IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 23, 2021, accepted October 16, 2021, date of publication October 20, 2021, date of current version November 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3121545

Blockchain-Based Solution for the Administration of Controlled Medication

AHMAD MUSAMIH[1], RAJA JAYARAMAN[1], KHALED SALAH[2], (Senior Member, IEEE), HAYA R. HASAN[2], IBRAR YAQOOB[2], (Senior Member, IEEE), AND YOUSOF AL-HAMMADI[2]

[1]Department of Industrial and Systems Engineering, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
[2]Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates

Corresponding author: Raja Jayaraman (raja.jayaraman@ku.ac.ae)

This work was supported by the Khalifa University of Science and Technology under Award CIRA-2019-001.

ABSTRACT Controlled drugs are open to abuse, misuse, and diversion. Therefore, they are regulated and tracked across the healthcare sector to protect the health of the general public which is a highly prioritized rule in the health professional's code of ethics. Healthcare centers that provide controlled medication to patients are still using manual papers to record controlled drugs production, delivery, prescription, administration, and disposal which causes delays in the system. Moreover, instances of controlled drugs misuse, abuse, and diversion still exist, which shows how the currently used system is inefficient in detecting such activities. Therefore, to ensure that the public health is safe and secure, an end-to-end system that tracks the whole healthcare supply chain is necessary. In this paper, we introduce a private Ethereum blockchain-based solution for the management of controlled medication. We ensure transparency, accountability, security, and data provenance by developing smart contracts that record all actions on an immutable ledger. We utilize off-chain storage, which is represented in the IPFS to store content that is large in size such as images. We present algorithms of the different phases in the proposed solution to illustrate how each phase will be carried out. We showcase the functionality of the proposed solution by performing tests and validating the smart contracts. We assess the performance of the proposed solution by conducting privacy, security, and confidentiality analysis. Performance evaluation shows that our solution is secure against common attacks and vulnerabilities and preserves the privacy and confidentiality of the patients. The smart contracts code is made publicly available along with the testing scripts.

INDEX TERMS Blockchain, controlled medication, drug traceability, Ethereum, medication administration, smart contracts.

I. INTRODUCTION

Controlled drugs can be a double-edged sword. They contribute in the better lives of many patients but if abused can lead to harm and possibly even addiction. Controlled substances are drugs whose manufacture, possession, and use are regulated by a government to protect the general public from harm which is an important rule in any health professional's code of ethics. Although they have many adverse effects, they constitute a core element in many healthcare systems to provide patients with valuable treatments. Therefore are commonly used in almost every healthcare facility. However, this common usage allows healthcare workers to misuse these drugs, which negatively affects the safety of patients, staff, and public health [1].

The Controlled Substances Act (CSA) categorizes controlled drugs into five main categories which are called schedules. Schedule I represents illicit drugs that are not allowed to be used medically, such as heroin. Schedule II drugs, such as morphine, can be used medically, however, they have high potential for abuse, and their acquisition, storage, and usage are strictly monitored. On the other hand, the higher schedules from III through V, classify drugs like codeine that have the least potential for abuse [2].

The addiction to controlled substances is the main reason why healthcare professionals tend to divert controlled substances. In most cases, controlled substances are diverted to satisfy the health professional's substance use disorder rather than trafficking [3]. Although healthcare centers attempt to

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks.

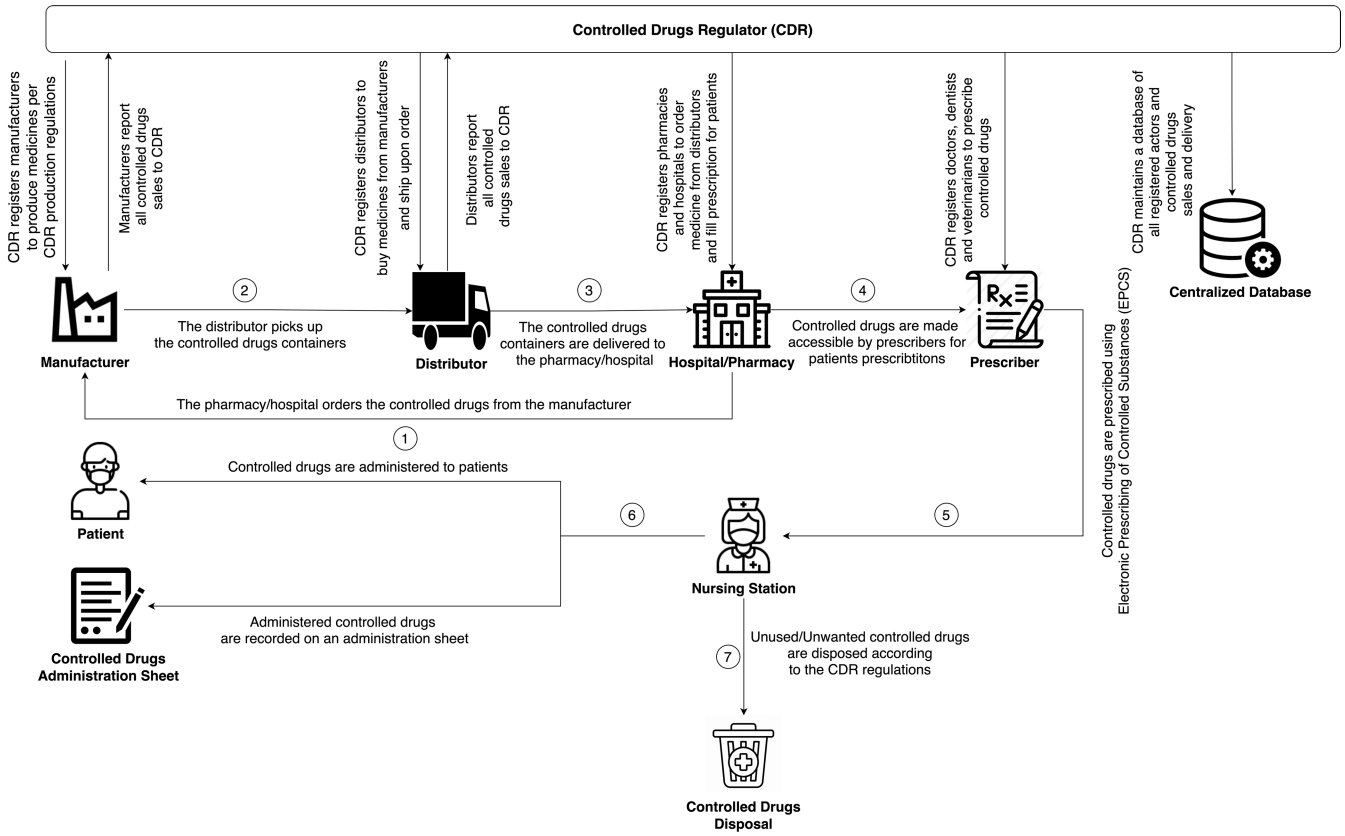


FIGURE 1. Flow diagram for the production and consumption of controlled drugs.

log the movements of controlled substances, there are still cases of theft and misuse. Manual documents are still being used by healthcare centers to record controlled drug supply, delivery, and disposal. Moreover, filling these documents manually causes delays in the dispensing process [4].

Figure 1 is a comprehensive illustration of the movements of controlled drugs from the moment the manufacturer is permitted to produce them until their consumption in a healthcare center whether by administering them to patients or disposing them. The Controlled Drugs Regulator (CDR) will have to register all the actors within the system such as the manufacturer, distributor, and hospital/pharmacy, and each one of those actors will have to report any actions they take to the CDR in the form of a report which is then stored in a centralized database that is controlled by the CDR. A typical flow of actions within this system is twofold; a production phase and consumption phase. The former begins with the hospital/pharmacy placing orders for controlled drugs which are received and prepared by the manufacturer. After that, the controlled drug containers are picked up by a distributor who is responsible for delivering the containers to the hospital/pharmacy, whereas the latter begins with the controlled drugs being accessible by the prescriber who then prescribes the controlled drugs to patients through an Electronic Prescribing of Controlled Substances (EPCS). The EPCS is then received by the nursing station to administer the controlled drugs to patients and record all the necessary information in a Controlled Drugs Administration Sheet. Finally, according to the CDR regulations, any unused or unwanted drug must be disposed properly according to predefined guidelines [5]–[7].

Despite having many attempts and efforts, controlled drug diversion still exists within the healthcare system. For example, a solution proposed in [8] suggests that early identification of controlled drug diversion can help in mitigating the issue by hiring a full team of professional staff who will continuously monitor the healthcare system to identify cases of controlled drug diversion, however, this solution is susceptible to human error and very costly if implemented at a large scale. Moreover, in [9], the American Society of Health System Pharmacists (ASHP) proposed guidelines that if followed will act as a preventive measure to the diversion of controlled substances. However, their guidelines require hiring trusted professionals who should manually record and monitor all procedures to prevent diversion of controlled substances, which might eventually add more complexity to the system and bring things back to the hired employees who potentially are the actual source of the problem. In this work, the proposed solution makes all actors accountable for their actions and any error identified, whether intentional or not, can be easily tracked and traced back. This is achieved by leveraging the intrinsic features of the blockchain

technology where all actions stored on an immutable ledger are tamper-proof.

Preventing and detecting controlled substances diversion requires overcoming challenges in transparency, traceability, accountability, data provenance, and trust. In this framework, the main goals of our work are described as follows:

- We present a blockchain-based solution that enables tracing the production and consumption phases of controlled drugs in a transparent, auditable, reliable, and secure environment.
- We design and develop smart contracts that register actors and generate events for all critical actions throughout the production and consumption phases.
- We integrate decentralized off-chain storage to store large-sized content such as images.
- We implement our solution using a private Ethereum blockchain to eliminate deployment and execution costs.
- We utilize the Proof of Authority(PoA) consensus algorithm to reduce energy consumption and put the authority of miners at stake to ensure trust and security.
- We present algorithms for all the main functions, complete implementation, testing, and validation details. The smart contracts code is made publicly available.[1]
- We conduct a general security analysis for the proposed system to evaluate its robustness against common security attacks and vulnerabilities. Moreover, smart contract security analysis is conducted to ensure no critical flaws exist within the smart contract.
- We showcase how our solution can be generalized for use in other security critical applications that demand a trusted environment to trace and track any misuse or diversion of sensitive items or products.

The rest of the paper is structured as follows. Section II presents the related work to controlled drugs diversion. Section III illustrates the proposed solution for the controlled drugs. Section IV illustrates the full implementation of our solution. Section V showcases testing results and evaluates the overall performance of the smart contracts. Section VI discusses and analyze the overall performance of the proposed solution. Finally, section VII sums up the contributions of our work, concludes the paper, and describes the directions of the future research.

II. RELATED WORK

We present a comprehensive review of existing work addressing issues in controlled medication management. Blockchain and non-blockchain solutions are both included and the efforts are categorized accordingly.

