Securing the Healthcare Supply Chain Using Blockchain-Enabled Smart Contracts

Muhammad Saad[1], Hamail Ashraf[2], Muhammad Saffi Ullah Khan[3],
Muhammad Awais Javed[4], Ahmad Naseem Alvi[5], Ahmed Alfakeeh[6]*
Department of Electrical Engineering, COMSATS University Islamabad, 45550, Islamabad, Pakistan[1,2,3,4,5]
Department of Information Systems, King Abdul Aziz University, Jeddah, Saudi Arabia[6]

***Abstract*—The blockchain technology is an innovative tool that has shown its effectiveness in many sectors such as health-care and many are likely to experience a revolutionary transformation. In healthcare, the blockchain functions like a distributed network which is constantly updated with records ensuring that you cannot eliminate or alter them without the consensus to do so. In other words, blockchain-based smart contracts will enable clients to do transactions without having to rely on intermediaries and thus it will be more reliable. Smart contracts regulate and constitute numerous activity and dealings of stakeholders, automating processes, augmenting visibility, maximizing productivity, and standing against the clock. Healthcare supply chains may be exploited by blockchain technology that addresses pain points such as connectivity, traceability, as well as fighting counterfeit medicines. The purpose of our analysis was to advance traceability between the healthcare supply chain elements, such as the transportation of pharmaceuticals or medical equipment. And for its more practical implementation we have worked on minimizing the cost of smart contracts being deployed on the blockchain for the health care logistic system.**

***Keywords*—Blockchain; smart contracts; supply chain**

I. Introduction

With the advancement in technology and everything being digitalized, the supply chain management has become difficult to run by the old methods. With the beginning of a new era and digitalized market, it has become difficult to keep record of the supply chain system. From keeping record of raw material, manufacturing, transportation of goods to the market and keeping record of whether the items are damaged, expired, delivered, or lost, the need for a digitalized and secure system has emerged. Many difficulties are faced during transportation of goods like food, medicines, machineries, electronics, and almost all the items that can be damaged during long-distance transportation. This leads to complexity and ambiguity between the buyer and the supplier. So, the overall system becomes inefficient. COVID-19 is a major example [1], [2], [3] of why we need to move on to a new digital supply chain system that can provide more efficiency and transparency in the whole supply chain market [4], [5], [6].

We can exclude these problems and make a more transparent system by shifting towards the smart contract between the supplier and the buyer. These smart contracts will not just provide transparency but also will execute automatically, eliminating the middleman. Smart contracts are capable of keeping records safe on a blockchain system. These smart contracts can help in eliminating the constraints between the customer and the manufacturer. Records are saved in a ledger in a blockchain consisting of secured blocks that can eventually reduce the payment issues, increase the efficiency of the supply chain system, and reduce the problems being faced during transportation in the healthcare system. All the data will be stored on the supply chain in different blocks, which will also eliminate the risk of data being on a centralized unit [7], [8], [9], [10].

Major problems in a healthcare supply chain system are keeping track record of in-stock items, shortage of items, expiry of medicines, and managing the required temperature of the medicine. These problems must be dealt with; otherwise, the system becomes inefficient. For this, open-source blockchains are used in which smart contracts are translated and then deployed. Transaction of any agreement will only be done once both parties have fulfilled their respective jobs. This reduces the ambiguity in the supply chain of the healthcare system. For instance, if a customer receives any damaged or expired medicine, then his transaction will not be completed unless or until the agreements of the smart contracts are fulfilled.

After moving towards a digital smart contract system which is secure and reliable, it is important to keep account of the cost of using a smart contract-enabled supply chain system. For that, work on reducing the gas amount is necessary to make it feasible to be deployed in the healthcare supply chain system.

Concluding this, a smart contract-enabled healthcare supply chain system is necessary to fill the flaws in the old supply chain system, such as keeping the record safe, ensuring efficiency, having a decentralized system, and having transparency among the buyer and the manufacturer by eliminating the middleman from this whole process.

A. What is Blockchain?

The term *blockchain* was first introduced in 1991 by Stuart Haber and W. Scott Stornetta, describing the concept of linked blocks secured by cryptographic codes. Bitcoin was the first real-world application to implement blockchain technology, with its public version 1.0 launched in 2009. Later, the second generation of blockchain, referred to as version 2.0, emerged around 2013–2014 and gained practical use with the launch of Ethereum and its smart contracts in 2015. More recently, industries have adopted version 3.0 of blockchain, aiming to develop custom solutions suited to their own needs.

*Corresponding authors.

Blockchain is essentially a decentralized platform that enables secure and transparent sharing of information. It operates as a distributed digital ledger—a modern equivalent of traditional account books—where transactions are recorded. This ledger is maintained across a peer-to-peer network, making the data nearly tamper-proof. Each transaction requires digital signatures for authentication, thus minimizing the risk of fraud. The decentralized and immutable nature of blockchain makes the stored data highly secure and reliable.

B. Blocks in Blockchain

Blockchain is a decentralized and distributed digital ledger that records transactions across multiple systems, making them secure and transparent. At the core of this technology are *blocks*, which are digital containers used to store data. Each block includes a list of transactions, a timestamp, and a unique cryptographic hash as well as the hash of the previous block.

These blocks are connected sequentially, forming a chain—hence the name *blockchain*. The security of blockchain arises from the interlinking of blocks through cryptographic hashes, which ensures that each block references its predecessor. When a new transaction occurs, it is grouped with others into a block. Once the block is full, it is added to the blockchain and linked to the previous block using a hash function.

The hashing process uses SHA-256 (Secure Hash Algorithm 256-bit), a cryptographic function that generates a fixed-length, 256-bit signature from input data. This hash acts as a digital fingerprint, providing a unique identifier for each block. Any modification in a block will change its hash, thus invalidating the chain if the tampering is not corrected throughout.

This immutable nature of hash chaining ensures that data cannot be altered once added, making blockchain systems tamper-resistant. During the formation of a block, the system computes the hash of the block's data. This hash not only acts as the block's unique ID but also serves as a link to its predecessor.

Miners use SHA-256 to construct a proof-of-work, solving a computational puzzle to find a valid random integer (nonce). This process adds security and consensus to the network. Additionally, users interact with the blockchain using public and private cryptographic keys, ensuring secure communication and transaction validation.

