tracking	Decentralized Application (DApp)
Our Solution	Healthcare Products Supply Chain	Ethereum	Private Permissioned	PoA	IPFS	Yes	Yes	Yes	Yes
[17]	Blood Donation Supply Chain	Ethereum	Public Permissionless	PoW	None	Yes	No	No	No
[18]	Drug Prescription Management	Hyperledger Fabric	Private Permissioned	PBFT	IPFS	Yes	No	No	No
[19]	Mobile Healthcare Data Management	Ethereum	Consortium	NA	IPFS	Yes	No	No	No
[20]	Secured Healthcare Data Sharing	Ethereum	Private Permissioned	PoA	None	Yes	No	No	No
[21]	Healthcare Products Authentication	Ethereum	Private Permissioned	PoW	IPFS	Yes	No	No	No
[22]	Fragmented Medical Records	Hyperledger Fabric	Private Permissioned	PBFT	None	Yes	No	No	No
[23]	Healthcare Data Access Control	Ethereum	Public Permissionless	PoW	None	Yes	No	No	No
[24]	EMRs Management System	Ethereum	Public Permissionless	PoW	None	Yes	No	No	No
[25]	Medical Data Access Control	Ethereum	Public Permissionless	PoW	None	Yes	No	No	No

D. Generalization

Our proposed solution provides a management solution for the ownership of healthcare products. Moreover, it allows tracing the delivery process thoroughly. However, it can also be generalized and extended to other applications within the healthcare industry, and also in other industries as well. This can be achieved by customizing the nomenclature and requirements within the smart contracts to fit the needs of the new application.

In terms of infrastructure setup, as long as the used blockchain is EVM based and supports the use of NFTs and the ERC-721 standard, similar logic of our smart contracts can be implemented . Furthermore, if an application requires monitoring the condition of the items being listed, then it would be recommended to add Internet-of-Things (IoT) devices to record any abnormality that occurs on the blockchain [37], and this addition modification will just require the user to modify the way NFT metadata is handled because it has to be modified whenever an abnormality occurs; however, the previous state of the metadata should be stored and maintained on the blockchain to ensure data provenance.

Moreover, the NFT trading algorithm can be upgraded to support auctions where a seller might be interested in listing a unique product without setting a fixed price for it. This can be achieved by storing the bids of the interested buyers within a certain time window, and the buyer gets to accept a certain bid or just reject them all.

In addition, our solution can be extended to any other application that requires certificate transparency. This can be easily achieved since the ERC-721 NFTs are basically a tokenized version of a physical product, and the blockchain inherently records time-stamped data of all the transactions that are related to NFTs. Therefore, the use of NFTs can act as a digital certificate that is tamper-proof, auditable, and transparent.

Finally, the high-level system architecture depicted in Fig. 2 will pretty much remain the same unless the new application does not require off-chain storage, where, in this case, the metadata will be directly stored on the blockchain, and it cannot be modified.

E. Advantages, Challenges, and Limitations

Although the use of blockchain technology and NFTs brings advantages for the healthcare supply chain, they still have not reached a high level of adoption and acceptance worldwide, as they still face some challenges and limitations. The key advantages, challenges, and limitations are described as follows.

1) *Data ownership:* The user who mints the NFT for healthcare product is assured of retaining the ownership of that product because it is stored permanently on the blockchain.
2) *Data uniqueness and counterfeit prevention:* Once an NFT is minted for a healthcare product, a unique address is generated, which cannot be replicated or replaced. Therefore, each product is guaranteed to be unique.
3) *Tokenization and monetization:* Any healthcare product can be tokenized and converted into an NFT, which can be easily traded among participants by using the Product Management smart contract that facilitates the trading process. This provides participants with a secure and trustworthy ecosystem where they can trade their healthcare products without the need for a third party.
4) *Data provenance:* All the interactions with the developed smart contracts are permanently stored on the blockchain. Therefore, data provenance is ensured as the history of each participant or NFT can be traced.
5) *Accountability:* All the interactions with the smart contracts are timestamped and permanently stored on the blockchain. Therefore, in the case of a dispute, the entity that is at fault can be easily identified.