A. NON-BLOCKCHAIN-BASED EFFORTS IN CONTROLLED MEDICATION MANAGEMENT

In the context of non-blockchain-based efforts that improve controlled medication management, a number of reviews that discuss the current issues related to controlled medication are published and different methods and tools are used to provide solutions for controlled medication management. A notable effort that came into existence on May 1, 1971, is the establishment of the Controlled Substances Act which aims to enhance the manufacturing, importation, exportation, distribution, and dispensing controlled substances [10]. Gabay has described controlled substances registration and schedules, ordering and record keeping, and dispensing [5]–[7]. In [5], controlled substances have been categorized into five main groups. Schedule I includes substances with high abuse potential and no medical use and therefore these medications cannot be prescribed, dispensed, or administered. For example, Heroin and Marijuana. Schedule II includes substances with high abuse potential and acceptable medical use and therefore these medications can be prescribed, dispensed, or administered. For example, Morphine, Codeine, and Hydrocodone. Schedule III includes substances with moderate abuse potential, which is less than Schedule II but higher than Schedule IV. For example, Anabolic steroids and Ketamine. Schedule IV includes substances with moderate abuse potential, which is less than Schedule III but higher than Schedule V. For example, Propoxyphene and Butorphanol. Schedule V includes substances with the least potential for abuse. For example, Robitussin AC and Phenergan with Codeine.

Every pharmacy that dispenses controlled medications is required to register with the DEA [10]. In [6], the ordering and record keeping processes are described. Schedule II substances can be only ordered by filling a special form [11] which is then used for every distribution, purchase, or transfer. Moreover, these forms are required to be separated from other forms. For Schedules III to V controlled substances, a receipt is required to be kept by the pharmacy that is ready for retrieval at any time. In addition to that, every transaction of a controlled substance must be kept for at least two years. In [7], the dispensing requirements, electronic prescriptions, and fraudulent prescriptions are described. Similar to other medications, prescriptions of controlled medications are required to contain the pharmacy name and address, the number of prescription, patient name, dispensing date, prescriber name, use instructions, and cautionary statements. The only difference between controlled medications and other medications is that the US Food and Drug Administration (FDA) requires a label with the following statement: "CAUTION: Federal law prohibits the transfer of this drug to any person other than the patient for whom it was prescribed." to be attached with controlled substances.

In [12], guidelines for safety and management of controlled medications are described. It mainly focuses on prescribing, administering, recording, and monitoring. The guidelines mainly provide users with recommendations on how to best handle controlled medications. Kaufman suggests and describes the use of Electronic Prescription for Controlled Medications [13]. The work proposed by Peter Kaufman aims to develop and verify an electronic prescribing system that

[1] https://github.com/DrugTraceability/ControlledMedications/blob/main/Code

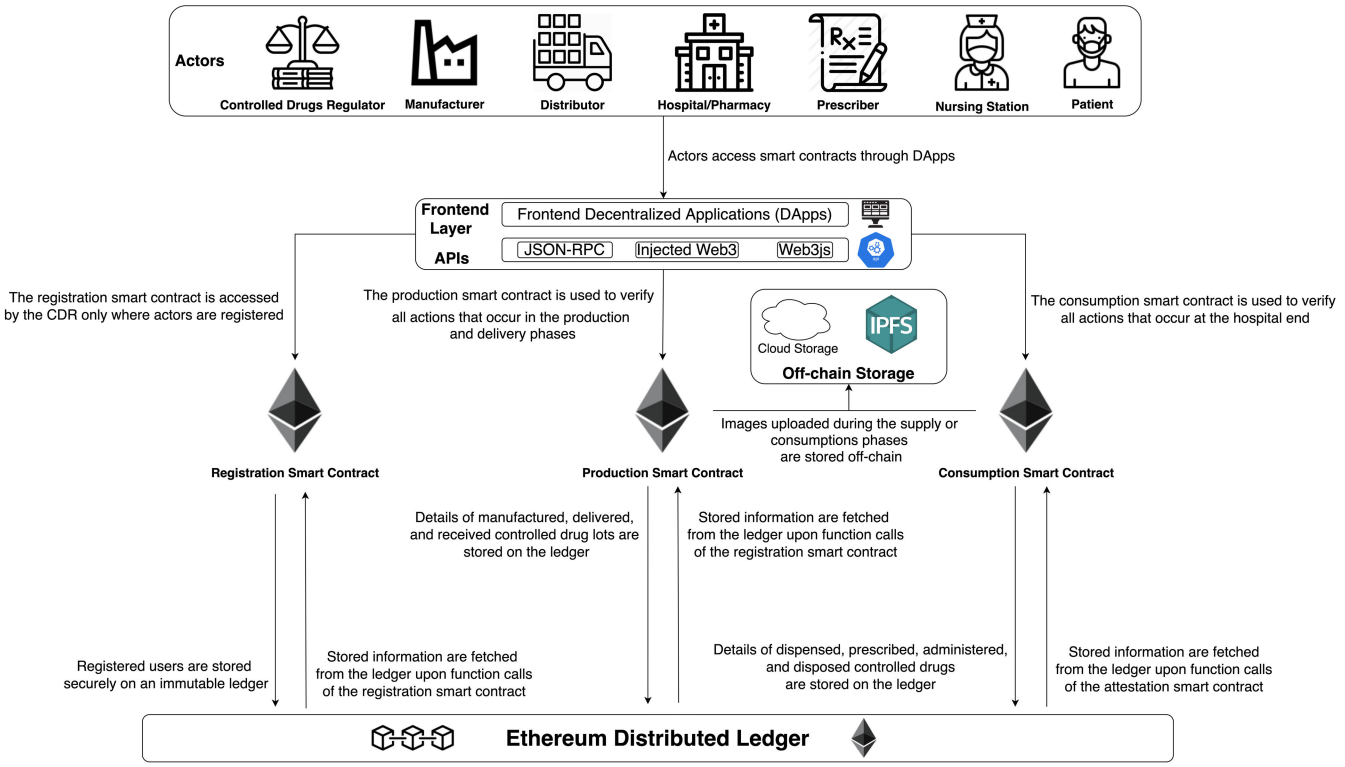


FIGURE 2. High-level system architecture of the blockchain-based solution for controlled medication.

allows secure and safe electronic prescription of controlled substances which is abbreviated as EPCS. Moreover, the proposed work aims to link the suggested solution with the existing Massachusetts Prescription Monitoring Program for Schedule II controlled substances.

In [25], the Centers for Disease Control and Prevention (CDC) proposes a prescription drug monitoring program called PDMP, which is an electronic database that tracks controlled drugs prescriptions. The solution promises to be universal and can be used across different countries. Moreover, the solution provides real-time tracking for dispensing controlled drugs to patients. Finally, the solution is expected to be easily used and accessed.

Non-blockchain-based solutions have been relying on documenting any actions taken related to controlled medication on paper documents which can make handling controlled medications cumbersome. Moreover, using electronic documents instead of paper documents can help in storing and retrieving data when needed, however, it adds a security issue since these documents are stored in a centralized database and can be vulnerable to cyber attacks.

B. BLOCKCHAIN-BASED EFFORTS IN CONTROLLED MEDICATION MANAGEMENT

In the context of blockchain-based efforts that aim to improve the security and safety of controlled medications, a number of reviews and potential solutions have been published and proposed. The authors in [14] describe how blockchain technology can improve the security and safety of controlled medications. First, they explain what can go wrong in a pharmacy while handling controlled medications. For example, the registration process can include discrepancies and errors in dispensing controlled medications due to increased pressure on the pharmacy. Secondly, the authors explain the potential benefits of blockchain in this specific use case. For example, reducing complexity and costs, reducing errors, enhancing security, improving resilience, sharing trusted transactions, creating audit trail for transactions, and improving transparency between actors.

The authors in [15] proposed a solution using blockchain technology to create a secure distributed ledger that allows tracking and tracing controlled medications throughout their life cycle. Such a solution should eliminate the use of paper in documenting any event that is related to controlled medication. However, the article does not have a real implementation of the proposed solution and is only limited to suggestions of how blockchain technology can improve controlled medication management.

The authors in [16] proposed a blockchain-based framework called SecureRx for secure access to prescription records. The proposed framework allows its users to access the patient history to decide if it is appropriate to prescribe the medication to the patient or not. The results showed that the proposed solution is functional and it has an adequate scalability when the number of nodes is small. However, the proposed solution was built on a private Ethereum blockchain

that utilizes the Proof of Work (PoW) consensus algorithm which means there are gas fees for every transaction which is paid as an incentive for the miners. It is mentioned in the paper that Ropsten testnet will be used along with Ether faucet to provide users with the needed Ethereum for gas fees, however, in a real use case this is not possible because the miners will simply only accept Ether that is useable on the Mainnet.

Singh *et al.* [17] proposed an Internet of Things (IoT) sensor-based blockchain framework that tracks and traces drugs as they are transferred through the supply chain. The raft consensus algorithm is used, however, this consensus algorithm is only suitable for small-sized networks which is not the case with supply chains. Therefore, the authors decided to use bloXroute server to solve this scalability issue which utilizes blockchain servers that are low-latency and high capacity. The platform used for the proposed solution is Hyperledger Fabric. One drawback for using this approach is that renting blockchain servers can be quite costly.

Zhu *et al.* [18] proposed a blockchain-based solution for medication anti-counterfeiting and traceability. The proposed solution leverages smart contract to code all the required functions, and the consensus algorithm used is an enhanced practical Byzantine fault tolerance (PBFT) instead of the normal PBFT to reduce the amount of communication among the nodes. This enhanced consensus algorithms works by using cumulative points for the participating, and dishonest nodes get dropped out. One bottleneck for using the PBFT consensus algorithm in general is that as the number of nodes increases the number of transactions per second (TPS) decreases. Finally, the proposed solution requires relatively large data storage and no off-chain storage such as IPFS was utilized, and storing such data on-chain can be quite costly.

Sahoo *et al.* [19] proposed a blockchain-based model to eliminate drug counterfeiting. The proposed solution does not have a real implementation, however, it discusses how blockchain technology can be used to eliminate drug counterfeiting. The proposed approach suggests dividing the supply chain into the main participating actors excluding the consumer to reduce complexity and data storage on the blockchain.

Uddin *et al.* [20] described the role of blockchain technology in drug traceability along with its architectures and open challenges. The paper presents an overview of product traceability issues that exist within the pharmaceutical supply chain and it explains how blockchain technology can be leveraged to provide data provenance, track and trace drugs to mitigate the issue of counterfeiting. Moreover, The paper discusses two blockchain architectures, Hyperledger Fabric and Hyperledger Besu to identify the more appropriate for meeting the critical requirements for drug traceability. The critical requirements include privacy, trust, transparency, security, authorization, authentication, and scalability. Additionally, blockchain challenges in pharmaceutical supply chains are described and discussed.

Jamil *et al.* [21] proposed a novel medical blockchain model for drug supply chain integrity management in a smart hospital. The proposed solutions uses Hyperledger Fabric architecture that is based on blockchain technology to implement the solution. The solution showed good results in terms of transactions throughput and latency, however, the solution was tested with a small-sized network, and a real use case would normally include a large number of participants. One of the disadvantages of Hyperledger Fabric is that it is still not adopted like the Ethereum blockchain, therefore, its performance in real cases is very hard to predict and test.

