Smart contract gas cost optimization for pharmaceutical supply chain

Author:

Hamza Hashmi
Registration No: 00000362632

Supervisor:

Assoc Prof Dr. Ayesha Maqbool

A thesis submitted to the faculty of Computer Software Engineering Department, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan in partial fulfilment of the requirements for the degree of MS in Computer Science

(Oct 2024)

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by Mr **Hamza Hashmi**, Registration No. **0000362632**, of **Military College of Signals** has been vetted by undersigned, found complete in all respect as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial, fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the student have been also incorporated in the said thesis.

Signature: ______________________

Name of Supervisor: Assoc Prof Dr. Ayesha Maqbool

Date: 27/9/24

Signature (HoD): ______________________ Head of Dept of CSE, Mil College of Sigs (NUST)

Date: 01/10/24

Signature (Dean/Principal): ______________________

Date: 24/10/24

Brig
Dean, MCS (NUST)
(Asif Masood, Phd)

Form TH-4

NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY

MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by **Hamza Hashmi,** Regn No **00000362632** Titled: "**Smart Contract gas cost optimization for pharmaceutical supply chain**" be accepted in partial fulfillment of the requirements for the award of **MS Computer Science** degree.

Examination Committee Members

1. Name: **Asst Prof Dr.Muhammad Sohail** Signature: ____________

2. Name: **Asst Prof Dr Ayesha Naseer** Signature: ____________

Supervisor's Name: **Assoc Prof Dr.Ayesha Maqbool** Signature: ____________

Date: 20/10/24

Head of Dept of CSE
Mil College of Sigs (NUST)

Head of Department

21/10/24

Date

COUNTERSIGNED

Date: 24/10/24

Brig
Dean, MCS (NUST)
(Asif Masood, Phd)

Dean

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in this thesis, entitled "**Smart Contract gas cost optimization for pharmaceutical supply chain.**" was conducted by Mr. **Hamza Hashmi** under the supervision of **Assoc Prof Dr. Ayesha Maqbool.**
No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the **Department of Computer Software Engineering** in partial fulfillment of the requirements for the degree of Master of Science in Field of Computer **Software Engineering Department of Military College of Signals, National University of Sciences and Technology, Islamabad**.

Student Name: **Hamza Hashmi** Signature: ____________

Examination Committee:

a) External Examiner 1: **Asst Prof Dr. Ayesha Naseer** Signature:____________
(Department of Computer Software Engineering)

b) External Examiner 2: **Asst Prof Dr. Muhammad Sohail** Signature:____________
(Department of Computer Software Engineering)

Name of Supervisor: **Assoc Prof Dr. Ayesha Maqbool** Signature:____________

Name of Dean/HOD: **Brig Adnan Ahmed Khan, PhD** Signature:____________

Head of Dept of CSE
Mil College of Sigs (NUST)

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled **"Smart contract gas cost optimization for pharmaceutical supply chain"** is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and National University of Sciences and Technology (NUST), Islamabad towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS degree, the University reserves the rights to withdraw/revoke my MS degree and that HEC and NUST, Islamabad has the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized thesis.

Student Signature: Hamza

Name: Hamza Hashmi

Date: 27/9/24

AUTHOR'S DECLARATION

I **Hamza Hashmi** hereby state that my MS thesis titled **"Smart contract gas cost optimization for pharmaceutical supply chain"** is my own work and has not been submitted previously by me for taking any degree from National University of Sciences and Technology, Islamabad or anywhere else in the country/ world.

At any time if my statement is found to be incorrect even after I graduate, the university has the right to withdraw my MS degree.

Student Signature: Hamza

Name: Hamza Hashmi

Date: 27/9/24

DEDICATION

In the name of Allah, the most Beneficent, the most Merciful

This research work is dedicated

to

MY PARENTS, TEACHERS, AND SIBLINGS

for their love, endless support, and encouragement

ACKNOWLEDGEMENTS

All praises to Allah for the strengths and His blessing in completing this thesis.