1) Challenges:

1) *Interoperability with legacy systems:* The healthcare industry is a well-established industry that already leverages so many technologies, and it is very challenging to integrate blockchain technology and NFTs into the already existing solutions because it would require major changes in the infrastructure of the industry. Therefore, numerous feasibility studies must be well articulated and conducted to decide the applicability of these technologies to the healthcare industry. Moreover, the blockchain technology itself has issues with interoperability with other blockchains. As a result, the choice of blockchain type is also critical [38], [39].
2) *Smart contract security:* The design of smart contracts is one of the most challenging parts of implementing blockchain-based solutions. Since the blockchain is decentralized and permissionless, anyone can technically develop smart contract codes and deploy them on the blockchain. Although this gives a low entry point for everyone, it results in the introduction of many poorly coded smart contracts. Furthermore, there is currently no standard way of coding smart contracts to follow except for a few cases, such as the ERC-721 standard that is used for NFTs. However, trading NFTs through smart contracts will expose them to the vulnerabilities of unstandardized and unregulated smart contracts. Overall, smart contract vulnerabilities and security risks remain

one of the main challenges for blockchain technology and NFTs.

3) *Storage limitations:* In NFTs, the storage of data requires a large-sized storage system that is capable of accommodating it. Numerous solutions are currently available for storage, such as IPFS and Swarm. However, the nodes maintaining those decentralized storage systems must remain incentivized to maintain the stored files permanently [40].
4) *Lack of blockchain and NFTs experts:* Blockchain technology and NFTs have attracted a lot of attention. However, they still lack technical experts. Therefore, this challenge will make scaling the use of blockchain technology and NFTs at a large scale very difficult.
5) *Lack of regulation framework:* The decentralization of blockchain technology makes imposing regulations on the implemented applications very difficult [41]. Many large institutions are interested in the technology of blockchain and NFTs. However, the lack of regulations makes them hesitant when it comes to investing. Similarly, many individual investors refuse to invest in blockchain technology and NFTs because of the lack of regulations [42]. This challenge has been one of the most contradictory challenges because blockchain technology was first introduced to allow decentralization and eliminate the need for a third party.
6) *High infrastructure cost:* The healthcare industry already has an established infrastructure for the management of healthcare products, and although this infrastructure has numerous limitations and drawbacks, shifting to a new one can be quite costly as it requires the implementation of a completely new infrastructure. Therefore, a shift towards a new infrastructure should be done gradually [43]. In addition, healthcare products should be categorized and prioritized based on their suitability for the new infrastructure, and then, the performance of our proposed solution for each category should be assessed and evaluated to determine if it improves the status quo. Once all healthcare product categories are tested and evaluated, the overall efficiency and efficacy of our proposed solution is determined, and a decision is made to decide if the shift to the new infrastructure is justified or not.

2) Limitations:

1) *Scalability:* Scalability in blockchain technology refers to the ability to support high transactional throughput even when the network expands so that it can handle an acceleration in adoption. Improving the scalability of a blockchain is not impossible. However, it comes as a trade-off with security and decentralization, and this is referred to as the blockchain trilemma, which means that only two of these properties can the blockchain have [44]. There are currently many efforts and attempts to solve or at least improve the scalability of the Ethereum blockchain, such as on-chain scaling using Sharding and off-chain scaling using layer 2 solutions, such as Rollups, Sidechains, and Plasma [45].
2) *Cost:* Blockchain-based solutions developers can either deploy their smart contracts on a public or private Ethereum blockchain. In the former, users must spend gas to make a transaction, and spending gas costs Ether. This creates an issue for the users because the price of gas is volatile depending on the blockchain congestion, and the price of Ether itself is also not stable. However, in the latter, the blockchain developer can eliminate gas costs to provide users with a consistent experience when making transactions. Moreover, the blockchain developer can opt for another type of consensus algorithm, such as proof-of-stake or PoA to reduce gas costs.