III. A BLOCKCHAIN-BASED SOLUTION FOR THE ADMINISTRATION OF CONTROLLED MEDICATION

In this section, we present our proposed solution for the management and administration of controlled medication. A high-level illustration of the system architecture is shown in Figure 2. As can be seen in the figure, there are three smart contracts used in our solution namely Registration, Production and Consumption smart contracts (SCs). Moreover, frontend Decentralized Applications (DApps) are to be used by the actors to access the functions and events of the registration, production, and consumption SCs. In addition, actors who are required to document their actions in the form of a digital image will be granted access to off-chain storage. Web3.js libraries are used in the process of building DApps to allow interactions with the smart contracts over HTTP (e.g., Metamask) or IPC connection (e.g., Mist). Each smart contract will have unique functions that are only executable by pre-authorized actors. Moreover, any large-sized content will be stored off-chain to optimize the performance of the smart contracts and reduce costs if the smart contracts are to be built on a "Mainnet" that requires gas fees. The components of the proposed solution are described below.

- **Actors:** The actors include a Controlled Drug Regulator (CDR), manufacturers, distributors, hospitals/pharmacies, prescribers, and nursing stations. Each actor will have a special role in the controlled drug medication supply chain which will grant them access rights to restricted functions in the smart contracts. Moreover, actors will have the privilege of accessing information that is stored on-chain which allows them to view logs, transactions, and events. Additionally, actors can access images that are stored off-chain which includes information related to the controlled drug only, and no information related to the patients will be stored to preserve their privacy and confidentiality.
- **Frontend DApp:** The frontend DApp is the User Interface (UI) which allows actors to access their pre-authorized functions in a user friendly way. Moreover, the DApp will access the needed functions and events by using web3.js libraries.
- **Ethereum Distributed Ledger:** The Ethereum distributed ledger is the core component of the proposed

solution. All logs, transactions, and events are permanently stored on the ledger and they cannot be altered or removed. Therefore, this component provides traceability, accountability, and transparency to the proposed solution. A private Ethereum blockchain also adds authorization, confidentiality, and privacy.

- **Registration Smart Contract:** This smart contract is responsible for giving permission to actors in the system. It registers new actors and assigns them to specific roles which allows them to access restricted functions based on their newly assigned roles.
- **Production Smart Contract:** This smart contract is responsible for the manufacturing and delivery processes of the controlled medication. First, the manufacturer will have to produce the controlled medication Lot. Second, the distributor will deliver the controlled medication lot to the hospital/pharmacy. Finally, the hospital/pharmacy will confirm the reception of the controlled medication Lot. Every step in this process will be logged and stored on the ledger and off-chain storage is used if deemed necessary. For every controlled drug Lot produced, an image is stored off-chain and its Interplanetary File System (IPFS) hash is stored on-chain for tracing purposes.
- **Consumption Smart Contract:** This smart contract is responsible for the administration of controlled medication to patients and the disposal of unwanted or unused controlled medication. First, the prescriber will prescribe the controlled medication to a patient with all necessary details. Second, the nursing station will have access to the details of the prescription and administer the controlled medication to the patient accordingly. Finally, the unused or unwanted controlled medication is disposed according to the CDR regulations. All the previous steps are logged and stored on the ledger and off-chain when necessary. Any storage off-chain is accompanied with a hash on-chain for data provenance.
- **Off-chain Storage:** The off-chain storage is needed in the proposed solution when images in the production or consumption phases are needed to be stored for traceability purposes.

These components all together ensure that the origin of the controlled medication is traceable, all actors are accountable for their actions, and all administered and disposed controlled drugs can be traced.

The following subsections illustrate the details of the used blockchain, the consensus algorithm, and a detailed sequence diagram where interactions among the system components are presented.

A. PRIVATE PROOF-OF-AUTHORITY ETHEREUM NETWORK

Controlled medication prescriptions and information in general are considered highly confidential and should not be accessed by unauthorized entities. Therefore, a private blockchain fits the needs of such an application because the stored information are invisible to the public and only permissioned users are allowed to view it. Ethereum blockchain offers the option for enterprises to build their own private blockchain to improve security, privacy, and confidentiality. Moreover, private blockchains have the option of using consensus algorithms other than Proof-of-Work which is used by the ''Mainnet'' of the Ethereum Blockchain. For example, the Proof-of-Authority is a consensus algorithm that is commonly used for private enterprises because it reduces energy consumption and provides higher transaction throughput because it does not need the nodes to do any mining work, rather, it only requires them to stake their identity reputation in exchange for becoming a validator [12].

B. ACTORS INTERACTIONS IN THE PROPOSED SOLUTION

The main interactions among the actors within the proposed solution are illustrated in Figure 3 and they can be divided into three main phases as described below.

- **Registration:** In the registration phase, the CDR registers the authorized manufacturers, distributors, and hospitals/pharmacies. Moreover, the authorized hospital/pharmacy will be eligible for registering prescribers and nurses. Once all actors are registered, they will have access to pre-authorized functions in the next two phases.
- **Production:** In the production phase, the manufacturer is responsible for producing controlled medication lot and storing its information on the Ethereum network and uploading images to the IPFS. Moreover, the distributor is responsible for handling the delivery process of the controlled medication lot from the manufacturer to the hospital/pharmacy. Finally, the hospital/pharmacy will confirm the reception of the controlled medication Lot.
- **Consumption:** In the consumption phase, the hospital/pharmacy will announce that the controlled medication is ready for dispensing. After that, the prescriber stores the amount of doses needed for the patient, and the patient is identified by an Ethereum Address which will not reveal any personal information about the patient. Next, the nurse will administer the controlled medication to the patient based on the provided prescription, and details such as the number of doses administered and patient Ethereum Address are stored on the Ethereum network, and an image of the administered controlled medication is stored on the IPFS. Finally, the unused or unwanted controlled medication is disposed, and the Ethereum Address of the nurse disposing the controlled medication and the amount disposed are stored on the Ethereum network, moreover, an image of the disposed controlled medication is stored on the IPFS.

These interactions with the three smart contracts will ensure that only authorized actors can access the smart contracts functions, the origin of the controlled medication can be traced, and all actors are accountable for their actions.

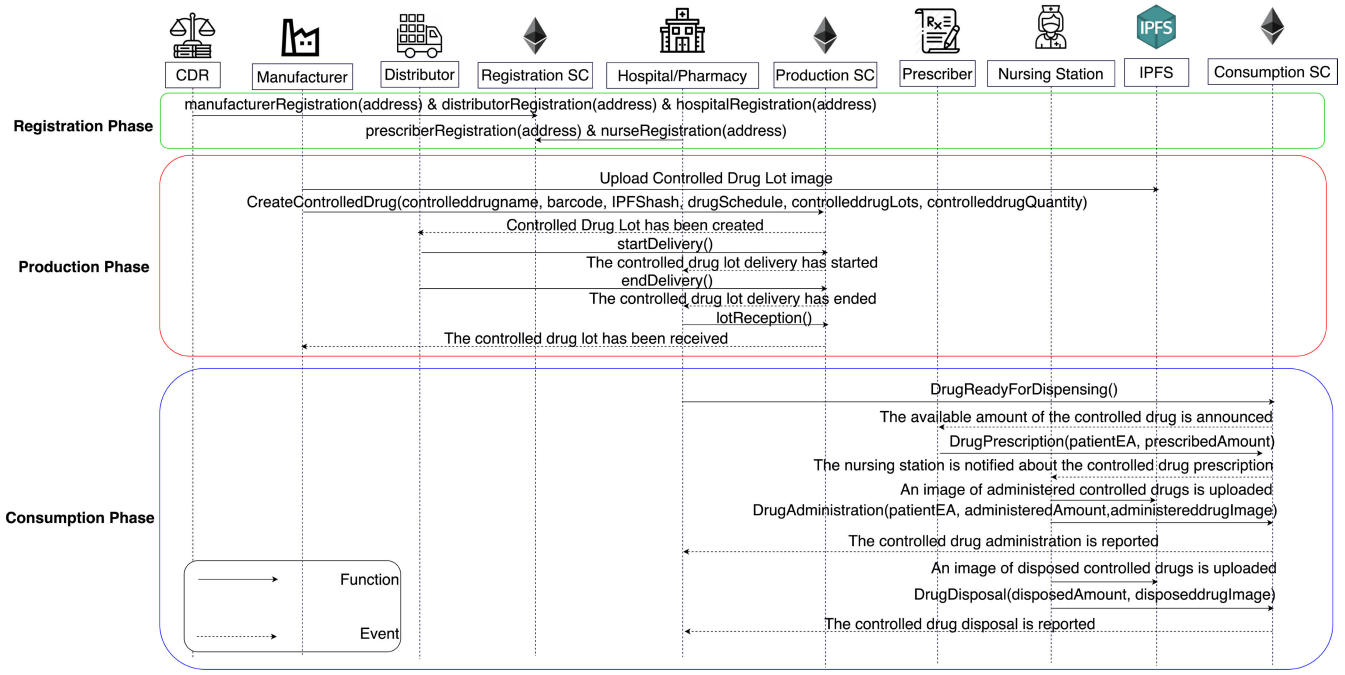


FIGURE 3. The sequence diagram of all the interactions in the proposed solution.

C. TECHNICAL DESCRIPTION OF THE PROPOSED SOLUTION

Figure 4 describes the in-depth details of our proposed solution. In the front-end, the authorized actors are able to access their pre-authorized functions by using a DApp. The DApp can either be a mobile application or a web browser application (Webapp). DApps are built using the web3.js which is a collection of libraries that contains the necessary tools to fetch logs and events from the smart contracts by interacting with a local or remote Ethereum node. On the other hand, the back-end includes Metamask which is an injected Web3 provider. The role of Metamask is to make it easier for DApp developers to access the Ethereum blockchain without running a full Ethereum node locally. Metamask uses the Infura gateway by default to interact with the Ethereum blockchain via JSON-RPC. Infura is responsible for fetching the required information from the blockchain which are then provided to the DApp users via Metamask.

In our solution, a private Ethereum network is used, therefore, the public Ethereum address of the user accessing the DApp must be registered. Finally, IPFS is used to store large-sized contents such as images. It works by DApp users uploading the content to the IPFS, then they are provided with an IPFS hash which is stored on the Ethereum blockchain, and when the content is needed, the Ethereum blockchain refers to it via the IPFS hash.

IV. IMPLEMENTATION OF THE PROPOSED SOLUTION

The proposed solution is developed by using a private Ethereum network where only authorized actors and validation nodes are added to the network. The smart contracts are written in Solidity language and compiled and tested using REMIX IDE which is an online web-based development environment for writing and executing smart contracts. The full code[2] has been made publicly available on Github.

A. PRIVATE PROOF OF AUTHORITY ETHEREUM NETWORK SETUP

In our solution, a private PoA Ethereum network is setup locally with a single validation node to deploy the smart contracts and only permit authorized nodes to validate and confirm transactions. Moreover, because of the high sensitivity and confidentiality of the stored information, a private blockchain is used. Additionally, PoA consensus algorithm is used instead of PoW to reduce energy consumption, and to improve scalability and transaction throughput [22].