C. Types of Blockchain

1) Public blockchain: Public blockchains are also called permission-less blockchains. Anyone can participate in the validation of transactions and data. Public blockchains are suitable for networks that have strict transparency requirements. Users can view both recent and historical records and can participate in mining. Bitcoin and Ethereum are examples of public blockchains.

2) Private blockchain: Private blockchains are used within specific organizations for the validation of transactions and data. They are more private and secure than public blockchains. Although they maintain a peer-to-peer connection and decentralized structure, they operate on a smaller scale. Only authorized parties can access this blockchain and participate in the network. Hyperledger and Corda are examples of private blockchains.

3) Hybrid blockchain: Hybrid blockchains combine characteristics of both public and private blockchains. Businesses that require both transparency and privacy often prefer this model. By using hybrid blockchains, organizations can establish both public and private systems, control which blockchain data is publicly accessible, and define who has access to specific data.

4) Consortium blockchain: A consortium blockchain is a permissioned blockchain where access decisions are controlled by a pre-selected group of nodes. It is commonly used by groups of organizations that aim to collaborate and share data in a decentralized manner while retaining a level of control over the network. Participants in a consortium blockchain typically know and trust each other. This model is well-suited for industries where multiple organizations need to collaborate while preserving data privacy and integrity.

D. Hashing Algorithm

A cryptographic security algorithm known as Secure Hash Algorithm 256-bit (SHA-256) is used in blockchain systems. This algorithm is secure and computationally difficult to reverse. SHA-256 generates hash values that are unique and irreversible, and each block in the blockchain is assigned such a hash value. Being a 256-bit value, it ensures that no two different blocks can have the same hash. Blockchain encryption is achieved by applying this algorithm, which involves a series of mathematical operations that transform input data into a fixed-length encrypted output. This process guarantees the integrity and security of the data within the blockchain.

E. Smart Contract and Use Cases

Smart contracts are digital agreements that can automatically execute predefined "if/else" conditions. These contracts are deployed on the blockchain and execute automatically when the encoded terms, defined by the participating parties, are met. Smart contracts enhance transparency and reliability by eliminating the need for intermediaries. Once a transaction is completed, it is irreversible. Depending on the type of blockchain, only authorized individuals may access the outcome of a transaction.

In supply chain systems, for example, a transaction is finalized only when the receiver confirms the items were delivered as specified in the smart contract. Otherwise, the transaction is halted until all conditions are satisfied. This enhances supply chain efficiency and reduces risk. Additionally, costs are minimized as smart contracts automate processes without requiring middlemen.

Smart contracts have several real-world applications:

- Financial services: Smart contracts enable decentralized finance (DeFi), automating trading and asset management without intermediaries. They also facilitate automated payments for organizations.
- Supply chain management: They track goods from raw material collection to delivery, ensuring quality control throughout the transportation process.

- Voting systems: Smart contracts help secure election processes, enhancing integrity and preventing manipulation.
- Social media and copyright: Content ownership and copyright enforcement can be managed through smart contracts to prevent unauthorized use of data.

F. Problem Statement

Traditional healthcare supply chain systems face challenges related to inefficiency and complexity. Tracking goods during transportation has become increasingly difficult. Common issues include maintaining inventory records, item shortages, medicine expiration, and temperature control requirements.

Ambiguity in manual record-keeping, corruption, and breaches of supply chain agreements have led to distrust between suppliers and customers. This highlights the need for a secure and digital smart contract-enabled system. Although smart contracts are partially deployed in this sector, their high deployment cost has hindered widespread adoption.

While these systems reduce overall service costs by eliminating intermediaries, the cost of deploying smart contracts on blockchain platforms remains high. Therefore, optimizing and reducing the gas fees associated with smart contract execution is essential for expanding this technology in the global healthcare supply chain.

II. Literature Review

In [11], the authors proposed a framework for Personal Health Records (PHR), highlighting the issue that a patient's personal health data is highly confidential and may be exposed to hospitals in emergencies or uncertain situations. To ensure security, the author introduced an emergency healthcare system that manages PHR using blockchain-enabled smart contracts with permissioned data access.

In [12], the authors proposed a mobile healthcare solution utilizing smart contracts on blockchain. This system facilitates rapid action in emergencies where a doctor must urgently assess a patient. Patients can transmit real-time data such as blood pressure or oxygen levels via IoT devices, and doctors can immediately respond. The entire process is executed through smart contracts on the blockchain, enhancing both security and speed.

In [13], the authors addressed ambiguities in the supply chain system, particularly within the cyber and agricultural sectors. The study proposed a blockchain-based solution to address inefficiencies and enhance traceability and transparency.

In [14], the researchers focused on the supply and demand issues of COVID-19 medical equipment. The study highlighted flaws in traditional, centralized supply chain systems, such as their vulnerability to failure and lack of reliability. To overcome these, a blockchain-enabled supply chain model was proposed, offering decentralization, enhanced security, and improved trust among stakeholders.

In [15], the researchers reviewed over 100 published articles on Electronic Health Records (EHR). They compared the security, cost, and privacy of traditional supply chain systems versus blockchain-based systems, demonstrating the superiority of blockchain in terms of data integrity and privacy protection.

In [16], a solution for an Internet-based healthcare system was presented. The system allows patient appointments to be managed remotely using a blockchain-integrated app. Only authorized users or patients can access the data, ensuring privacy and preserving the integrity of health records.