VII. Conclusion

In this article, we proposed an NFT-based solution for healthcare products to sustain data ownership and data provenance in a manner that is decentralized, transparent, secure, reliable, auditable, and trustworthy. We used a decentralized storage system named IPFS to host the metadata of healthcare products and append them to an NFT. We developed smart contracts that allow NFT minting, transferring, and facilitate the trading process in a Product Management smart contract. Moreover, the developed smart contracts allow the tracing of NFTs from the moment they are minted until they are successfully delivered or disputed. Our proposed solution enables users to tokenize their healthcare products, which allows them to trade and trace them easily. Moreover, our solution provides data analytics tools with a trusted, secure, reliable, and transparent database, which allows them to conduct their analyses without the need for a third party and without worrying about the authenticity and completeness of data. Also, the NFT-based solution provides manufacturers with an easy way to offer their healthcare products and reach a large audience. We presented nine algorithms along with their implementation and testing details. We performed security analysis to show that our smart contracts' logic is implemented properly and that the codes do not suffer from any severe vulnerabilities. We showed the novelty and unique features of our solution by comparing it with existing solutions. We illustrated how our solution can be generalized and extended to fit the needs of other applications. Finally, we identified the challenges and limitations that might hinder the adoption process of our proposed solution.

References

[1] *Snapshots: Comparing Projected Growth in Health Care Expenditures and the Economy*. The Henry J. Kaiser Family Foundation, San Francisco, CA, USA. Accessed: Apr. 22, 2022. [Online]. Available: https://tinyurl.com/2daw9py3

[2] *Monitoring the Building Blocks of Health Systems: A Handbook of Indicators and Their Measurement Strategies*. World Health Organization, Geneva, Switzerland, 2010.

[3] T. Cueni, *The Pharmaceutical Industry and Global Health-Facts and Figures*. Geneva, Switzerland: International Federation of Pharmaceutical, 2021. Accessed: Apr. 22, 2022. [Online]. Available: https://tinyurl.com/yzfvmzb8

[4] *Everybodyś Business–Strengthening Health Systems to Improve Health Outcomes: WHOś Framework for Action*, World Health Organization, Geneva, Switzerland, 2007. Accessed: Apr. 22, 2022. [Online]. Available: https://apps.who.int/iris/handle/10665/43918

[5] *Mitigating and Preventing Medical Device Shortages and Prioritizing Public Health*, Silver Spring, MD, USA: United States Food and Drug Administration. Accessed: Apr. 22, 2022. [Online]. Available: https://tinyurl.com/2p8tx27t

[6] *New Survey of Nurses Provides Frontline Proof of Widespread Employer, Government Disregard for Nurse and Patient Safety, Mainly Through Lack of Optimal PPE*, Northwest Nazarene Univ., Nampa, ID, USA, 2020. [Online]. Available: https://tinyurl.com/3ft787j2

[7] K. Pilz, N. Shakfeh, M. Perry, and P. Yadav, *Data Science to Forecast Demand for Supplies at Health Clinics*. Washington, DC, USA: Center Global Develop., 2022.

[8] N. Sekhri, R. Levine, and J. A. Pickett, *Risky Business Saving Money and Improving Global Health Through Better Demand Forecasts*. Washington, DC, USA: Center Global Develop., 2007.

[9] S. Bag, S. Gupta, T.-M. Choi, and A. Kumar, "Roles of innovation leadership on using big data analytics to establish resilient healthcare supply chains to combat the COVID-19 pandemic: A multimethodological study," *IEEE Trans. Eng. Manage.*, to be published, doi: 10.1109/TEM.2021.3101590.

[10] P. Dutta, T.-M. Choi, S. Somani, and R. Butala, "Blockchain technology in supply chain operations: Applications, challenges and research opportunities," *Transp. Res. E, Logistics Transp. Rev.*, vol. 142, 2020, Art. no. 102067.

[11] R. Akkaoui, "Blockchain for the management of Internet of Things devices in the medical industry," *IEEE Trans. Eng. Manage.*, to be published, doi: 10.1109/TEM.2021.3097117.