To successfully setup and run a private PoA Ethereum network, a set of tools were used. First, **Geth** (The Go implementation of the Ethereum protocol) is used to create the node where public and private keys are generated and a password is setup. Second, **Puppeth** (The Ethereum private network manager) is used to create the genesis file, which is then used to initialize the blockchain. The very first block in the blockchain is called the genesis block and it is created based on the given parameters to the Puppeth tool. Puppeth allows the user to choose between PoW and PoA consensus algorithms, setting the average block time, choosing the validating/sealing nodes, and specifying the chain/network ID. Finally, by using Geth, the validating

[2]https://github.com/DrugTraceability/ControlledMedications/blob/main/Code

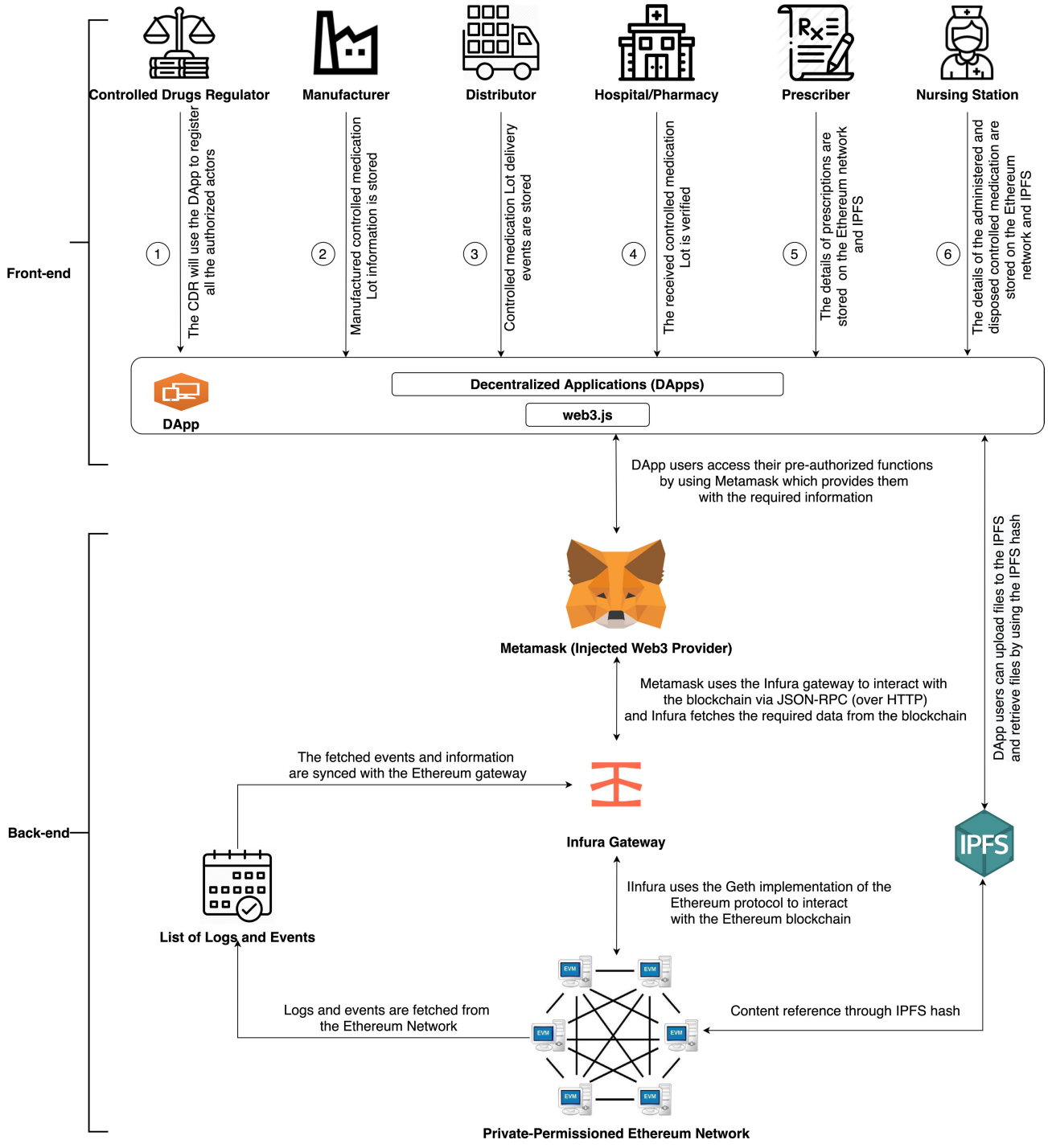


FIGURE 4. In-depth details of the front-end and back-end of the proposed solution.

node needs to be initialized in the created genesis file by Puppeth.

B. IMPLEMENTATION DETAILS

Once the private Ethereum blockchain is setup, authorized actors will be able to interact with it by using Metamask. This can be achieved by setting up a new network in Metamask's settings where the network name, RPC URL, and network/chain ID are specified. When the metamask is successfully connected to the added network, the users will be able to deploy smart contracts and execute functions based on their permissions.

The CDR will first deploy the Registration smart contract and assign manufacturers, distributors, and hospitals/pharmacies by running their respective functions. After that, the hospital/pharmacy will have to assign prescribers

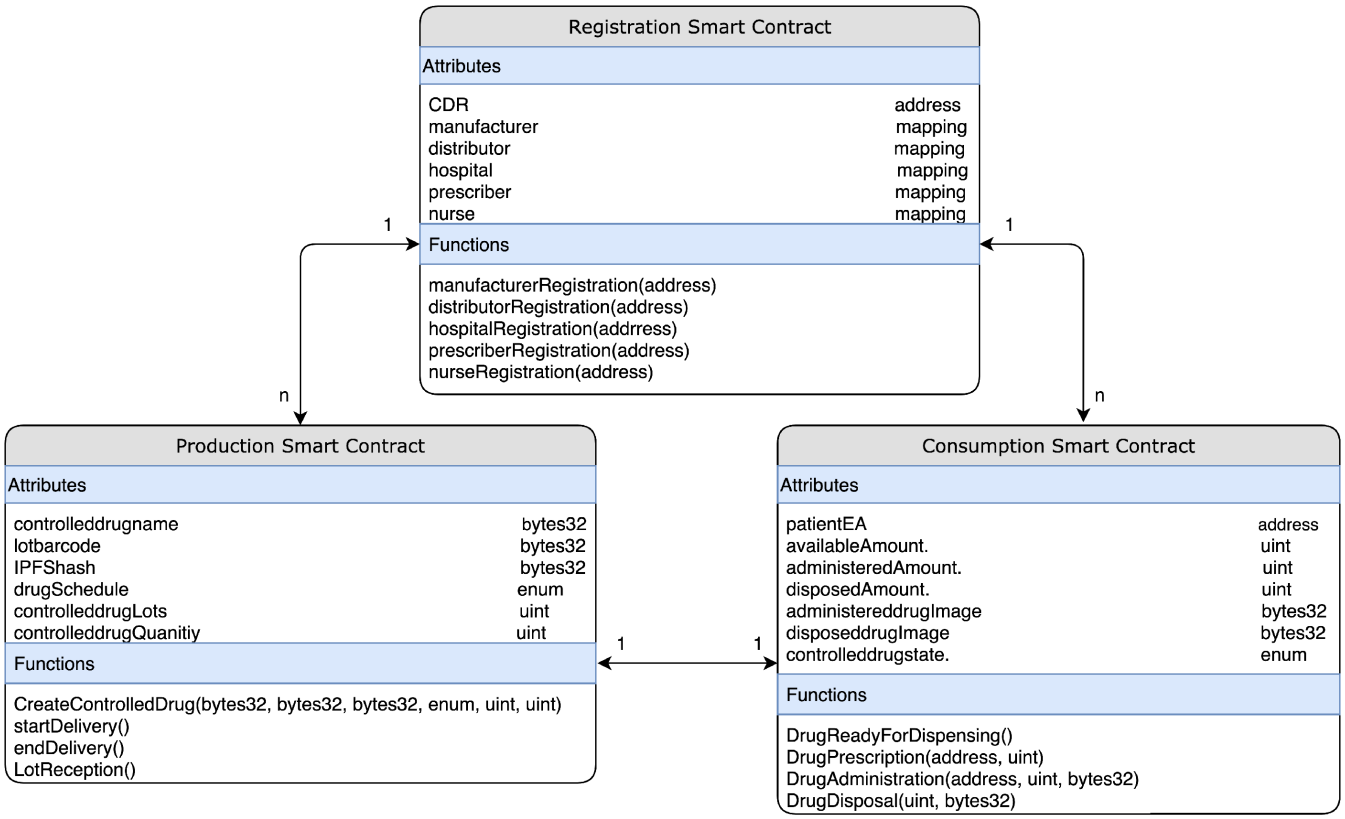


FIGURE 5. Entity-relationship diagram.

and nurses to permit them to execute their respective functions.

The manufacturer deploys the Production smart contract and creates a controlled medication lot where its name, barcode, IPFShash, schedule type, and quantity are all specified and an event is emitted with all the details. Next, the delivery process is handled by the distributor by running the start and end delivery functions and all the necessary information and logged and stored in the form of an event which is then emitted to all the concerned actors. Finally, the hospital/pharmacy announces the reception of the controlled medication Lot by running a function that emits an event with all the required details.

The hospital/pharmacy deploys the Consumption smart contract. The hospital/pharmacy updates the state of the controlled medication lot to ''Ready for dispensing'' by executing a function that emits an event with all the required details. After that, the prescriber will be able to provide patients with prescriptions based on the available amount of controlled medication and emit an event with the details of the prescription. Then, the nursing station handles the administration process by accessing the prescription details, and then an event is emitted with all the details once the administration process is finished. Finally, any unwanted or unused controlled medication will be disposed, and the details of the disposal process are all logged and stored.

Figure 5 illustrates the relationship between the different actors and smart contracts. The Registration smart contract will be deployed with a set of attributes as shown in the figure. There can only be one CDR in the system, therefore, it is declared as an address. The rest of the actors are declared as mappings because there can be several of them in the system. Moreover, the Registration smart contract has four main functions, *manufacturerRegistration*, *distributorRegistration*, *prescriberRegistration*, and *nurseRegistration* which are used to register the actors based on their roles in the system.

The Production smart contract has a set of attributes that are used to define the details of the controlled medication. In addition to that, the *regcontract* attribute is necessary to allow actors to access their respective functions based on the permissions given to them in the Registration smart contract. Furthermore, there are four main functions in the Production smart contract, *CreateControlledDrug*, *startDelivery*, *endDelivery*, and *LotReception* which are used to create, deliver, and receive the controlled medication Lot by the manufacturer, distributor, and hospital/pharmacy respectively.

The Consumption smart contract has a set of attributes that are used for preparing prescriptions for the patients, administer the controlled medication to the patients, and dispose the controlled medication when needed. There are

four main functions in the Consumption smart contract, *DrugReadyForDispensing*, *DrugPrescription*, *DrugAdministration*, and *DrugDisposal*.