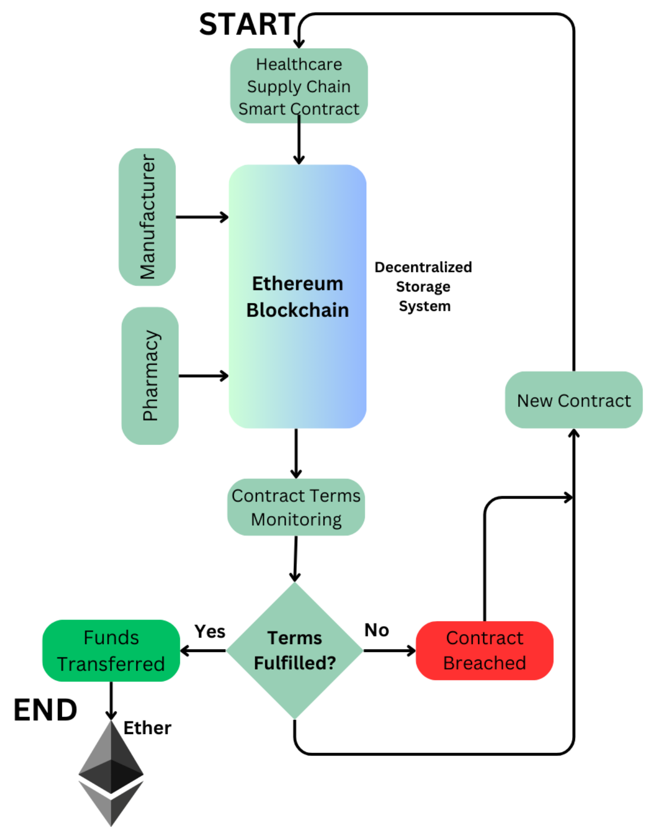

Fig. 1. Blockchain-based healthcare supply chain.

III. Implementation

The Blockchain-based healthcare supply chain is shown in Fig. 1.

A. Working of Smart Contracts

The terms of a smart contract must first be agreed upon by the participating parties. These terms are then converted into programming code, typically consisting of multiple conditional (if-then) statements. This code is stored on the blockchain network and replicated across all participating nodes. Each node executes the code, and once all network participants confirm that the predefined conditions are met, the transaction is automatically executed.

A smart contract is self-executing and includes the following basic elements:

1) Offer: The contract is initiated using Solidity with predefined conditions. The party initiating the contract provides the offer.

2) Negotiation: Upon deployment to the blockchain, the contract becomes visible to all involved parties. They may verify or negotiate the terms before finalization.

3) Approval: Once all parties agree, the trigger event (e.g., deadline, condition met) causes the contract to be signed and rendered immutable.

4) Self-satisfying conditions: Smart contracts can autonomously verify conditions using real-time data from oracles or IoT devices such as sensors and cameras.

B. Software Used

1) Solidity: A high-level programming language designed by Ethereum specifically for developing smart contracts. It compiles into machine-level code executed by the Ethereum Virtual Machine (EVM). Solidity supports variables such as boolean, integers, arrays, and modifiers, and is influenced by Python, C++, and JavaScript. It enables development of various contract types including voting, auctions, and multi-signature wallets.

2) Remix IDE: A browser-based Integrated Development Environment (IDE) for writing, compiling, and debugging Solidity smart contracts. It includes syntax highlighting, testing tools, and a built-in Solidity compiler for deploying contracts directly to the Ethereum network.

3) MetaMask: A widely used non-custodial digital wallet and browser extension that allows users to store, manage, and transact cryptocurrencies like Ether. It provides a secure, user-friendly interface for interacting with blockchain applications without requiring repeated key entries.

4) Ethereum: An open-source, decentralized platform for building and deploying smart contracts. Ether (ETH) is its native cryptocurrency, used to pay transaction (gas) fees and incentivize miners. Ethereum ensures tamper-proof, transparent, and fraud-resistant operations.

5) Ethereum Virtual Machine (EVM): The runtime environment for executing Ethereum smart contracts. EVM provides isolation between contracts, guards against denial-of-service attacks, and supports secure and traceable transactions across the decentralized network.

C. Decentralized Storage System

A decentralized storage system does not rely on a central server but instead distributes data across multiple nodes in a blockchain-enabled network. This system supports censorship resistance and immutability, allowing users to store, read, and share files in a peer-to-peer manner.

The InterPlanetary File System (IPFS) is one such decentralized storage network. It enables the publication of files, directories, and websites using cryptographic hashes stored on the blockchain. IPFS enhances security, privacy, and scalability and is often used to host web content and non-fungible tokens (NFTs).

D. Healthcare Supply Chain

The healthcare supply chain encompasses the manufacturing, distribution, and patient delivery of medical supplies and pharmaceuticals. This system must remain functional even during emergencies such as pandemics or natural disasters [17]. Blockchain enables fully digital, smart contract-based operations between healthcare providers, pharmacies, and insurers, reducing discrepancies in contracts and disputes over claims.

Key challenges include tracking inventory, expiration, shortages, and ensuring cold-chain integrity. Blockchain addresses these through immutable records that enhance transparency, prevent tampering, and authenticate pharmaceutical provenance. Each product batch can be assigned a unique identity recorded on the blockchain, allowing stakeholders to trace its journey from manufacturer to pharmacy.

Moreover, blockchain automates essential processes like order management, inventory tracking, and payments using smart contracts, reducing manual errors and enhancing efficiency. Encryption, consensus mechanisms, and access controls ensure data security and accuracy.

1) Gas cost optimization: Gas costs in smart contracts—particularly in healthcare applications—can be optimized by improving loop efficiency and minimizing unnecessary storage lookups. Direct access to storage arrays within loops significantly reduces execution costs, making smart contract deployment more feasible for large-scale healthcare systems.

E. Functions

The following are explanations of the key output functions referenced in Fig. 2 of the Healthcare Supply Chain Smart Contract:

- addDrugslot(uint256 _lot, uint256 _unitPrice): Adds a batch of drugs to the inventory for sale. Marks each medication in the lot as “For Sale” with a set price, and emits an event to notify users.
- buyDrugslot(uint256 _lot): Allows users to purchase a drug lot. Updates the buyer’s address and changes the status of each item in the lot to “Sold”, then emits a confirmation event.
- ConvertTemp(uint256 fahrenheit, uint256 celsius, uint256 _drugBatch): Converts temperatures between Fahrenheit and Celsius. Updates environmental history for each product and emits a conversion event.
- setSunSensitiveLot(uint256 _lot, bool isSunSensitive): Flags a drug lot as sunlight-sensitive using a Boolean input.
- setMedicineIngredient(string memory medicineName, string memory activeIngredient): Assigns the active ingredient to the specified medicine.
- dateformatof(uint256 _lot, uint256 _MFG_date, uint256 _EXP_date): Sets manufacturing and expiry dates for a drug lot and emits a related event.