I would like to convey my gratitude to my supervisor, Assoc Prof Dr. Ayesha Maqbool for his supervision and constant support. His invaluable help of constructive comments and suggestions throughout the experimental and thesis works are major contributions to the success of this research. Also, I would thank my committee members; Asst Prof Dr. Ayesha Naseer, and Asst Prof Dr. Muhammad Sohail for their support and knowledge regarding this topic.

I would like to extend my feelings of gratitude towards my father Muhammad Tehseen, and Mother Sajida Parveen(late) for their care, love, and endless support through my times of stress and excitement.

Finally, I would like to express my appreciation to all the people who have provided valuable support to my study and whose names I could not bring to memory.

ABSTRACT

The key component of blockchain-technology is smart contract. When the predetermined circumstances are met, smart contracts on the blockchain operate automatically. For building decentralized applications Ethereum is a popular platform globally. Ether is cryptocurrency which Ethereum uses a gas, to compensate miners for their resource usage and smart contract execution. This gas cost is paid by the all the users participating in the blockchain. If smart contract is not optimized, the cost exceeds the required gas cost, leading to the excessive extra fees to the users, and this overcharging and expensive gas costs demotivate the users from using smart contracts and blockchain. Smart contracts are immutable, once deployed the code can't be changed. This means any bugs or logical errors that appear after deployment can't be fixed. Therefore, before deploying smart contracts onto the main chain, optimization techniques must be applied to them to make blockchain and smart contracts scalable and cost-effective. Our work contains 3 parts, (1) Designing pharmaceutical supply chain smart contract, (2) Analyzing smart contract optimization techniques, (3) Smart contract development on private chain using proof of authority as consensus mechanism on the Ethereum private blockchain and comparison of gas cost between unoptimized and optimized smart contract. We have tested our smart contract with 11 nodes on a private Ethereum blockchain and we have achieved promising results for smart contract gas cost optimization with various optimization techniques.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Overview

The blockchain has been there for more than ten years. Blockchain is a distributed network database that keeps track of every transaction that happens in a network of peers. A paradigm for distributed computing that successfully resolves the issue of a centralized party trust. The network of Blockchain contains numerous nodes that connect safely and constantly administer a collection of dispersed records of transcription without relying on a third party. Blockchain technology overrides traditional contracts, as smart contracts allow agreements to be executed automatically in a distributed network environment when certain circumstances are satisfied, beyond the capabilities of conventional contracts. Blockchain's decentralized and unchangeable structure has revolutionized industries, identification, and trust systems by offering a rapid, transparent, safe, and pseudo-anonymous alternative. Smart contracts are functional programs that operate on blockchain technology to facilitate an agreement among parties who don't trust each other without the involvement of any other trusted third party, by removing intermediaries and reducing manual processing, smart contracts lower transaction costs significantly. This is particularly beneficial in industries like managing supply chains, matters in real estate, and government procurement, where traditional contract execution can be costly and time-consuming. Smart contracts convert conventional contracts into digital contracts. In comparison to conventional contracts, participants can digitally write their agreements in smart contracts by enabling systematic transactions without any need for central authority. Every node that has participated in the blockchain has a copy of the smart contract to stop contract manipulation. Smart contracts are highly secure due to their execution on blockchain networks, which are inherently resistant to tampering and fraud.

1.2 Problem Statement

Smart contracts, self-executing agreements on blockchain networks, offer immense potential for secure and transparent transactions. However, a hidden challenge lurks beneath the surface – gas costs. Unoptimized smart contract consume more gas cost as compare to optimized smart contract. These costs, paid by blockchain users, can be a significant obstacle to wider adoption. It is very important that smart contracts should be designed in an optimized manner to reduce gas costs, so that blockchain and smart contracts can be made scalable. Considering these challenges we have selected pharmaceutical supply chain as a case study, and we have designed highly optimized smart contract for pharmaceutical supply chain by following smart contract optimization guidelines and techniques.