Finally, the Registration smart contract will have a 1:n relationship with the other two smart contracts because there can only be one Registration smart contract, whereas multiple Production and Consumption smart contracts can exist for different controlled medications. Moreover, the Production and Consumption smart contracts have a 1:1 relationship because each controlled medication can only have 1 production and consumption smart contracts.

C. PROPOSED SOLUTION ALGORITHMS

To further clarify the details of the implemented functions in the smart contracts, various algorithms are developed and presented. The following are the main algorithms of our proposed solution.

- **Registration Algorithm**
  Algorithm 1 represents the registration phase of the proposed solution. In this phase of the solution, manufacturers, distributors, hospitals, prescribers, nurses, and the CDR need to be assigned. It should be noted that unlike the other entities, the CDR gains access to registration functions by simply deploying the smart contract, namely, *manufacturerRegistration*, *distributorRegistration*, and *hospitalRegistration* where the Ethereum addresses of the actors are assigned to their respective roles. Moreover, the actor with the hospital Ethereum address will need to assign prescribers and nurses by using the *prescriberRegistration* and *nurseRegistration* functions. This algorithm is necessary to permit only the registered actors to execute functions and view logs and events. All actors (except the CDR) are registered in a mapping because multiple actors with the same role can exist in the same system. Finally, registration functions can be accessed at any time to register new actors as needed to avoid the need for deploying new smart contracts later on.
- **Production Algorithm**
  Algorithm 2 represents the production phase. In this phase, the controlled drug Lot must be registered properly to identify its source at later stages. The manufacturing process is represented in the *CreateControlledDrug* function where the manufacturer needs to add the *controlleddrugname*, *lotbarcode*, *IPFShash*, *schedule*, *controlleddrugLots*, and *controlleddrrugQuantity*. The controlled drug Lot goes through different states that are identified in the *lotstate* variable which is necessary to inform other entities of the current state of the controlled drug Lot. Therefore, the manufacturer needs to update the *lotstate* from "NotReady" to "Manufactured" so that other concerned actors are notified. Then, an event is emitted declaring the end of the manufacturing process with all the details of the manufactured controlled drug Lot. The second step in the production phase is the delivery process where the manufactured controlled drug Lot is delivered from the manufacturer to the hospital/pharmacy. The delivery process is represented in three functions, *startDelivery*, *endDelivery*, and *LotReception* where the distributor is able to start the delivery if the *lotstate* is "Manufactured". Once the delivery process starts, the *lotstate* is updated to "EnRoute" to inform actors that the controlled drug Lot is currently being delivered. Then, when the distributor arrives at the hospital, the *lotstate* is updated to "DeliveryEnded" and the hospital is informed. The last step in the production phase is the reception of the controlled drug Lot by the hospital/pharmacy, where the hospital

Algorithm 1 Registration

Input: *manufacturer* EA, *distributor* EA, *hospital* EA, *prescriber* EA, *nurse* EA, *CDR* EA

Output: *registeredManufacturers_list*: A list of manufacturers *EAs* in the form of mapping, *registeredDistributors_list*: A list of distributors *EAs* in the form of mapping, *registeredHospitals_list*: A list of hospitals *EAs* in the form of mapping, *registeredPrescribers_list*: A list of prescribers *EAs* in the form of mapping, *registeredNurses_list*: A list of nurses *EAs* in the form of mapping

initialization **if** *caller* == *CDR* **then**
  | *registeredManufacturers_list*[*manufacturer*] = *true*
else
  └ Revert contract state and show an error.
/* Manufacturer registration is complete */
if *caller* == *CDR* **then**
  | *registeredDistributors_list*[*distributor*] = *true*
else
  └ Revert contract state and show an error.
/* Distributor registration is complete */
if *caller* == *CDR* **then**
  | *registeredHospitals_list*[*hospital*] = *true*
else
  └ Revert contract state and show an error.
/* Hospital registration is complete */
if *caller* == *hospital* **then**
  | *registeredPrescribers_list*[*prescriber*] = *true*
else
  └ Revert contract state and show an error.
/* Prescriber registration is complete */
if *caller* == *hospital* **then**
  | *registeredNurses_list*[*nurse*] = *true*
else
  └ Revert contract state and show an error.
/* Nurse registration is complete */

Algorithm 2 Production

Input: *controlleddrugname*, *lotbarcode*, *IPFShash*, *schedule*, *controlleddrugLots*, *controlleddrrugQuantity*, *lotstate*

Output: Events declaring the end of the manufacturing and delivery processes

initialization **if** *caller* == *manufacturer* ∧ (*lotstate* == *NotReady*) **then**
  Add *controlleddrugname*, *lotbarcode*, *IPFShash*, *schedule*, *controlleddrugLots*, and *controlleddrugQuantity*
  Update *lotstate* to "Manufactured"
  Emit an event declaring the end of the manufacturing process
else
  Revert contract state and show an error.
/* Manufacturing process is complete */
if *caller* == *distributor* ∧ (*lotstate* == *Manufactured*) **then**
  Update *lotstate* to "EnRoute"
  Emit an event declaring the start of the delivery process
else
  Revert contract state and show an error.
if *caller* == *distributor* ∧ (*lotstate* == *EnRoute*) **then**
  Update *lotstate* to "DeliveryEnded"
  Emit an event declaring that the controlled drug Lot has been delivered
else
  Revert contract state and show an error.
if *caller* == *hospital* ∧(*lotstate* == *DeliveryEnded*) **then**
  Update *lotstate* to "Received"
  Emit an event declaring that the controlled drug lot has been received by the hospital
else
  Revert contract state and show an error.
/* The delivery process is complete */

will update the *lotstate* to "Received" to confirm that the delivery process was successful by emitting an event. This concludes the production process, and the hospital is ready to dispense, prescribe, and administer the controlled drugs to patients.

- **Consumption Algorithm**
  Algorithm 3 represents the consumption phase. In this phase, controlled drugs are dispensed, prescribed, and administered to patients while ensuring accountability and data provenance by storing the required information on an immutable ledger. Moreover, there is a variable that contains the different states of the controlled drug that is called *controlleddrugstate* which is used to inform other actors of the current state of the controlled drug. The consumption phase includes four main functions,*DrugReadyForDispensing*, *DrugPrescription*, *DrugAdministration*, and *DrugDisposal*. In the first function, the hospital will update the *availableAmount* which reflects the quantity of the available controlled drugs, update the *controlleddrugstate* to "ReadyForDispensing", and emit an event to notify the prescribers of the availability of the newly arrived controlled drug.
  In the second function, the prescriber will use the available controlled drugs to write new prescriptions to patients, which is done by adding the patient EA and prescribed amount. Moreover, the prescriber will update the *controlleddrugstate* to "Prescribed" and emit an event to notify the nursing station that the prescription is ready.
  In the third function, the nurse will handle the administration process by adding the patient EA which should match the EA in the prescription, the administered amount, and upload an image of the administered controlled drug. In addition to that, the nurse will have to update the available amount of the controlled drug, update the *controlleddrugstate* to "Administered", and finally all the important information about the administration process and logged and stored in the form of an event and emitted to all concerned actors.
  Finally, in the fourth function, the nurse will have to properly dispose any unwanted or unused controlled drugs that are dispensed and prescribed. The disposal process is done by adding the disposed amount and image of the disposed controlled drug. Moreover, the available amount is updated, the *controlleddrugstate* is changed to "Disposed", and finally all the necessary details are emitted in the form of an event to all concerned actors and the end of the disposal process is declared. This concludes the administration and disposal processes.

V. TESTING AND VALIDATION

In this section, we test and validate the Registration, Production, and Consumption smart contracts. The execution details of each function are obtained by REMIX IDE. Table 1 shows the actors with their corresponding Ethereum addresses which will be used as a reference in the process of testing and validating the solution. The main purpose of this section is to confirm that the smart contracts are deployed and their functions are executed properly. Moreover, it is necessary to inspect the output of each executed function to confirm that information is stored as expected. Figures 6 to 14 illustrate the output of the execution of each function in the proposed solution which are necessary to show that the correct addresses and events are being stored on the blockchain. Each function's logs and events are presented below.

A. REGISTRATION SMART CONTRACT FUNCTIONS

In this subsection, the execution of registration functions is illustrated. Since all registration functions work in a similar way, only manufacturer registration function will be illustrated for simplicity. Figure 6 illustrates the details of registering a manufacturer. The Ethereum Address of the actors

Algorithm 3 Consumption

Input: *patientEA*, *availableAmount*, *prescribedAmount*, *administeredAmount*, *disposedAmount*, *administereddrugImage*, *disposeddrugImage*, *controlleddrugstate*

Output: Events declaring the details of the prescription, administration, and disposal of the controlled drugs

initialization **if** *(caller == hospital)* $\wedge$ *(controlleddrugstate == NotReady)* **then**
  Update *availableAmount*
  Update *controlleddrugstate* to "ReadyForDispensing"
  Emit an event declaring that the end of the dispensing process
else
  Revert contract state and show an error.

`/* Dispensing process is complete */`

if *(caller == prescriber)* $\wedge$ *(controlleddrugstate == ReadyForDispensing)* **then**
  Add *patientEA* and *prescribedAmount*
  Update *controlleddrugstate* to "Prescribed"
  Emit an event declaring the details of the prescription
else
  Revert contract state and show an error.

if *(caller == nurse)* $\wedge$ *(controlleddrugstate == Prescribed)* $\wedge$ *(prescribedAmount* $\leq$ *availableAmount)* **then**
  Add *patientEA*, *administeredAmount*, and *administereddrugImage*
  Update *availableAmount*
  Update *controlleddrugstate* to "Administered"
  Emit an event declaring the administration of the controlled drug to patients
else
  Revert contract state and show an error.

`/* The administration process is complete */`

if *(caller == nurse)* $\wedge$ *(controlleddrugstate == Prescribed)* $\wedge$ *(disposedAmount* $\leq$ *availableAmount)* **then**
  Add *disposedAmount* and *disposeddrugImage*
  Update *availableAmount*
  Update *controlleddrugstate* to "Disposed"
  Emit an event declaring that the controlled drug has been disposed
else
  Revert contract state and show an error.

`/* The disposal process is complete */`

eligible to run this function is displayed in the "from" field which is the CDR in this case, and it should be noted that if another actor attempts to execute this function, it will fail. The "to" field shows the Ethereum address of the Registration smart contract which is where the function was executed. The "gas" field shows the total amount of used gas in the transaction, however, the private blockchain has the gas price set to zero. Therefore, the actor will not spend Ether to execute a function. But, the gas is still needed to prevent functions with very large gas amount from congesting the network. The "hash" field shows the transaction hash, which can be referred to when the details of the transaction are needed later on. Finally, the "decoded input" shows the input of the CDR which is the Ethereum address of the newly added manufacturer.

TABLE 1. The ethereum address of each actor in the testing and validation process.