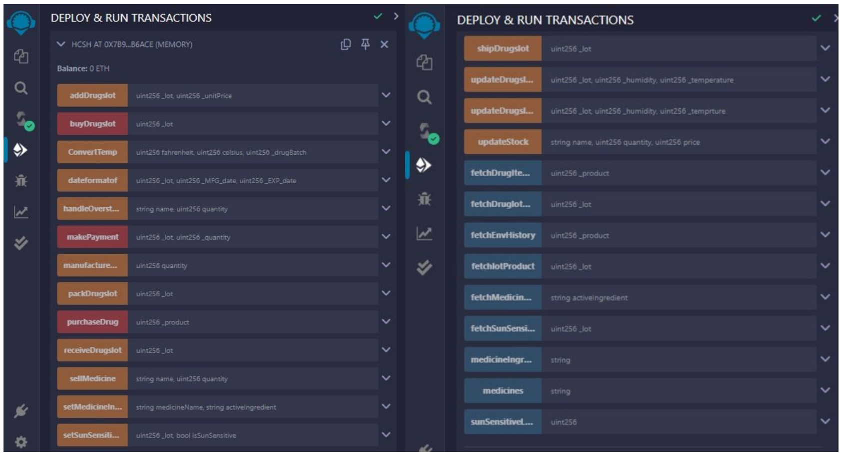

Fig. 2. Output functions.

- handleOverstock(string memory name, uint256 quantity): Manages surplus stock by flagging overstock conditions and emitting a notification event.
- makePayment(uint256 _lot, uint256 _quantity): Finalizes payment for a drug lot. Validates total price and updates status to "Purchased" upon success.
- manufactureDrugslot(uint256 quantity): Used to manufacture a drug batch. Assigns unique product IDs, updates statuses, and emits a manufacturing event.
- packDrugslot(uint256 _lot): Packages a drug lot. Confirms manufacturer identity and changes status to "Packed", recording the timestamp.
- purchaseDrug(uint256 _product): Purchases a specific drug product. Updates its status to "Purchased" and logs an event.
- receiveDrugslot(uint256 _lot): Marks a shipped lot as "Received", checks for environmental conditions, and emits a receipt confirmation.
- sellMedicine(string memory name, uint256 quantity): Sells a specified quantity of medicine. Triggers a stockout event if quantity drops to zero.
- shipDrugslot(uint256 _lot): Marks a sold lot as "Shipped" and updates the statuses of all items in the lot. Emits a shipment event.
- updateDrugslotShippmentEnv(uint256 _lot, uint256 _humidity, uint256 _temperature): Updates the environmental shipping conditions for a drug lot. Logs history and emits an event.
- updateDrugslotStockEnv(uint256 _lot, uint256 _humidity, uint256 _temperature): Modifies storage environment for a received lot. Only applicable if the drugs haven't been purchased yet.
- updateStock(string memory name, uint256 quantity, uint256 price): Updates a medicine's inventory and price in the pharmacy database.
- fetchDrugItemData(uint256 _product): View function returning product data such as ownership, manufacturer info, and price.
- fetchDruglotData(uint256 _lot): Retrieves data for a drug lot including sample item info, current owner, product count, and cost.
- fetchEnvHistory(uint256 _product): Returns environmental update history such as humidity and temperature records for a product.
- fetchlotProduct(uint256 _lot): Returns an array of product IDs associated with a given drug lot.
- fetchSunSensitive(uint256 _lot): Returns a Boolean indicating whether a lot is sunlight-sensitive.
- fetchMedicineIngredient(string memory activeIngredient): Returns the name of the medicine associated with the given active ingredient.

F. Contribution to the UN Sustainable Development Goals (SDGs)

Smart contracts contribute to achieving the United Nations Sustainable Development Goals (SDGs) in several ways:

- Goal 3 – Good Health and Well-Being: They enhance healthcare access, reduce costs, and improve quality by automating processes and reducing errors.
- Goal 9 – Industry, Innovation, and Infrastructure: Blockchain-based smart contracts promote investment in innovative healthcare systems and decentralized digital infrastructure.
- Goal 16 – Peace, Justice, and Strong Institutions: By enabling transparency and accountability, smart contracts help reduce corruption in healthcare supply chains.
- Goal 17 – Partnerships for the Goals: Through secure and verifiable data sharing, blockchain fosters stakeholder collaboration for sustainable and lasting development.

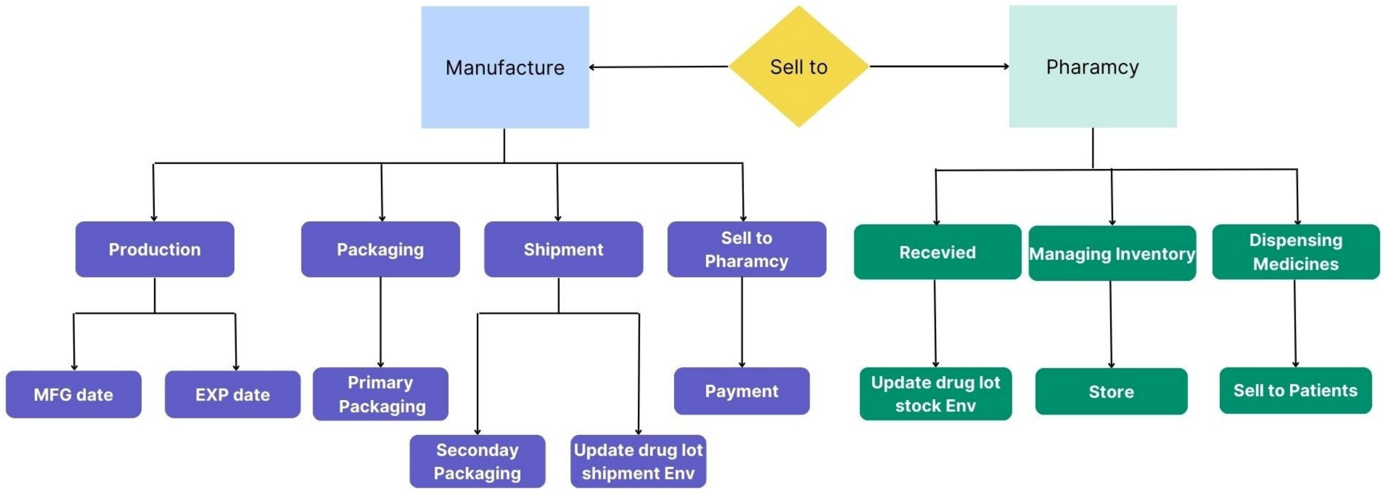

Fig. 3. The flow chart of the healthcare supply chain with all stake holders and functions.