[12] A. Park, J. Kietzmann, L. Pitt, and A. Dabirian, "The evolution of nonfungible tokens: Complexity and novelty of NFT use-cases," *IT Professional*, vol. 24, no. 1, pp. 9–14, Jan./Feb. 2022, doi: 10.1109/MITP.2021.3136055.

[13] M. Dowling, "Is non-fungible token pricing driven by cryptocurrencies?," *Finance Res. Lett.*, vol. 44, 2022, Art. no. 102097.

[14] M. Nadini, L. Alessandretti, F. Di Giacinto, M. Martino, L. M. Aiello, and A. Baronchelli, "Mapping the NFT revolution: Market trends, trade networks, and visual features," *Sci. Rep.*, 11, 2021, Art. no. 20902.

[15] W. Rehman, H. E. Zainab, J. Imran, and N. Z. Bawany, "NFTs: Applications and challenges," in *Proc. 22nd Int. Arab Conf. Inf. Technol.*, 2021, pp. 1–7, doi: 10.1109/ACIT53391.2021.9677260.

[16] S. S. Kanhere, "Keynote: Transparent, trustworthy and privacy-preserving supply chains," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops Affiliated Events*, 2021, pp. 654–654, doi: 10.1109/PerComWorkshops51409.2021.9431136.

[17] S. Sadri, A. Shahzad, and K. Zhang, "Blockchain traceability in healthcare: Blood donation supply chain," in *Proc. 23rd Int. Conf. Adv. Commun. Technol.*, 2021, pp. 119–126, doi: 10.23919/ICACT51234.2021.9370704.

[18] S. Chenthara, H. Wang, K. Ahmed, F. Whittaker, and K. Ji, "A blockchain based model for curbing doctors shopping and ensuring provenance management," in *Proc. Int. Conf. Netw. Netw. Appl.*, 2020, pp. 186–192, doi: 10.1109/NaNA51271.2020.00040.

[19] The MediLedger Project. Accessed: Apr. 22, 2022. [Online]. Available: https://www.mediledger.com/network

[20] W. Ni, X. Huang, J. Zhang, and R. Yu, "HealChain: A decentralized data management system for mobile healthcare using consortium blockchain," in *Proc. Chin. Control Conf.*, 2019, pp. 6333–6338, doi: 10.23919/ChiCC.2019.8865388.

[21] N. A. Asad, M. T. Elahi, A. A. Hasan, and M. A. Yousuf, "Permission-based blockchain with proof of authority for secured healthcare data sharing," in *Proc. 2nd Int. Conf. Adv. Inf. Commun. Technol.*, 2020, pp. 35–40, doi: 10.1109/ICAICT51780.2020.9333488.

[22] B. Vardhini, S. N. Dass, R. Sahana, and R. Chinnaiyan, "A blockchain based electronic medical health records framework using smart contracts," in *Proc. Int. Conf. Comput. Commun. Informat.*, 2021, pp. 1–4, doi: 10.1109/ICCCI50826.2021.9402689.

[23] G. Ramyasri and S. J. Hussain, "Access control of healthcare data using blockchain technology," in *Proc. 2nd Int. Conf. Smart Electron. Commun.*, 2021, pp. 353–357, doi: 10.1109/ICOSEC51865.2021.9591931.

[24] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in *Proc. 2nd Int. Conf. Open Big Data*, 2016, pp. 25–30, doi: 10.1109/OBD.2016.11.

[25] M. Younis, W. Lalouani, N. Lasla, L. Emokpae, and M. Abdallah, "Blockchain-enabled and data-driven smart healthcare solution for secure and privacy-preserving data access," *IEEE Syst. J.*, vol. 16, no. 3, pp. 3746–3757, Sep. 2022, doi: 10.1109/JSYST.2021.3092519.

[26] J. Cunningham et al., "Non-fungible tokens as a mechanism for representing patient consent," *Stud. Health Technol. Informat.*, vol. 294, pp. 382–386, 2022.