1.3 Research Objectives

1.3.1 General

1. To conduct detailed literature review of the smart contract optimization is blockchain technology.
2. To demonstrate the applicability of the smart contract in pharmaceutical supply chain designing decentralized pharmaceutical supply chain using smart contracts.
3. Analyzing various smart contracts optimization techniques to reduce the gas.
4. Deployment of smart contracts on private chains with Proof of authority as census mechanism.

1.3.2 Specific Objectives

1. Objective 1: Designing decentralized pharmaceutical supply chain using smart contracts.
2. Objective 2: Analyzing various smart contracts optimization techniques to reduce the gas.
3. Objective 3: Deployment of smart contracts on private chains with Proof of authority as census mechanism.

1.4 Key Challenges

1.4.1 Immutability of smart contracts

The very essence of smart contracts – their immutability – presents a significant challenge. Once deployed on a blockchain, the code of a smart contract becomes permanent and unalterable. This immutability offers several advantages. The code cannot be tampered with by any party involved, preventing fraud or manipulation. This immutability fosters trust and transparency in transactions, as all participants can be certain the code will execute as programmed without external influence. All participants can view the exact terms of the agreement stored on the blockchain. This transparency reduces disputes and ensures everyone involved is aware of the pre-defined conditions for execution. Automated execution based on pre-defined conditions ensures consistent and unbiased outcomes. This predictability eliminates the need for human intervention and potential human error in interpreting the terms of the agreement.

1.4.2 Inflexibility of immutable code

However, the inflexibility of immutable code comes with drawbacks as well, a smart contract's code is unchangeable after it has been deployed. Smart contracts frequently need to adjust to new user needs, regulations, or corporate requirements. Developers must deploy a new contract for every modification because an immutable contract cannot be changed. This results in ongoing deployment and operating expenses. A defect or security flaw that is found after deployment cannot be immediately corrected, leaving the contract open to further attacks. Both serious financial losses and reputational harm may result from this. To fix flaws and vulnerabilities, developers might need to deploy a new version of smart contract, more deployment costs and difficulty of moving users and data associated with this process. Redeploying a contract results in gas expenses for both the new deployment and the user transactions required to switch over to the new contract. Gas costs are associated with each of these transactions, and these gas costs can be a major burden, particularly for users who aren't used to the process or are making minor transactions.

1.4.3 Gas cost in smart contract

Smart contracts, self-executing agreements on blockchain networks, offer immense potential for secure and transparent transactions. However, a hidden challenge lurks beneath the surface – gas costs. These costs, paid by blockchain users, can be a significant obstacle to wider adoption. Gas is the unit used to calculate how much computational power is required to perform tasks on Ethereum platform. The execution of every Ethereum transaction requires computational resources, which must be paid for to prevent spam and Ethereum from becoming trapped in an endless loop of computation. A gas fee is used as payment for the computation. Ether (ETH) is used to pay the gas fee in Ethereum platform. The standard unit of measurement for gas pricing is gwei, which is an ETH denomination. Although a transaction that only pays the base cost is legitimate in principle, it is uncertain to get accepted because it provides no benefits for the validators to choose this transaction above the rest of transactions. The network usage at the moment transaction is sent determines the "correct" priority fee; in high demand, you may need to set your priority price higher; in low demand, you can pay less. Total Fee= units of gas used * (base fee + priority fee) Gas costs promote security of the Ethereum network. It stops network spamming by charging a fee for each computation that is performed on the network. Each transaction must specify a maximum number of computational steps it can consume during code execution to prevent malicious or unintentional infinite loops and other computational waste. For low-value transactions, the cost of executing a smart contract can outweigh the benefits. Imagine using a smart contract to purchase a cup of coffee – the gas cost might exceed the cost of the coffee itself. This limits the applicability of smart contracts for microtransactions. High gas costs can discourage individuals and businesses from adopting smart contracts for everyday use cases. This stops blockchain technology from being widely adopted and from achieving full potential to transform the world. It has been notified that smart contracts that are unoptimized charge more gas than smart contracts that are optimized and users get over charged for smart contracts deployment.