Smart contracts and blockchain collectively offer transformative tools to support multiple SDGs across sectors, with particularly strong applications in healthcare. The flow chart of the healthcare supply chain with all stake holders and functions is shown in Fig. 3.

G. Stakeholders

Stakeholders in the healthcare supply chain include a diverse group of individuals and organizations, such as patients, physicians, nurses, pharmaceutical companies, hospitals, insurance providers, and suppliers. Each group has unique interests, requiring a well-integrated and coordinated approach.

A complex network of entities ensures the continuous flow of medical supplies and services. These collaborative networks are essential for maintaining the efficiency and quality of healthcare delivery. Efficient healthcare supply chain management reduces costs and enhances operational performance by optimizing inventory control and procurement.

Through better coordination and traceability, blockchain technology ensures high-quality patient care, prepares systems for emergencies (e.g., pandemics), and strengthens collaboration among upstream and downstream partners. It also enhances accountability, data security, and transparency in the flow of goods across the supply chain.

H. Manufacturer

Manufacturers are a vital component of the healthcare supply chain, responsible for producing and distributing medical equipment and devices. Blockchain technology offers manufacturers a powerful tool to enhance transparency, trust, and operational reliability across their business processes.

An efficient healthcare supply chain ensures the tracking and control of goods from production to wholesalers—and in some cases, directly to pharmacies or hospitals—ensuring timely and secure delivery.

1) Product traceability: Blockchain enables end-to-end product tracking, allowing manufacturers to trace items throughout the entire supply chain. This ensures product authenticity and aids in efficient recall management if necessary.

2) Supply chain efficiency: By integrating blockchain with industrial operations, manufacturers can significantly enhance process efficiency. Smart contracts can automate tasks such as payment settlements, inventory tracking, and order processing, reducing manual intervention and errors.

3) Quality control and compliance: Manufacturers can maintain immutable records of certifications, audit results, and quality control procedures on the blockchain. This not only ensures regulatory compliance but also establishes product credibility and builds customer trust.

4) Supplier management: Manufacturers rely on various suppliers for raw materials and components. Blockchain offers a secure and transparent platform for supplier verification, performance tracking, and compliance monitoring, improving supplier relationships and reducing operational risk.

I. Functions

The following are explanations of the output functions presented in Fig. 4 of the Manufacturer Smart Contract:

- assignMeAsManufacturer(): This function assigns the caller's address to the list of manufacturers using the internal function _addManufacturer. It provides the caller with the credentials required to perform manufacturer-specific tasks in the contract.
- renounceMeFromManufacturer(): This function removes the caller's address from the list of authorized manufacturers using the internal function _removeManufacturer, effectively revoking their manufacturer privileges.
- _addManufacturer(address account): An internal function that adds the specified account to the manufacturer list. Upon successful addition, it emits the

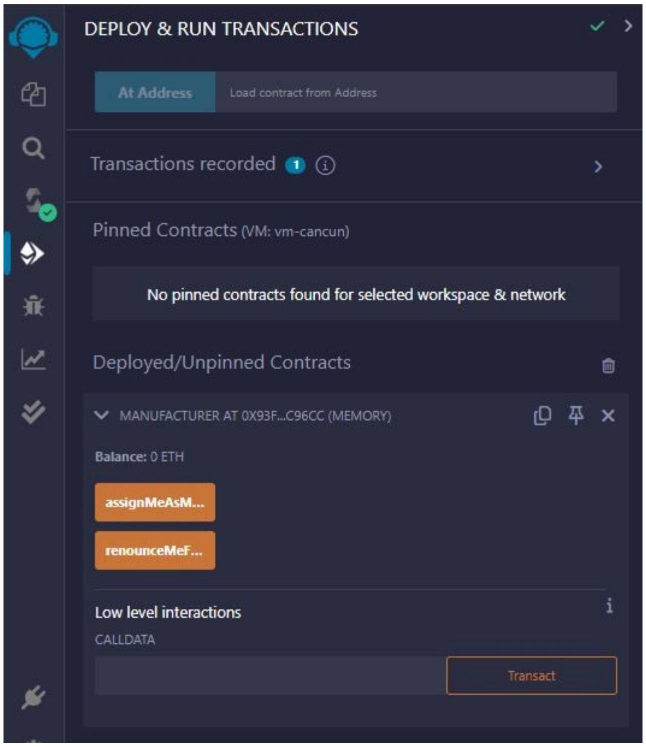

Fig. 4. Manufacturer output functions.

`ManufacturerAdded` event. This is typically used when a new manufacturer is being onboarded.

- _removeManufacturer(address account): This internal function removes the specified account from the list of manufacturers. It emits the `ManufacturerRemoved` event to confirm the action. This is typically used to revoke manufacturer access either voluntarily or administratively.

J. Pharmacy

Pharmacies form the second tier in the healthcare supply chain. They purchase pharmaceuticals and medical supplies in bulk from manufacturers. Due to high competition among pharmaceutical companies, pharmacies often have limited bargaining power and rely on wholesalers for the supply of prescription drugs to their outlets. Their functions are given in Fig. 5.

1) Medication procurement and inventory management: Pharmacies are responsible for sourcing pharmaceuticals from suppliers and maintaining sufficient inventory. Their role is crucial in avoiding shortages, minimizing stockouts, and ensuring timely access to essential medications.

2) Quality assurance and regulatory compliance: Pharmacies follow strict quality control protocols to ensure that medications remain effective and safe. This includes the handling of controlled substances, proper storage, and temperature regulation in line with regulatory standards.