[27] Z. Y. Shae and J. J. P. Tsai, "On the design of medical data ecosystem for improving healthcare research and commercial incentive," in *Proc. IEEE 3rd Int. Conf. Cogn. Mach. Intell.*, 2021, pp. 124–131, doi: 10.1109/CogMI52975.2021.00024.

[28] N. Nizamuddin, H. R. Hasan, and K. Salah, "IPFS-blockchain-based authenticity of online publications," in Proc. Int. Conf. Blockchain, 2018, pp. 199–212, doi: 10.1007/978-3-319-94478-4_14.

[29] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Accessed: Apr. 23, 2022. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[30] H. R. Hasan et al., "A blockchain-based approach for the creation of digital twins," *IEEE Access*, vol. 8, pp. 34113–34126, 2020, doi: 10.1109/ACCESS.2020.2974810.

[31] G. Wang and M. Nixon, "SoK: Tokenization on blockchain," in *Proc. 14th IEEE/ACM Int. Conf. Utility Cloud Comput. Companion*, 2021, pp. 1–9.

[32] *Solidity Documentation: Release 0.8.14*, *Ethereum Foundation, Zug, Switzerland*. Accessed: Apr. 23, 2022. [Online]. Available: https://tinyurl.com/3v2m5zwp

[33] W. Entriken, D. Shirley, J. Evans, and N. Sachs, "EIP-721: Non-fungible token standard," *Ethereum Improvement Proposals*, no. 721, Jan. 2018. Accessed: Apr. 23, 2022. [Online]. Available: https://eips.ethereum.org/EIPS/eip-721

[34] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "Keccak," *Advances in Cryptology—EUROCRYPT 2013* (Lecture Notes in Computer Science), vol. 7881, T. Johansson and P. Q. Nguyen, Eds. Berlin, Germany: Springer, 2013.

[35] *MythX Guide*, MythX, Brooklyn, NY, USA, 2021. Accessed: Apr. 24, 2022. [Online]. Available: https://docs.mythx.io//

[36] *Smart Contract Vulnerability Coverage*, MythX, Brooklyn, NY, USA, 2021. Accessed: Apr. 24, 2022. [Online]. Available: https://mythx.io/detectors/

[37] J. Lockl, V. Schlatt, A. Schweizer, N. Urbach, and N. Harth, "Toward trust in Internet of Things ecosystems: Design principles for blockchain-based IoT applications," *IEEE Trans. Eng. Manage.*, vol. 67, no. 4, pp. 1256–1270, Nov. 2020, doi: 10.1109/TEM.2020.2978014.

[38] T. F. Stafford and H. Treiblmaier, "Characteristics of a blockchain ecosystem for secure and sharable electronic medical records," *IEEE Trans. Eng. Manage.*, vol. 67, no. 4, pp. 1340–1362, Nov. 2020, doi: 10.1109/TEM.2020.2973095.

[39] L. Soltanisehat, R. Alizadeh, H. Hao, and K.-K. R. Choo, "Technical, temporal, and spatial research challenges and opportunities in blockchain-based healthcare: A systematic literature review," *IEEE Trans. Eng. Manage.*, to be published, doi: 10.1109/TEM.2020.3013507.

[40] S. Biswas, K. Sharif, F. Li, Z. Latif, S. S. Kanhere, and S. P. Mohanty, "Interoperability and synchronization management of blockchain-based decentralized e-health systems," *IEEE Trans. Eng. Manage.*, vol. 67, no. 4, pp. 1363–1376, Nov. 2020, doi: 10.1109/TEM.2020.2989779.

[41] S. Alzahrani, T. Daim, and K.-K. R. Choo, "Assessment of the blockchain technology adoption for the management of the electronic health record systems," *IEEE Trans. Eng. Manage.*, to be published, doi: 10.1109/TEM.2022.3158185.

[42] T. -M. Choi, "Financing product development projects in the blockchain era: Initial coin offerings versus traditional bank loans," *IEEE Trans. Eng. Manage.*, to be published, doi: 10.1109/TEM.2020.3032426.