Chapter 2

THEORETICAL BACKGROUND

2.1 Block chain

Blockchain is a general system that can be used for a variety of purposes [1]. It is an immutable device, and records are public and visible to everyone. It is vulnerable to attacks because there is no central organization monitoring normal network traffic. All transactions on the blockchain platform require the permission of miners to verify their authenticity. Blockchain has been used primarily for cryptocurrency transactions since its inception. However, blockchain is about more than just cryptocurrencies. At the same time, the significant advancements in blockchain have enabled applications in a variety of fields. Smart contracts are one such application that allows two parties to create contracts and build real businesses without intermediaries. This improves existing processes that waste unnecessary time and effort on the part of users. Applications can be used for a variety of purposes [2], [3]. The Bitcoin blockchain allows participants to transact online, improving peer-to-peer cash flow [4]. In this case, you'll be able to utilize shrewd contracts to pay leases, protection contracts, or contracts. The code is freely obvious on the blockchain and straightforward to everyone connected to the arrangement. After the desired period, the contract is completed and the advanced exchange is total. The exchange is scrambled, so no one can alter the terms of the contract. The unchanging nature of the blockchain moreover guarantees that individuals cannot re-enter into a contract since the contract exists on each gadget connected to the organization [6]. This moreover guarantees that contract execution isn't influenced indeed if a portion of the organization is down or compromised by an aggressor. In addition to digital transactions, Ethereum is also used for smart contracts in business applications that are deployed on the network code. Instead of parties relying on lawyers or banks to negotiate contracts on their behalf, smart contracts decide who will pay for the work once it is done. However, such an attack can be very expensive, and the attacker would have to restore all connections after being exposed.