3) Contribution to public health initiatives: Pharmacies play a key role in public health efforts related to drug therapy, testing, and disease screening. As accessible healthcare

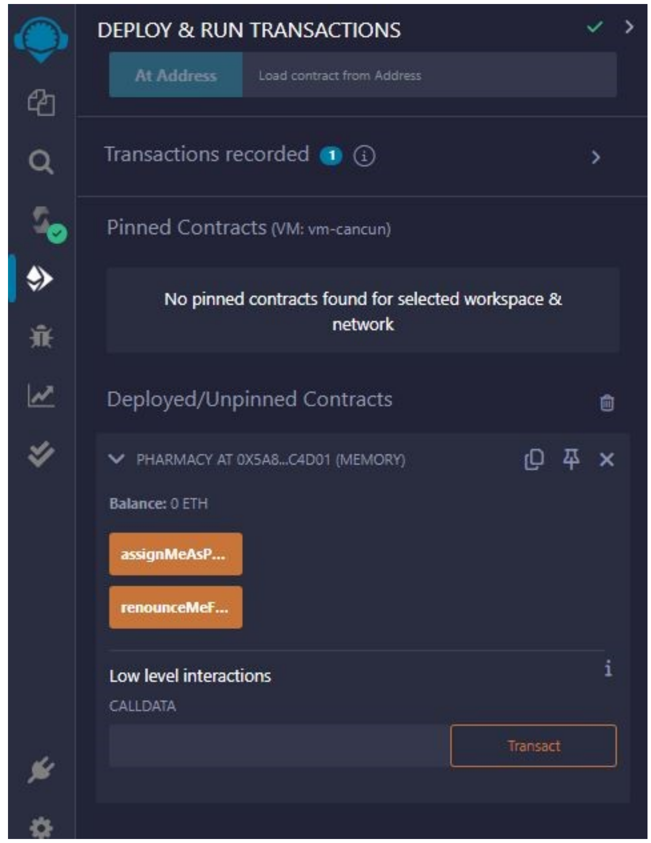

Fig. 5. Pharmacy output functions.

providers, they support disease prevention and chronic illness management at the community level.

K. Functions

The following are explanations of the output functions presented in Fig. 5 of the Pharmacy Smart Contract:

- assignMeAsPharmacy(): This public function allows any account to designate itself as a pharmacy. It uses `msg.sender` as the account reference and calls the internal function `_addPharmacy` to register the sender. This enables seamless self-assignment of the pharmacy role within the contract.
- renounceMeFromPharmacy(): This public function allows a pharmacy account to relinquish its role. It invokes the internal function `_removePharmacy` using `msg.sender` to remove the sender from the list of authorized pharmacies. It allows voluntary withdrawal from the pharmacy system.
- _addPharmacy(address account): An internal function that registers the provided address as an authorized pharmacy. It emits the `PharmacyAdded` event with the account address, and updates the contract state by granting pharmacy rights to that account.
- _removePharmacy(address account): An internal function that revokes pharmacy status for a given address. It emits the `PharmacyRemoved` event, signaling

that the specified account has been removed from the list of authorized pharmacies and no longer holds related privileges.

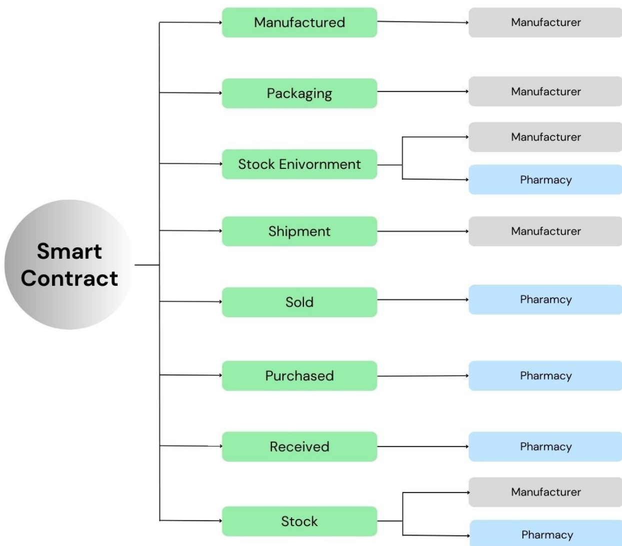

Fig. 6. System overview of blockchain based contract.

Fig. 6 shows that there are three distinct smart contracts, each having a different execution and transition cost depending on the degree of code complexity. The flowchart illustrates how registration contracts are utilized by two stakeholders: Manufacturing and Drugstore. Each stakeholder's address is recorded, and the system ensures that no duplicate addresses are used during registration. Once registration is successfully completed, the manufacturing process is initiated as the next phase.

IV. Results and Discussion

A. Cost Evaluation

Currently, there is no unified platform that enables stakeholders to exchange and communicate data at low cost. Ethereum's native cryptocurrency, ether (ETH), is used to pay for gas fees. Gas prices are expressed in gwei, which is a subunit of ETH:

- 1 ether (ETH) = 1,000,000,000 gwei (10^9 gwei)
- 1 gwei = 0.000000001 ETH (10^{-9} ETH)
- 1 ETH = 10^{18} wei (smallest unit)

The value of ETH, and thus gas cost in USD, fluctuates with market dynamics. Stakeholders must account for these costs when deploying or executing smart contracts.

B. Gas Fee

To execute any compiled smart contract on the Ethereum blockchain, a transaction fee—termed as gas—must be paid. The Ethereum Virtual Machine (EVM), responsible for processing smart contracts, consumes gas based on the computational complexity of the transaction.

Gas acts as a safeguard to prevent the network from being overloaded with overly complex or infinite-loop contracts. The more operations and storage a contract uses, the more gas it consumes.

The cost of manufacturer contract is based on the gas used, transaction cost, and execution cost, as described in Table I.

TABLE I. Cost of Manufacturer Contract

Function Name	Gas Used	Transaction Cost	Execution Cost
assignMeAsManufacturer()	25731	22374 gas	1310 gas
renounceMeFromManufacturer()	25756	22396 gas	1332 gas

Gas Cost Calculation:

$$\text{Gas Cost} = \text{Gas Used} \times \text{Gas Price}$$

The gas price depends on network congestion and user willingness to pay for faster processing.

C. Execution Cost

Execution cost is the amount of gas consumed to run the transaction or smart contract logic on the Ethereum network. This cost varies with the contract's complexity and the amount of data processed. Efficient coding practices can help reduce execution cost, which is important given that Ethereum users must pay these costs with every contract interaction.

D. Transaction Cost

The transaction cost is equivalent to the gas fee, which represents the amount of gas used to execute a transaction on the blockchain network. A transaction is required to carry out all operations, such as reading or writing data. The gas price—determined by network congestion and user willingness to pay—is multiplied by the gas consumed to calculate the total cost. Higher gas prices during peak periods can significantly affect the total cost of blockchain interaction, especially when executing complex smart contract functions.