	Ethereum Address
CDR	0x0eB7a13688C9f6993E1d4bDa4aBf11aDB7d780dA
Manufacturer	0x79EA7FaC951aA455Ecb6bF8809dd44Ba08a2d2FD
Distributor	0xC9b744C8e5Bf3D458623ea99076CaA7D4658c989
Hospital	0xBd1b2F1AAED3C9f229f82e69155BEc9b90E93A2c
Prescriber	0x3eFb8A5C9c953376861Ec79f4040982F9e0AD98f
Nurse	0x2B46C8C1C8f5f97211B8E5216dA422dFf644CdC8
Patient	0xcB5e9eB9Cdf5527A50101227796125EF401ef127

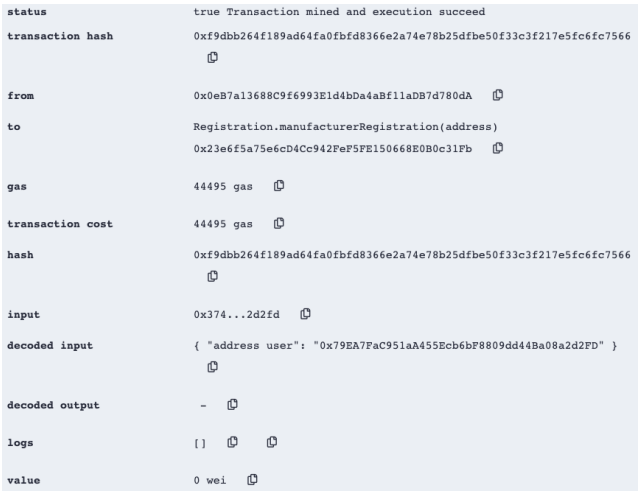


FIGURE 6. Successful execution of the manufacturerRegistration function.

B. PRODUCTION SMART CONTRACT FUNCTIONS

- **CreateControlledDrug:** An illustration of a successful attempt to run the function that creates new controlled drug is shown in Figure 7. This function basically allows the manufacturer, who is the only actor allowed to execute it, to store the details of the newly manufactured controlled drug Lot on the Ethereum network. The Ethereum address of the actor who runs the function is displayed in The "from" field which must be the manufacturer in this case. The "to" field shows the Ethereum address of the Production smart contract which is where the function is located. The "transaction cost" field shows how much gas is needed to execute the function. The "hash" field shows the transaction hash which is unique for every function execution. Finally, the "logs" field shows the information that

is stored on the Ethereum network in the form of an event. It should be noted here that the input for the fields that require a string-type input (e.g, the controlled drug name) are replaced with a bytes32 input instead because using a string-type input might cause overflow in the system since string-type inputs are unlimited. To make this process seamless in a real use case scenario, the web3.js can be used to convert the user's input from string/text to bytes32 and vice versa when information is fetched from the Ethereum network in the form of bytes32.

FIGURE 7. Successful execution of the CreateControlledDrug function.

- **StartDelivery and EndDelivery:** Figures 8 and 9 illustrate the details of the delivery process. The Ethereum address of the actor who runs the function is displayed in The "from" field which must be the distributor in this case. The "to" field represents the Ethereum address of the smart contract that the distributor interacted with which is the Production smart contract in this case. The "transaction cost" field shows the total amount of gas spent to execute the function. The "hash" shows the address that can later on be accessed to confirm the details of this specific execution of the function. Finally, the "logs" field shows the logs and events of the function.
- **LotReception:** Figure 10 illustrates the details of the last step in the delivery process, which is the reception of the controlled drug lot by the hospital. The Ethereum address of the actor who runs the function is displayed in The "from" field which must be the hospital in this case. The "to" field represents the Ethereum address of the Production smart contract in which the function is executed. The "transaction cost" field shows the total amount of gas spent to execute the function. Finally, the "logs" field represents the details of the reception process where the Ethereum address of the receiving hospital is logged and stored.

FIGURE 8. Successful execution of the StartDelivery function.

FIGURE 9. Successful execution of the EndDelivery function.

FIGURE 10. Successful execution of the LotReception function.

C. CONSUMPTION SMART CONTRACT FUNCTIONS

- **DrugReadyForDispensing:** Figure 11 illustrates the details of announcing the readiness of the controlled drugs for dispensing. The Ethereum address of the actor who runs the function is displayed in The "from" field which must be the hospital in this case. The "to" field represents the Ethereum address of the smart contract that the hospital interacted with which is the Consumption smart contract. The "transaction cost" field shows the total amount of gas spent to execute this function. The "hash" field shows a unique address that can be used to refer to this specific execution of the function. Finally, the "logs" field shows the emitted event details where the hospital Ethereum address, the controlled

FIGURE 11. Successful execution of DrugReadyForDispensing function.

drug Lots amount, and the amount of controlled drugs inside each lot are all shown and logged.

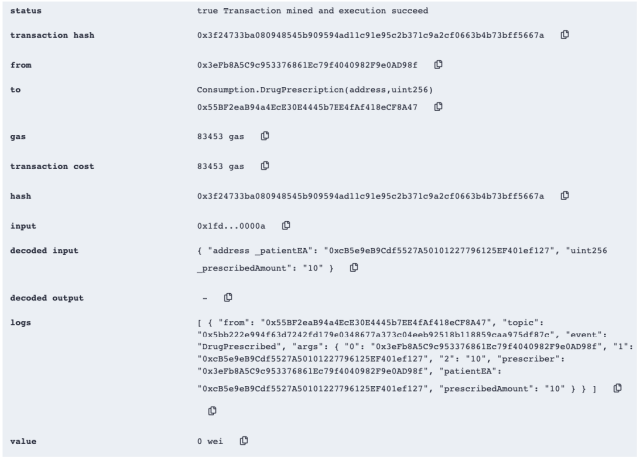


FIGURE 12. Successful execution of DrugPrescription function.

- **DrugPrescription:** Figure 12 shows the details of prescribing a controlled drug to a patient. The Ethereum address of the actor who runs the function is displayed in The ''from'' field which must be the prescriber in this case. The ''to'' field shows the Ethereum address of the Consumption smart contract which is where this function is executed. The ''transaction cost'' shows the total amount of gas used to execute the function. The ''decoded input'' shows the input of the prescriber which is the Ethereum address of the patient and the prescribed amount. Finally, the ''logs'' fields show the details of the emitted event which is logged and stored on the Ethereum network.
- **DrugAdministration:** Figure 13 shows the details of the controlled drug administration process. The Ethereum address of the actor who runs the function is displayed in The ''from'' field which must be the nurse in this case. The ''to'' field represents the Ethereum address of the smart contract that the nurse interacted with which is the Consumption smart contract. The ''transaction cost'' field shows the total amount of gas used to execute the function. The ''hash'' field shows the unique address of this specific execution of the function. The ''decoded input'' field shows what the nurse inserted as an input to the function such as the patient EA, administered amount, and the IPFS hash of the uploaded image. Finally, the ''logs'' field represents the details of the drug administration process where the Ethereum address of the patient and nurse, the administered amount, and the IPFS hash of the administered drug image are logged and stored.

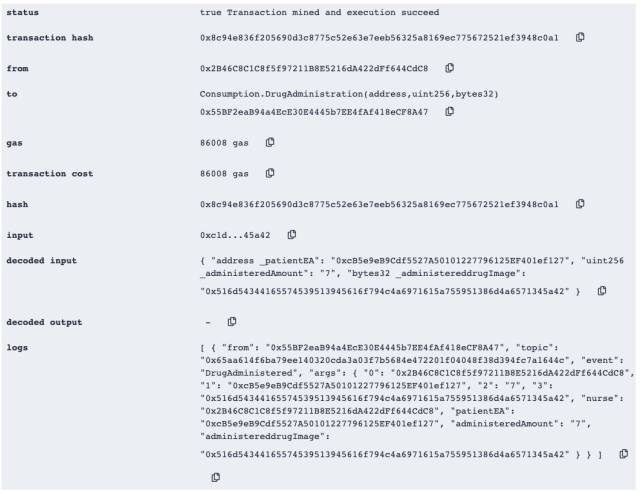


FIGURE 13. Successful execution of DrugAdministration function.

- **DrugDisposal:** Figure 14 shows the details of the disposal process of the controlled drug. The Ethereum address of the actor who runs the function is displayed in The ''from'' field which must be the nurse in this case. The ''to'' field represents the Ethereum address of the smart contract that the nurse interacted with which is the Consumption smart contract. The ''transaction cost'' field shows the total amount of gas used to execute the function. The ''decoded input'' shows what the nurse can insert as an input to the function such as the disposed amount and the IPFS hash of the uploaded image. Finally, the ''logs'' field represents the details of the drug administration process where the Ethereum address of the nurse, the disposed amount, and the IPFS hash of the disposed drug image are logged and stored.

VI. DISCUSSION

In this section, various analyses are performed on our proposed solution to evaluate its privacy and confidentiality and to estimate its security level. Moreover, a comparison of our proposed solution with existing solutions is performed. Finally, how our solution can be extended and generalized to other systems is discussed.

A. PRIVACY, CONFIDENTIALITY, AND SECURITY ANALYSIS OF THE PROPOSED SOLUTION

Although the solution is implemented using a private Ethereum blockchain, the privacy and confidentiality of the

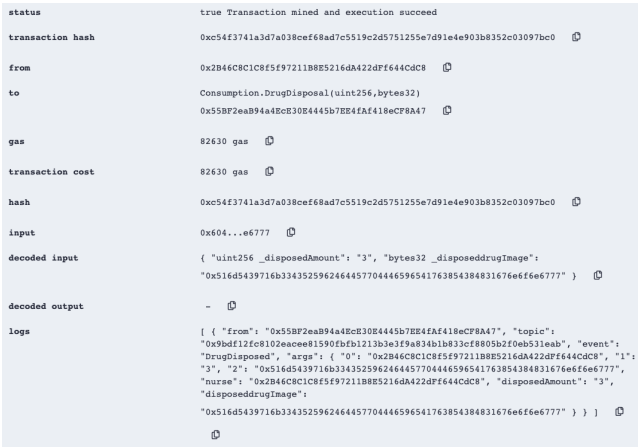


FIGURE 14. Successful execution of DrugDisposal function.

patient details must be protected. Therefore, the designed smart contracts permit the authorized actors to only add the patients Ethereum address when prescribing or administering a controlled drug to them. This ensures that any personal information about the patients are never stored on-chain.

Moreover, integrity, accountability, authorization, and availability are important aspects of the proposed solution that are analyzed to estimate the security of the proposed solution. Moreover, some common attacks to blockchain-based applications are discussed to further evaluate the security of the proposed solution.

- **Integrity:** The event-based approach used in the proposed solution for controlled medication management allows the users to track and trace controlled drugs when they are being produced and delivered because every transaction is recorded and stored on an immutable ledger.
- **Accountability and Non-repudiation:** The Ethereum smart contracts written in Solidity language allows using a feature called the ''Modifier'' which permits specific actors to run specific functions. Therefore, all actors are accountable for their actions, and any mistake or error will be permanently stored in the form of an event in an immutable ledger, and these mistakes or errors can be traced to identify their sources.
- **Availability:** Unlike the public Ethereum blockchain, a private Ethereum blockchain allows trusted nodes to join the network and do the mining, and such networks usually have a smaller number of validating nodes compared to a public Ethereum blockchain. Therefore, maintaining a high availability percentage is highly dependent on the number of validating nodes and their quality of service.
- **MITM Attacks:** In such attacks, the attacker needs to acquire the private key of the sender to steal or change the contents of a transaction and then sign it. Therefore, performing such attacks in a blockchain network are extremely difficult because it is not possible for the intruder to get the private keys of the senders unless the sender willingly does that.