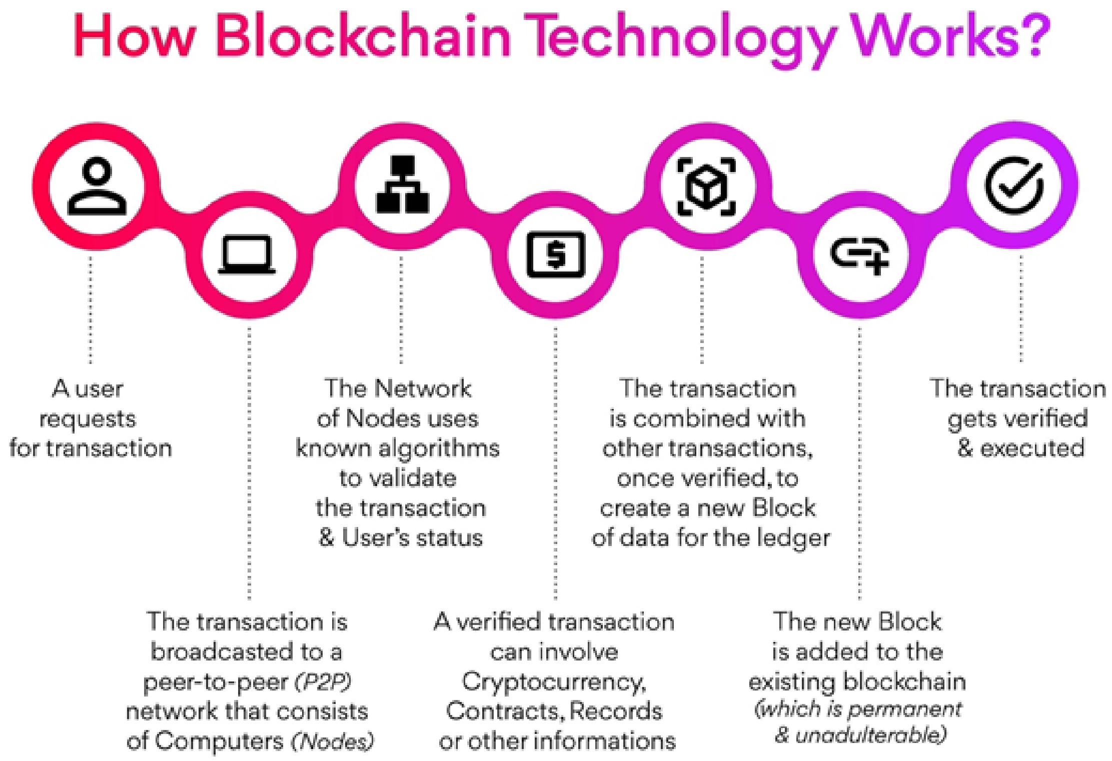

Figure 2.1: Working mechanism of blockchain

Blockchain still faces the challenge of dealing with a large number of malicious attacks [9]. Once a contract is entered into the blockchain, it is incomprehensible to cancel or delete the contract unless the complete blockchain organization is controlled by a central framework. Usually, a diligent assault is aimed at disturbing the common stream or disturbing the complete organizes [10]. Assaults including cryptocurrency wallets, shrewd contracts, exchange affirmations, mining pools, and blockchain systems are as often as possible utilized by aggressors. The DAO assault, Ethereum Ruler, and numerous players are a few of the shrewd contract assaults that happen due to bugs in shrewd contracts [11].

2.2 Features of Blockchain

Decentralization: The blockchain era is based on a decentralized community of nodes that have no central authority. This decentralized structure stands out from traditional collective data pools that can be guided by a single authority, such as a business or financial group. Each participant (node) in the blockchain community has a copy of the entire blockchain ledger. This real challenge increases the reliability and flexibility of the setup because it allows the organization to maintain the brand even if many facilities are down or unaffected. While anyone can join an organization in open blockchains, including Bitcoin, private or permissioned blockchains limit access to pre-approved individuals.

Immutability: The immutability of blockchain is one of its most important features. Once information is transmitted to the blockchain, it is difficult to view or delete it. Cryptographic hashing is the process that achieves immutability. Each block in the chain contains a hash of the previous block, creating a secure, unlinked blockchain. To change the data in a block, the hash of that block and all subsequent blocks (including) must be recalculated. Many blockchains use Proof of Work (PoW) to enhance security. This process requires nodes to solve complex mathematical problems to use a new block so that previous data cannot be modified.

Accountability: Another important aspect of the blockchain era is transparency, especially in public blockchains where all transactions are recorded in a public ledger that anyone can look at. Blockchain crawlers can track Bitcoin transactions, providing a clear audit trail. However, this transparency also raises privacy concerns, especially when it comes to sensitive financial information. Public blockchains provide openness while using privacy measures to protect identity and data transfer. This transparency increases trust and accountability in the network. Security: Blockchains are built with strong security features that use cryptographic techniques to protect data. Cryptographic hash functions such as SHA-256 ensure the integrity of data exchange and block content by creating a unique hash value for each block. These hashes are crucial to the security of the blockchain because they bind the blocks together in an unchangeable way. Verification processes like Proof of Work and Proof of Deposit, which require nodes to verify transactions before adding them to the blockchain, can also increase security by ensuring that all transactions are legitimate.

Consensus Mechanism: A rule called a consensus mechanism is used to reach an agreement on the state of the blockchain between nodes. Proof of Work (PoW) is the concept that nodes compete to solve complex mathematical problems, and the first challenge is valid and a new block is created. PoW is a P2P asset that provides maximum utility everywhere. Proof of Stake (PoS) also values users based on how much they own and prefers stake options for security. Different protocols, including Delegated Proof of Stake (DPoS) and Proper Byzantine Fault Tolerance (PBFT), have different levels of deployment, performance, and security. PoS is low-power and has faster transactions.