The transaction costs for each function required to register a Pharmacy Smart Contract are displayed in Table II.

TABLE II. Cost of Pharmacy Contract

Function Name	Gas Used	Transaction Cost	Execution Cost
assignMeAsPharmacy()	25756	22396 gas	1332 gas
renounceMeFromPharmacy()	25731	22374 gas	1310 gas

Table III lists the execution costs associated with each core function in the Healthcare Supply Chain Smart Contract. These costs vary with function complexity and data manipulation.

E. Published Smart Contract Comparison

Table IV presents a gas cost comparison between our contract titled "Blockchain-enabled Smart Contracts for Logistic Control" and the published contract "Making Drug Supply Chain Secure, Traceable and Efficient: A Blockchain and Smart Contract Based Implementation" [18]. The comparison

TABLE III. COST OF HEALTHCARE SUPPLY CHAIN CONTRACT

Function Name	Gas Used	Transaction Cost	Execution Cost
addDrugslot	57879	25193 gas	2849 gas
buyDrugslot	55737	23925 gas	2721 gas
dateformatof	74768	24520 gas	3036 gas
manufactureDrugslot	203518	3983828 gas	3962624 gas
packDrugslot	58309	23907 gas	2703 gas
purchaseDruglot	32043	24010 gas	2806 gas
receiveDrugslot	36052	23907 gas	2703 gas
shipDrugslot	36090	23885 gas	2681 gas
updateDrugslotShippmentEnv	65307	24554 gas	3058 gas
updateDrugslotStockEnv	125444	24498 gas	3014 gas
handleOverstock	56275	48934 gas	27286 gas
sellMedicine	32767	25596 gas	3972 gas
updateStock	78117	67927 gas	46163 gas
setMedicineIngredient	53768	46754 gas	24826 gas
setSunSensitiveLot	53440	46469 gas	25125 gas

TABLE IV. GAS COSTS OF PUBLISHED CONTRACT IN ETH

Functionality	Our Contract	Published Contract	Published Contract (2024 Price)
Contract Deployment	0.021727111 ETH	0.002937147 ETH	0.007214388 ETH
Create Pharmacy	0.000141805 ETH	0.000443929 ETH	0.001090404 ETH
Create Manufacturer	0.000141667 ETH	0.000443862 ETH	0.001090239 ETH
Manufacture to Add Medicine	0.000555173 ETH	0.000860005 ETH	0.002112394 ETH
Pharmacy to Buy Medicine	0.000555503 ETH	0.000637359 ETH	0.001565518 ETH

includes real-time gas costs using ETH prices at two different time points: November 2022 and April 2024.

Note: The Ether price during the published contract (November 2022) was $2.2415, while in April 2024 it was $5.55057. This fluctuation directly affects cost-per-transaction, highlighting the need for gas optimization strategies in contract design.

The cost of smart contracts as well as their different functions is plotted in Fig. 7 and Fig. 8.

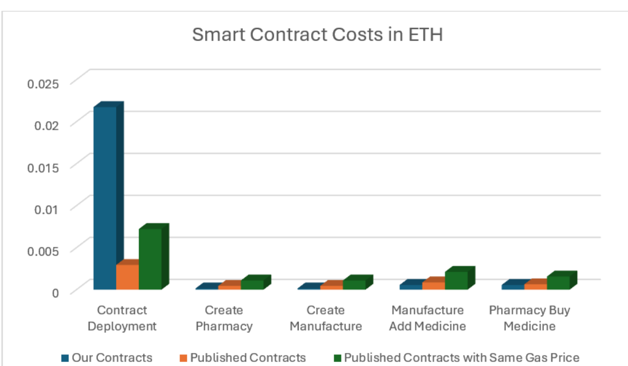

Fig. 7. Cost of smart contracts in ETH.

If we take in account of the comparison of costs in order to create multiple Manufacturers and Pharmacies, the sum of costs are illustrated in Fig. 9 and Fig. 10.

The major factors that take control of the environmental factors for the logistic control in our contract are illustrated below with their fetching functions, by which the caller can see the data of these factors without any need of the transactions.

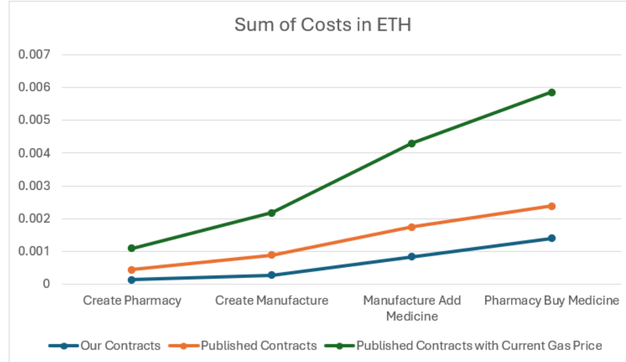

Fig. 8. Cost of different functions in smart contract.

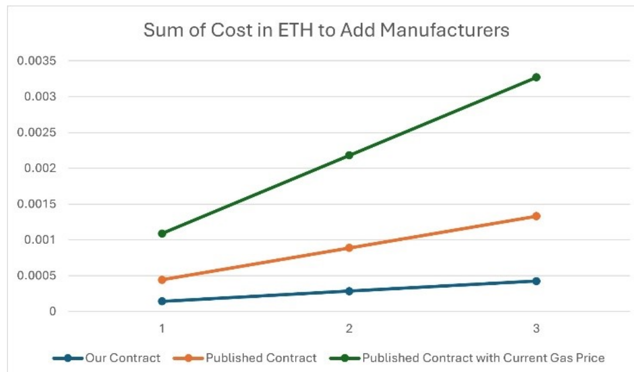

Fig. 9. Cost of adding three manufacturers in smart contract.

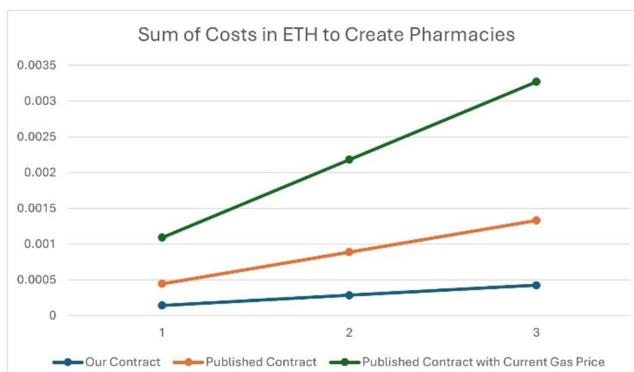

Fig. 10. Cost of adding three pharmacies in smart contract.