B. SMART CONTRACT SECURITY ANALYSIS

Smart contracts are susceptible to some common vulnerabilities such as integer overflow and underflow, callstack depth attack, transaction-ordering dependence (TOD), reentrancy, timestamp dependency, and parity multisig bug. Therefore, security analysis is required to ensure that none of these vulnerabilities exist in our smart contracts. To achieve this, the Oyente tool is utilized. The tool is run by using a docker container to test the vulnerability of the smart contracts [24]. The results of executing the Oyente tool are presented in Figure 15 and ensures that there are no vulnerabilities in our smart contracts, therefore, the smart contracts can be considered robust and secure against such common attacks.

```
INFO:root:contract remote_contract.sol:Consumption:
INFO:symExec:   ============ Results ===========
INFO:symExec:     EVM Code Coverage:                      87.5%
INFO:symExec:     Integer Underflow:                      False
INFO:symExec:     Integer Overflow:                       False
INFO:symExec:     Parity Multisig Bug 2:                  False
INFO:symExec:     Callstack Depth Attack Vulnerability:   False
INFO:symExec:     Transaction-Ordering Dependence (TOD): False
INFO:symExec:     Timestamp Dependency:                   False
INFO:symExec:     Re-Entrancy Vulnerability:              False
INFO:symExec:   ====== Analysis Completed ======
INFO:root:contract remote_contract.sol:Production:
INFO:symExec:   ============ Results ===========
INFO:symExec:     EVM Code Coverage:                      88.4%
INFO:symExec:     Integer Underflow:                      False
INFO:symExec:     Integer Overflow:                       False
INFO:symExec:     Parity Multisig Bug 2:                  False
INFO:symExec:     Callstack Depth Attack Vulnerability:   False
INFO:symExec:     Transaction-Ordering Dependence (TOD): False
INFO:symExec:     Timestamp Dependency:                   False
INFO:symExec:     Re-Entrancy Vulnerability:              False
INFO:symExec:   ====== Analysis Completed ======
INFO:root:contract remote_contract.sol:Registration:
INFO:symExec:   ============ Results ===========
INFO:symExec:     EVM Code Coverage:                      99.9%
INFO:symExec:     Integer Underflow:                      False
INFO:symExec:     Integer Overflow:                       False
INFO:symExec:     Parity Multisig Bug 2:                  False
INFO:symExec:     Callstack Depth Attack Vulnerability:   False
INFO:symExec:     Transaction-Ordering Dependence (TOD): False
INFO:symExec:     Timestamp Dependency:                   False
INFO:symExec:     Re-Entrancy Vulnerability:              False
INFO:symExec:   ====== Analysis Completed ======
```

FIGURE 15. Smart contracts vulnerability analysis.

C. SCALABILITY ANALYSIS OF THE PROPOSED SOLUTION

According to [26], the scalability of a private Ethereum blockchain that uses PoA as its consensus algorithm is not significantly affected by the network size. This is because the way nodes are setup make them seal blocks in a pre-assigned and static time interval unlike PoW where the addition of more nodes to the network improves the overall performance of the system. Furthermore, having more nodes in the network can help in improving the availability and security of the system, however, it should be noted that there is a specific number of nodes that the system can have to maintain its optimal performance, and if this number is exceeded, nodes might get out of sync and uncle blocks might exist if the time needed for propagating information exceeds the block period. Furthermore, with the right network configuration (block period, block size, and number of nodes), a private PoA Ethereum blockchain can reach a throughput of 328 transactions per seconds which is much higher than the

15 transactions per second of the Mainnet of the Ethereum blockchain. Therefore, our proposed solution should be able to handle a large number of transactions in real time.

D. COMPARISON WITH THE EXISTING SOLUTIONS

A comparison between our solution and other non-blockchain and blockchain based solutions is presented in tables 2 and 3 respectively.

TABLE 2. Comparison with existing non-blockchain based solutions.

	Michael Gabay [5] [6] [7]	NICE guidelines [12]	PDMP [25]	Our Solution
Decentralization	No	No	No	Yes
Integrity	No	No	No	Yes
Security	No	No	No	Yes
Traceability	Yes	Yes	Yes	Yes
Accountability	No	No	No	Yes

Table 2 compares the existing non-blockchain based solutions to our solution based on critical parameters. Decentralization, which refers to storing the same data on different machines, which is the opposite of centralization, is used as a parameter to show which solution prevents a central authority from having full control over the stored data. The comparison shows that our solution is the only one that works in a decentralized manner, whereas the others are all centralized. Moreover, integrity is another important parameter that refers to how accurate the stored data is. Because the other solutions are controlled centrally, there are no guarantees that the stored data cannot be changed, whereas our solution has the data stored in multiple machines, which ensures that no central authority can manipulate the stored data. Additionally, security against common cyber attacks and vulnerabilities is another important parameter that is used to evaluate the performance of our solution compared to other solutions. The centralized nature of the other solutions make them susceptible to attacks because it is easier to attack a single point rather than attacking multiple machines which is the case of our solution. Furthermore, traceability, which refers to the ability of tracing the origin of the controlled drugs to ensure their authenticity, is a common parameter among all solutions because it is the ultimate goal of each solution. Finally, accountability, which refers to how the solution can identify the source of each error and mistake in the system, is another important parameter that needs to be implemented properly in the solution. Our solution leverages blockchain technology which ensures that every taken action is stored permanently on an immutable ledger and therefore accountability is ensure, on the other hand, the other solutions allow their actors to log their actions manually which allows manipulation and increases the likelihood of human error that is difficult to detect.

Table 3 compares our solution with the existing blockchain-based solutions. The comparison is conducted based on important parameters such as blockchain platform, mode of operation, consensus algorithm, data storage, tracking capability, throughput, and cost. Like some of the existing solutions proposed in [16] and [18], our solution has also used Ethereum network in a private mode of operation;

TABLE 3. Comparison with the existing blockchain-based solutions.

	Our Solution	SecureRx [16]	Singh R et al. [17]	P.Zhu et al. [18]	Jamil.F [21]
Blockchain Platform	Ethereum	Ethereum	Hyperledger Fabric	Ethereum	Hyperledger Fabric
Mode of Operation	Private	Private	Private	Private	Private
Consensus Algorithm	PoA	PoW	Raft	Improved PBFT	NA
Off-Chain Data Storage	Yes	No	No	No	No
tracking Capability	Yes	Yes	Yes	Yes	Yes
Throughput	High	Low	High	Low	High
Transaction Cost	None	High	None	None	None
Infrastructure Cost	Low	Low	High	Low	High

whereas, the solutions proposed in [17] and [21] have used Hyperledger Fabric. The consensus algorithm is important to showcase how scalable the solution can be because it describes how the nodes communicate with each other to reach a consensus. Our solution and [18] both used a consensus algorithm that is not the standard PoW to reduce the communication among the nodes and increase throughput. Moreover, the use of such consensus algorithms removes the need for transaction costs. On the other hand, [16] uses PoW consensus algorithm which affects the throughput negatively as the number of nodes increases. Moreover, as the network becomes more congested, transaction costs increase. The solutions proposed in [17] and [21] use Hyperledger Fabric which provides very high throughput because of its consensus algorithms. Finally, the infrastructure cost in the Ethereum-based solutions is relatively low compared to Hyperledger Fabric based solutions which requires renting blockchain servers to allow the solution to work properly at a large scale while maintaining high throughput.

E. GENERALIZATION

Our solution illustrates how private Ethereum blockchain can be utilized to fulfill the needs of a controlled medication manufacturing, delivery, administration, and disposal system. The designed smart contracts that represent the different phases of the whole process are customized for the needs of this specific application. However, other systems that have high sensitive items that require tracking, tracing, and accountability can have these smart contracts customized to their needs.

The setup of the private PoA Ethereum blockchain will remain the same and the DApps will be connected to it in a similar way. However, some systems will require their Traceable Resource Unit (TRU) to be tracked in real-time which requires the container to be added as an actor to the system so that any violation is recorded and reported in real-time [23]. Moreover, other system might have more than one distributor in a single delivery process, therefore, the delivery process function will need to be modified to accommodate the additional steps.

Figure 2 can be used to further clarify what changes are needed to customize the proposed solution to other systems. First, the actors and their interactions will most likely be different, therefore, the Registration smart contract needs to be modified to address these differences. Moreover, off-chain storage might not be needed if the application does not involve large-sized content. Finally, the developed algorithms for the designed smart contracts can be easily altered

to meet the requirements of the new system, but they can still be under the same designation.

VII. CONCLUSION

In this paper, we presented a blockchain based approach for controlled medication management. We proposed a private Ethereum blockchain-based solution that enables registering actors and producing and consuming controlled drugs in a way that is decentralized, traceable, accountable, transparent, secure, and auditable. We developed smart contracts that can replace the manual sheets by automatically recording and logging events that are related to the production and consumption of controlled drugs. The issue of limited storage space on the Ethereum network has been overcome by utilizing off-chain storage where images are uploaded and stored. We present a high-level system architecture details, system component interactions, and algorithms for the different phases of the proposed solution along with their implementation, testing, and validation details. We analyze the security, privacy, and confidentiality of the proposed solution to ensure that smart contracts are safe against common attacks and vulnerabilities and that no information related to any actor in the system is made public, and our solution showed good results in securing the system and keeping the sensitive and confidential information private, moreover, our smart contract security analysis showed that no vulnerabilities exist in our smart contract code. We compared our solution with existing solutions and it showed that it meets the requirements of a controlled medication management system, whereas other solutions were lacking in some critical areas such as privacy and security. We present how our solution can be generalized and extended to other systems that are experiencing similar issues or require similar specifications. Although our solution has managed to address major issues related to controlled medication, it still has some issues. For example, the immutability feature of blockchains can have inverse consequences because storing wrong information on-chain due to human error cannot be corrected. Another example is interoperability which requires all entities to use the same blockchain network to execute the solution smoothly. In the future, we aim to implement the full solution using Quorum platform which is a very popular private blockchain platform for applications that require high security and confidentiality.

REFERENCES

[1] M. Fan, D. Tscheng, M. Hamilton, B. Hyland, R. Reding, and P. Trbovich, "Diversion of controlled drugs in hospitals: A scoping review of contributors and safeguards," *J. Hospital Med.*, vol. 14, no. 7, pp. 419–428, 2019, doi: 10.12788/jhm.3228j.