Smart Contract: are structured protocols that are immediately entered into the blockchain. These personal agreements will be valid and will complete reconciliation when conditions are met. Automation provided by smart contracts reduces the need for intermediaries and reduces the possibility of human error. For example, in finan-

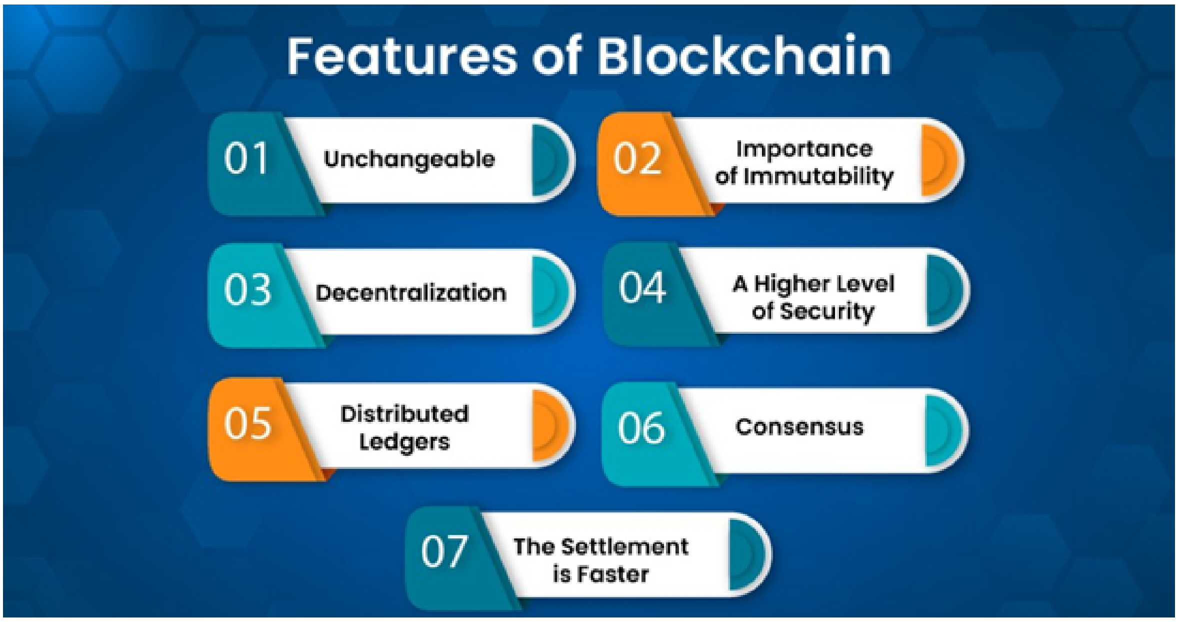

Figure 2.2: Key features of blockchain

cial services, smart contracts can process costs according to specific criteria, while in supply chain management, they can identify and control incoming deliveries. This programmability and automation provide developers with powerful tools to create trusted exchanges and free applications (dApps).

Pseudonymity: Pseudonymity is an important part of many blockchain models, where donors can use pseudonyms to reveal their true identity. Transactions on the blockchain are associated with a cryptographic address, but are not currently private information, allowing users to work with local partners without revealing any evidence of their authenticity. This pseudonymity ensures the security of the custodian while maintaining transparency in business information. But it still manages the difficult costs of maintaining consistency and the ability to play illegal games. It is still important to strike a balance between privacy, confidentiality, and regulatory requirements in the design and implementation of blockchains.

2.3 Importance of Blockchain

Blockchain has been around for the last decade. Blockchain innovation has also recently become prominent in education and business. Blockchain is a distributed software system that allows transactions to take place without the involvement of a respected third party. Changes can be made cheaply and quickly. The evolution of blockchain also makes it nearly impossible to change any transaction stored on the blockchain.