V. CONCLUSIONS AND FUTURE WORK

With our main goal in improving the overall gas cost and more probabilities in the environmental history of the drugs and to allow the exchange of data between multiple blockchain platforms, interoperability should become the primary area of research work since this would enable collaboration between different stakeholders. Besides the enhancement of supply chain process performance and actual time monitoring and decision-making through the electronic integration of emerging technologies such as IoT devices with smart contracts. Like the way the Internet altered business model, blockchain has nothing but the power to overhaul value exchange and have far reaching consequences in many different fields of life. Nevertheless, the management of expectation and dealing with the difficulties will be a must-have for the businesses in the fast-changing blockchain space.

To make interoperability among blockchain platforms possible, this should be the main area of research work because this would allow partnership among the various stakeholders. It's not only the advancement of supply chain process performance and the actual time monitoring and decision-making and the electronic integration of emerging technologies such as the IoT devices with smart contracts. The way the Internet changed the business model, blockchain will have no other power to overtake the value exchange and thus very high consequences in many areas of life. However, the instances of management of expectation and dealing with the challenges will be mandatory for the businesses in the rapidly evolving blockchain world. In the context of optimizing or improving the functionality and effectiveness of the contract in terms of overall cost and more environmental factors are yet to be discovered and researched more thoroughly to implement in the supply chain contracts.

Acknowledgment

This work was supported by the National Research Program for Universities (NRPU), Project Number: 15363, funded by Higher Education Commission (HEC) Pakistan.

References

[1] M. Muneeb, Z. Raza, I. U. Haq, and O. Shafiq, "Smartcon: A blockchain-based framework for smart contracts and transaction management," *IEEE Access*, vol. 10, pp. 23 687–23 699, 2022.

[2] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-enabled smart contracts: Architecture, applications, and future trends," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 11, pp. 2266–2277, 2019.

[3] J. Sun, H.-W. Long, H. Kang, Z. Fang, A. El Saddik, and W. Cai, "A multidimensional contract design for smart contract-as-a-service," *IEEE Transactions on Computational Social Systems*, pp. 1–17, 2025.

[4] D. Mishra and S. Phansalkar, "Blockchain security in focus: A comprehensive investigation into threats, smart contract security, cross-chain bridges, and vulnerabilities detection tools and techniques," *IEEE Access*, vol. 13, pp. 60 643–60 671, 2025.

[5] G. Kumar, R. Saha, M. Conti, and W. J. Buchanan, "Leagan: A decentralized version-control framework for upgradeable smart contracts," *IEEE Transactions on Services Computing*, vol. 18, no. 3, pp. 1529–1542, 2025.

[6] X. Zhou, K. Honari, H. Liang, S. Rouhani, and S. Dick, "Blockchain-based decentralized stochastic energy management in iot-enabled smart grid with voltage regulation coordination," *IEEE Internet of Things Journal*, pp. 1–1, 2025.

[7] J. Cai, J. Chen, T. Zhang, X. Luo, X. Sun, and B. Li, "Detecting reentrancy vulnerabilities for solidity smart contracts with contract standards-based rules," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 3662–3676, 2025.

[8] C. Xu, H. Xu, L. Zhu, X. Shen, and K. Sharif, "Enhanced smart contract vulnerability detection via graph neural networks: Achieving high accuracy and efficiency," *IEEE Transactions on Software Engineering*, vol. 51, no. 6, pp. 1854–1865, 2025.

[9] X. Zhang, Y. Song, Y. Du, C. Cai, H. Cheng, K. Xu, and Q. Li, "Smartupdater: Enabling transparent, automated, and secure maintenance of stateful smart contracts," *IEEE Transactions on Software Engineering*, vol. 51, no. 4, pp. 1266–1283, 2025.

[10] H. Huang, Z. Yin, Q. Chen, J. Zheng, X. Luo, G. Ye, X. Peng, Z. Zheng, and S. Guo, "Brokerchain: A blockchain sharding protocol by exploiting broker accounts," *IEEE Transactions on Networking*, pp. 1–16, 2025.

[11] A. R. Rajput, Q. Li, M. Taleby Ahvanooey, and I. Masood, "Eacms: Emergency access control management system for personal health record based on blockchain," *IEEE Access*, vol. 7, pp. 84 304–84 317, 2019.

[12] A. Jain, A. K. Tripathi, N. Chandra, and P. Chinnasamy, "Smart contract enabled online examination system based in blockchain network," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 2021, pp. 1–7.

[13] Y. Raj and S. B, "Study on supply chain management using blockchain technology," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2021, pp. 1243–1247.

[14] R. W. Ahmad, K. Salah, R. Jayaraman, I. Yaqoob, M. Omar, and S. Ellahham, "Blockchain-based forward supply chain and waste management for covid-19 medical equipment and supplies," *IEEE Access*, vol. 9, pp. 44 905–44 927, 2021.

[15] A. A. Mamun, S. Azam, and C. Gritti, "Blockchain-based electronic health records management: A comprehensive review and future research direction," *IEEE Access*, vol. 10, pp. 5768–5789, 2022.

[16] S. Yongjoh, C. So-In, P. Kompunt, P. Muneesawang, and R. I. Morien, "Development of an internet-of-healthcare system using blockchain," *IEEE Access*, vol. 9, pp. 113 017–113 031, 2021.

[17] A. Anusha, T. Shashidhar, M. Srivani, M. Srikanth, and A. Ramesh, "A blockchain-based approach for drug traceability in healthcare supply chain," *Journal of Engineering Sciences*, vol. 14, no. 4, pp. 195–209, 2023.

[18] K. C. Bandhu, R. Litoriya, P. Lowanshi, M. Jindal, L. Chouhan, and S. Jain, "Making drug supply chain secure, traceable and efficient: A blockchain and smart contract based implementation," *Multimedia Tools and Applications*, vol. 82, pp. 23 541–23 568, 2023.