[43] S. Ramzan, A. Aqdus, V. Ravi, D. Koundal, R. Amin, and M. A. Al Ghamdi, "Healthcare applications using blockchain technology: Motivations and challenges," *IEEE Trans. Eng. Manage.*, to be published, doi: 10.1109/TEM.2022.3189734.

[44] M. Schaffer, M. Di Angelo, and G. Salzer, "Performance and scalability of private ethereum blockchains," in *Business Process Management: Blockchain and Central and Eastern Europe Forum*, New York, NY, USA: Springer, 2019.

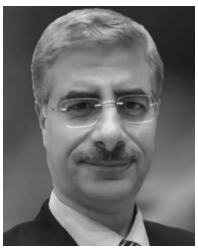
[45] *Scaling Overview*, Ethereum Foundation, Zug, Switzerland, Mar. 2022. Accessed: Apr. 25, 2022. [Online]. Available: https://ethereum.org/en/developers/docs/scaling/

Ahmad Musamih received the B.S. degree in electrical engineering from United Arab Emirates University, Al Ain, United Arab Emirates, in 2015, and the M.S. degree in engineering systems and management in 2018 from Khalifa University, Abu Dhabi, United Arab Emirates, where he is currently working toward the Ph.D. degree in engineering systems and management.

He is a Full-Time Researcher and a Graduate Student with the Department of Industrial and Systems Engineering, Khalifa University, where he is also a Research and Teaching Assistant. His research interests include blockchain, healthcare, management, and supply chain.

Ibrar Yaqoob (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Malaya, Malaysia, in 2017.

He is currently a Research Scientist with Khalifa University, Abu Dhabi, United Arab Emirates. He was a Research Professor with the Department of Computer Science and Engineering, Kyung Hee University, Seoul, South Korea, where he had joined as a Postdoctoral Fellow. He has been selected as a highly cited researcher worldwide by Clarivate (Web of Science). He works on the Editorial Boards of IEEE, Elsevier, and Springer journals, including *IEEE Network Magazine* and *Future Generation Computer Systems*. He has been involved in IEEE/ACM international conferences and workshops in various capacities, such as co-Chair, Track Chair, and Technical Program Committee Member. His current research focuses on leveraging blockchain, NFTs, and metaverse for healthcare, supply chain and logistics, IoT, and smart cities. He has also conducted research in the areas of mobile edge-cloud computing, IoT, computer networks, and big data.



Khaled Salah received the B.S. degree in computer engineering with a minor in computer science from Iowa State University, Ames, IA, USA, in 1990, and the M.S. degree in computer systems engineering and the Ph.D. degree in computer science from the Illinois Institute of Technology, Chicago, IL, USA, in 1994 and 2000, respectively.

He is currently a Full Professor with the Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates. He has more than 220 publications and three U.S. patents, has been giving a number of international keynote speeches, invited talks, tutorials, and research seminars on the subjects of blockchain, the Internet of Things, fog and cloud computing, and cybersecurity. He is now leading a number of projects on how to leverage blockchain for healthcare, 5G networks, combating deepfake videos, supply chain management, and artificial intelligence.

Dr. Salah was the Chair of the Track Chairs for 2018 IEEE Global Communications Conference. He is an Associate Editor for *IEEE Blockchain Tech Briefs* and a Member of the IEEE Blockchain Education Committee.

Raja Jayaraman received the B.Sc. degree from the University of Madras, Chennai, India, and the M.Sc. degree from Anna University, Chennai, India, both in mathematics, the Master of Science degree in industrial engineering from New Mexico State University, Las Cruces, New Mexico, and the Ph.D. degree in industrial engineering from Texas Tech University, Lubbock, TX, USA.