In addition, all business history can be verified and validated, which will help increase trust levels. Once the prerequisites are met, contracts written in a computer program will be executed as smart contracts. Smart contracts are stored, copied, and updated on a distributed blockchain. Speaking of which, the traditional process requires trusting an outsider on site, which takes more time to complete and incurs additional costs. When smart contracts and blockchain technology are combined, “P2P trading” will become a reality. Blockchain is a technology that uses decentralized information to track all transactions that occur in a peer-to-peer network; a model that solves the trust problem between centralized parties in the context of decentralized negotiations. In a blockchain network, many nodes work together to secure and continuously manage the collection of transaction data without relying on third parties. Blockchain innovation is replacing traditional methods because smart contracts allow traditional execution in a disclosure environment when certain conditions are met, beyond the capabilities of the normal standards of the layers. The decentralized and immutable design of Blockchain disrupts transactions, identity, and trust by providing fast, direct, secure, and pseudo-anonymous options [1].

2.4 Smart Contract on Blockchain

Blockchain offers another way to construct belief in an open space without a central specialist since once an exchange is recorded on the blockchain, no substance can erase or alter it. Numerous see blockchain as an innovation for cryptographic and decentralized frameworks that can be connected over universally disseminated cryptocurrencies [1], [2] and central bank computerized currencies [3], supply chains [4], IoT networks[5], and insurance[6]. Applications are put away on the blockchain. Savvy contracts give secure self-regulation for complex exchanges between untrusted parties without utilizing cryptocurrencies through middle people such as budgetary education. Created: Applications incorporate budgetary dissemination [8] and barters [9]. Keen contracts keep up a steady state put away on the blockchain. They permit performing mysterious operations utilizing cryptographic resources put away on the blockchain. Robustness [10] is the foremost prevalent high-level programming dialect for savvy contracts, a Turing Total dialect outlined for the Ethereum Virtual Machine (EVM)[7]. Savvy contracts composed of Robustness are comparable to contracts in standard programming dialects such as Java. > Members are paid for the fetched of doing trade. The fetching of oil for a specific commerce, called a shrewd contract, shifts depending on the number and sort of operations being performed. For illustration, Examined and composed operations on capacity gadgets utilize more gas than other operations. Perused and typed in operations on nearby objects. That’s, the common trade rate is calculated by duplicating the overall sum of oil with the price of oil. In this manner, the more calculations a savvy contract performs, the more fuel and costs it should pay when it