[2] B. J. Kenny and P. M. Zito, "Controlled substance schedules," in *StatPearls [Internet]*. Treasure Island, FL, USA: StatPearls Publishing; 2021 Accessed: Jun. 9, 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK538457/

[3] D. Mark, "Coggins, PharmD, CGP, FASCP, drug diversion by health care professionals," *Today's Geriatric Med.*, vol. 9, no. 6, p. 6, 2021. Accessed: Jun. 9, 2021. [Online]. Available: https://www.todaysgeriatricmedicine.com/archive/ND16p6.shtml

[4] K. H. Berge, K. R. Dillon, K. M. Sikkink, T. K. Taylor, and W. L. Lanier, "Diversion of drugs within health care facilities, a multiple-victim crime: Patterns of diversion, scope, consequences, detection, and prevention," *Mayo Clinic Proc.*, vol. 87, no. 7, pp. 674–682, Jul. 2012.

[5] M. Gabay, "The federal controlled substances act: Schedules and pharmacy registration," *Hospital Pharmacy*, vol. 48, no. 6, pp. 473–474, Jun. 2013.

[6] M. Gabay, "Federal controlled substances act: Ordering and recordkeeping," *Hospital Pharmacy*, vol. 48, no. 11, pp. 919–921, Nov. 2013, doi: 10.1310/hpj4811-919.

[7] M. Gabay, "Federal controlled substances act: Dispensing requirements, electronic prescriptions, and fraudulent prescriptions," *Hospital Pharmacy*, vol. 49, no. 3, pp. 244–246, Mar. 2014, doi: 10.1310/hpj4903-244.

[8] K. Fleming, D. Boyle, J. Carpenter, and C. Linck, "A novel approach to monitoring the diversion of controlled substances: The role of the pharmacy compliance officer," *Hospital Pharmacy*, vol. 42, no. 3, pp. 200–209, Mar. 2007, doi: 10.1310/hpj4203-200.

[9] P. W. Brummond, D. F. Chen, W. W. Churchill, J. S. Clark, K. R. Dillon, D. Dumitru, L. Eschenbacher, T. Fera, C. R. Fortier, K. K. Gullickson, and K. Jurakovich, "ASHP guidelines on preventing diversion of controlled substances," *Amer. J. Health-Syst. Pharmacy*, vol. 74, pp. 325–348, Mar. 2017.

[10] (2020). *Pharamacist's Manual. An Informational Outline of the Controlled Substances Act*. Accessed: Jun. 8, 2021. [Online]. Available: https://www.deadiversion.usdoj.gov/pubs/manuals/index.html

[11] *U.S. Department of Justice Drug Enforcement Administration. DEA Form 222 U.S. Official Order Form for Controlled Substances*. Accessed: Jun. 8, 2021. [Online]. Available: https://tinyurl.com/3tpmpn9z

[12] (Oct. 2015). *Controlled Drugs: Safe Use and Management: NICE Guideline Short Version DRAFT*. Accessed: Jun. 9, 2021. [Online]. Available: https://www.nice.org.uk/guidance/ng46/documents/short-version-of-draft-guideline

[13] P. Kaufman. (2012). *Electronic Prescription of Controlled Substances: A Major Milestone in Healthcare it History Has Been Reached With DEA Lifting Restrictions on E-Prescribing*. Accessed: Jun. 9, 2021. [Online]. Available: https://tinyurl.com/2mkuxptf

[14] PwC. *Under Lock and Key. How Blockchain Secures Controlled Drugs*. Accessed: Jun. 8, 2021. [Online]. Available: https://www.pwc.co.uk/healthcare/pdf/pwc-blockchain-and-controlled-drugs.pdf

[15] D. Marbury, "How blockchain can reduce waste, fraud in pharmacy," *Drugs Topics J.*, vol. 163, no. 1, pp. 30–31, Jan. 2019. Accessed: Jun. 8, 2021. [Online]. Available: https://tinyurl.com/n4jthd8j

[16] M. Alnafrani and S. Acharya, "SecureRx: A blockchain-based framework for an electronic prescription system with opioids tracking," *Health Policy Technol.*, vol. 10, no. 2, Jun. 2021, Art. no. 100510, doi: 10.1016/j.hlpt.2021.100510.

[17] R. Singh, A. D. Dwivedi, and G. Srivastava, "Internet of Things based blockchain for temperature monitoring and counterfeit pharmaceutical prevention," *Sensors*, vol. 20, no. 14, p. 3951, Jul. 2020, doi: 10.3390/s20143951.

[18] P. Zhu, J. Hu, Y. Zhang, and X. Li, "A blockchain based solution for medication anti-counterfeiting and traceability," *IEEE Access*, vol. 8, pp. 184256–184272, 2020, doi: 10.1109/ACCESS.2020.3029196.

[19] M. Sahoo, S. S. Singhar, and S. S. Sahoo, "A blockchain based model to eliminate drug counterfeiting," in *Machine Learning and Information Processing* (Advances in Intelligent Systems and Computing), vol. 1101, D. Swain, P. Pattnaik, and P. Gupta, Eds. Singapore: Springer, 2020, doi: 10.1007/978-981-15-1884-3_20.

[20] M. Uddin, K. Salah, R. Jayaraman, S. Pesic, and S. Ellahham, "Blockchain for drug traceability: Architectures and open challenges," *Health Informat. J.*, vol. 27, no. 2, Apr. 2021, Art. no. 146045822110112, doi: 10.1177/14604582211011228.

[21] F. Jamil, L. Hang, K. Kim, and D. Kim, "A novel medical blockchain model for drug supply chain integrity management in a smart hospital," *Electronics*, vol. 8, p. 505, Apr. 2019, doi: 10.3390/electronics8050505.

[22] W. Gavin. (Nov. 2015). *PoA Private Chains*. Github. [Online]. Available: https://github.com/ethereum/guide/blob/master/poa.md

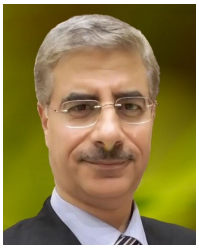
[23] P. Olsen and M. Borit, "The components of a food traceability system," *Trends Food Sci. Technol.*, vol. 77, pp. 143–149, Jul. 2018, doi: 10.1016/j.tifs.2018.05.004.

[24] R. Shigemura, G. Gonçalves, F. Oliveira, L. Coura, E. Júnior, L. Dias, A. Cunha, P. Tasinaffo, and J. Marques, "Wibx: Making smart contracts even smarter," in *Proc. Workshop Artif. Intell. Appl. Finance (WAIAF)*, 2019, pp. 1–8.

[25] (Mar. 14, 2018). *Prescription Drug Monitoring Program Training and Technical Assistance Center. History of Prescription Drug Monitoring Programs*. Accessed: Jun. 8, 2021. [Online]. Available: https://tinyurl.com/bhx7fv28
[26] M. Schäffer, M. D. Angelo, and G. Salzer, "Performance and scalability of private Ethereum blockchains," in *Business Process Management: Blockchain and Central and Eastern Europe Forum* (Lecture Notes in Business Information Processing), vol. 361, C. D. Ciccio, Ed. Cham, Switzerland: Springer, 2019, pp. 103–118.

AHMAD MUSAMIH received the B.S. degree in electrical engineering from United Arab Emirates University, United Arab Emirates, in 2015, and the M.S. degree in engineering systems and management from Khalifa University, in 2018, where he is currently pursuing 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. He is also a Research and Teaching Assistant with Khalifa University. His research interests include blockchain, healthcare, and supply chain.

RAJA JAYARAMAN received the Ph.D. degree in industrial engineering from Texas Tech University, the Master of Science degree in industrial engineering from New Mexico State University, and the bachelor's and master's degrees in mathematics from India. He is currently an Associate Professor at the Department of Industrial and Systems Engineering, Khalifa University, Abu Dhabi, United Arab Emirates. His research interests include application of blockchain technology, systems engineering and process optimization techniques to characterize, model and analyze complex systems with applications to supply chains, maintenance planning, and healthcare delivery. His postdoctoral research was 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 of supply chain data standards in the U.S. healthcare system.



KHALED SALAH (Senior Member, IEEE) received the B.S. degree in computer engineering with a minor in computer science from Iowa State University, 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, USA, in 1994 and 2000, respectively. In August 2010, he joined Khalifa University and is teaching graduate and undergraduate courses in the areas of cloud computing, computer and network security, computer networks, operating systems, and performance modeling and analysis. He is currently a Full Professor with the Department of Electrical and Computer Engineering, Khalifa University, United Arab Emirates. Prior to joining Khalifa University, he worked for ten years at the Department of Information and Computer Science, King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia. He has over 190 publications and three patents. He has been giving a number of international keynote speeches, invited talks, tutorials, and research seminars on the subjects of blockchain, the IoT, fog and cloud computing, and cybersecurity. He is a member of IEEE Blockchain Education Committee. He was a recipient of the Khalifa University Outstanding Research Award 2014/2015, the KFUPM University Excellence in Research Award of 2008/2009, and the KFUPM Best Research Project Award of 2009/2010. He was also a recipient of the Departmental Awards for Distinguished Research and Teaching in prior years. He is the Track Chair of IEEE GLOBECOM 2018 on Cloud Computing. He is an Associate Editor of *IEEE Blockchain Newsletter*. He serves on the editorial boards for many WOS-listed journals, including *IET Communications*, *IET Networks*, *JNCA* (Elsevier), SCN (Wiley), *IJNM* (Wiley), *J.UCS*, and *AJSE*.

HAYA R. HASAN received the B.S. degree in computer engineering from the American University of Sharjah, United Arab Emirates, in 2014, and the master's degree in electrical and computer engineering from Khalifa University, United Arab Emirates, in 2018. She is currently a Research Associate at the Department of Industrial and Systems Engineering, Khalifa University. She is passionate about research especially in the field of blockchain and smart contracts. She has publications in her area of interest, blockchain as well as in security.

IBRAR YAQOOB (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Malaya, Malaysia, in 2017. He is currently working with the Department of Electrical Engineering and Computer Science, Khalifa University, United Arab Emirates. Previously, he worked as a Research Professor at the Department of Computer Science and Engineering, Kyung Hee University, South Korea, where he completed his Postdoctoral Fellowship under the prestigious grant of Brain Korea 21st Century Plus. He worked as a Researcher and a Developer at the Centre for Mobile Cloud Computing Research (C4MCCR), University of Malaya. His numerous research articles are very famous and among the most downloaded in top journals. He has been listed among top researchers by Thomson Reuters (Web of Science) based on the number of citations earned in the last three years in six categories of computer science. His research interests include big data, blockchain, edge computing, mobile cloud computing, the Internet of Things, healthcare, and computer networks. He is serving/served as a guest/associate editor for various journals. He has been involved in a number of conferences and workshops in various capacities.

YOUSOF AL-HAMMADI received the bachelor's degree in computer engineering from the Khalifa University of Science and Technology (previously known as the Etisalat College of Engineering), Abu Dhabi, United Arab Emirates, in 2000, the M.Sc. degree in telecommunications engineering from the University of Melbourne, Australia, in 2003, and the Ph.D. degree in computer science and information technology from the University of Nottingham, U.K., in 2009. He is currently the Acting Dean of Graduate Studies and an Assistant Professor with the Electrical and Computer Engineering Department, Khalifa University of Science and Technology. His main research interests include the area of information security which include intrusion detection, botnet/bots detection, viruses/worms detection, machine learning and artificial intelligence, RFID security, and mobile security.