He is currently an Associate Professor with the Department of Industrial and Systems Engineering, Khalifa University, Abu Dhabi, United Arab Emirates. His postdoctoral research was centered on technology adoption and implementation of innovative practices in the healthcare supply chains and service delivery. He has led several successful research projects and pilot implementations in the area of supply chain data standards adoption in the U.S. healthcare systems. His research has appeared in top rated journals, including *Annals of Operations Research*, *IISE Transactions*, *Energy Policy*, *Applied Energy*, *Knowledge-Based Systems*, IEEE ACCESS, *Journal of Theoretical Biology*, and *Engineering Management Journal*. His expertise is in multicriteria optimization techniques applied to diverse applications, including supply chain and logistics, healthcare, energy, environment, and sustainability. His research interests include using blockchain technology, systems engineering, and process optimization techniques to characterize, model, and analyze complex systems with applications to supply chains, maintenance operations planning, and healthcare delivery.

Mohammed Omar is a Full Professor and the Founding Chair of the Department of Engineering Systems and Management (currently renamed as the Department of Industrial and Systems Engineering), Khalifa University, Abu Dhabi, United Arab Emirates . Prior to joining the Masdar Institute, Khalifa University, he was an Associate Professor and a Graduate Coordinator with Clemson University, Clemson, SC, USA. He was a part of the Founding Faculty Cohort of Clemson University Research Park, Greenville, SC. He has more than 100 publications in the areas of product lifecycle management, knowledge-based manufacturing, and automated testing systems, in addition to authoring several books and book chapters. He holds four U.S. and international patents. He was named as a Tennessee Valley Authority Fellow of two consecutive years during the Ph.D. research, in addition to being a Toyota Manufacturing Fellow. His professional career includes a Postdoctoral Service with the Center for Robotics and Manufacturing Systems and a Visiting Scholar with the Toyota Instrumentation and Engineering Division, Toyota Motor Company, Toyota, Japan. His group graduated seven Ph.D. dissertations and more than 35 M.Sc. theses. Four Ph.D. students are currently on academic ranks in U.S. universities. He has also led a National Science Foundation IUCRC Center and a part of the Department of Energy GATE Center of Excellence in Sustainable Mobility Systems. His current research group supported two Postdoctoral Scholar's Career Planning to become an Assistant Professor with Texas A&M at Qatar, Doha, Qatar, in 2013, and the University of Sharjah, Sharjah, United Arab Emirates, in 2015. His current research interests include capabilities in composite fabrication and manufacturing analytics at a laboratory Masdar City Campus.

Prof. Omar was the recipient of the Richard L. Kegg Award from the U.S. Society of Manufacturing Engineers, the SAE Foundation Award for Manufacturing Leadership, and the Murray Stokely Award from the College of Engineering, Clemson University. He is the Editor-in-Chief for *Journal of Materials Science Research* (Part of the Canadian Research Center) and an Associate Editor for *Soft Computing*, handling the areas of decision science and knowledge-based systems, in addition to his membership on several editorial boards and conference organizations. Furthermore, he serves on the Advisory Board for the Strata PJSC (part of Mubadala Aerospace).

Samer Ellahham received the bachelor's degree in biology and the M.D. degree from the American University of Beirut, Beirut, Lebanon.

He is currently a Cleveland Clinic Caregiver (Cleveland, OH, USA), seconded as Senior Cardiovascular Consultant, and the Director of Accreditation with the Quality and Safety Institute, Cleveland Clinic Abu Dhabi, Abu Dhabi, United Arab Emirates. He finished his internal medicine residency with Georgetown University Hospital-Washington Hospital Center, Washington, DC, USA, and his fellowship in cardiology with the Virginia Commonwealth University Health System, Richmond, VA, USA. He was with Georgetown University Hospital-Washington Hospital Center and in several clinical and leadership positions before moving to United Arab Emirates in 2008.

Dr. Ellahham is the Middle East Regional Chair, an ISQua Expert of the Patient Safety Movement Foundation, a member of the Aha Hospital Accreditation Science Committee, a member of the European Society of Cardiology Heart Failure Writing Group, an ex-Middle East Representative of the JCI Standards Subcommittee, and a member of the American College of Cardiology Accreditation Foundation Board.