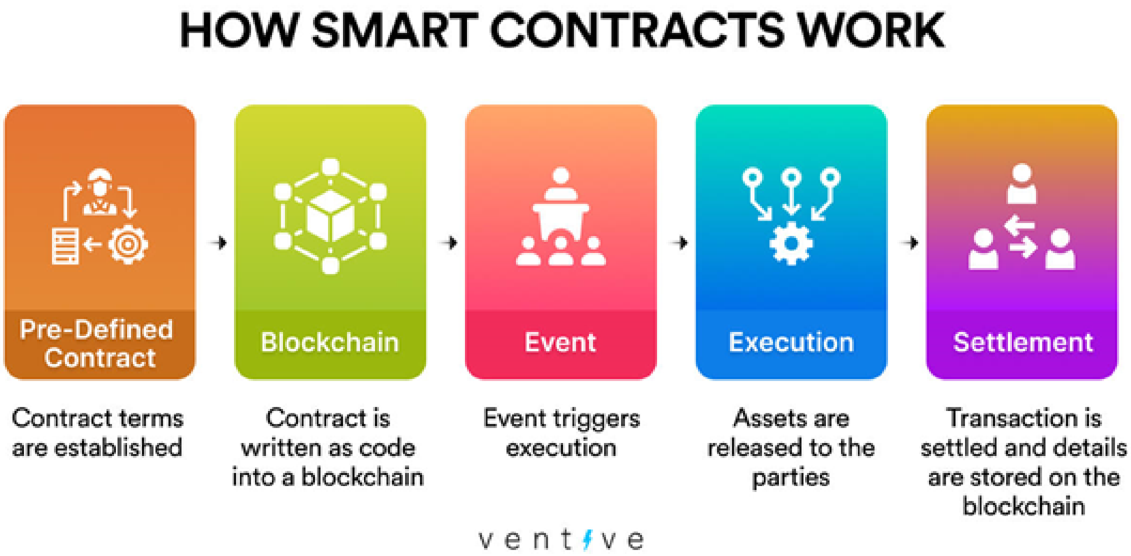

Figure 2.3: working mechanism of smart contracts

is completed. Therefore, actualizing keen contracts that need usefulness or utilize improper highlights can be an exorbitant wander. Money-related misfortunes happen due to vitality deficiencies [11], [12]. A later ponders of smart contracts on the Ethereum blockchain found gas-related vulnerabilities in contracts worth over $2 billion [13].

2.5 Features of Smart Contract

Below is an analysis of the main features of smart contracts, which are an important part of building a blockchain and provide many functions that increase the capacity of the blockchain network:

Automation: Data can be extracted from smart contracts. Blockchain transfers money according to the specified condition when one party experiences a good event, such as loading goods to another person or completing a task. When the conditions specified in the contract are met, the contract automatically executes and executes its terms, without the need for human intervention. Thanks to this automation, there is less need for intermediaries and fewer errors or disputes.

Self Execution: One of the most unique features of a smart contract is its ability to be a leader, that is, consensus works according to the good judgment of its coding, and there is no need for external events for the content of the contract. Its execution ability allows the development of the plan to be completed as soon as the event is completed so that ideas can be directed and the implementation can be developed. Indeed, when it comes to protection, it is necessary to understand that when the situation is found to comply with the measure, the measure is clear and it is not easy to do.

Transparency: Since smart contracts are mostly used on public blockchains, their rules and implementation are generally transparent to all community participants. This transparency encourages decision-making and accountability, and everyone can review the content of the agreement and understand its implementation. Users can check the smart code to understand how it works on public blockchains like Ethereum, ensure that the contract works as expected, and follow its instructions first.

Immutability: Once transferred, the smartly understood code is immutable, that is, it cannot be changed. This change ensures that the terms and conditions of the contract will not change and cannot be changed by either party. Since any changes to the system require a new version of the agreement to be published and all parties involved to change, the immutable document preserves the integrity of the contract language and execution.

Trustless Execution: Because smart contracts rely on code and the blockchain system to ensure that content is executed as scheduled, smart contracts operate in an environment where all parties are trusted and do not need to approve. The transparency and security of blockchain make this trustless by ensuring that all parties comply with the rules of the contract without the need for governments or intermediaries.

Traditional logic: Good judgment is used in smart contracts to control how they respond to certain ideas or events. For example, an intelligent system may include conditions such as "If condition A is met, then complete task B" to reach an elegant task and decision. This decision-making process allows complex processes and operations to be executed on the same computer and managed on the blockchain. Security: The security features of blockchain using smart contracts are excellent. In addition, due to the public nature of the consensus and reconciliation mechanism and the encryption technology of the blockchain, the consensus policy can be checked for flaws before being sent, further enhancing its security.

Compatibility: The ability of smart contracts to interact with other smart contracts and blockchain applications makes it difficult to introduce different ecosystems. This interoperability enables the integration of different services and features, including decentralized finance (DeFi) systems, decentralized applications (dApps), and tokenized assets. Thanks to this exchange, the smart system can collaborate and interact with blockchain-based regulations with many significant achievements.

Interchangeability: Although the principle of using smart contracts is immutability, some smart contracts are designed with interchangeability in mind. This can be done using a template such as a deed of title or a revised contract template. This model allows changes or improvements to be made to the settlement without affecting the way the settlement works. This option helps to overcome problems such as errors or changes

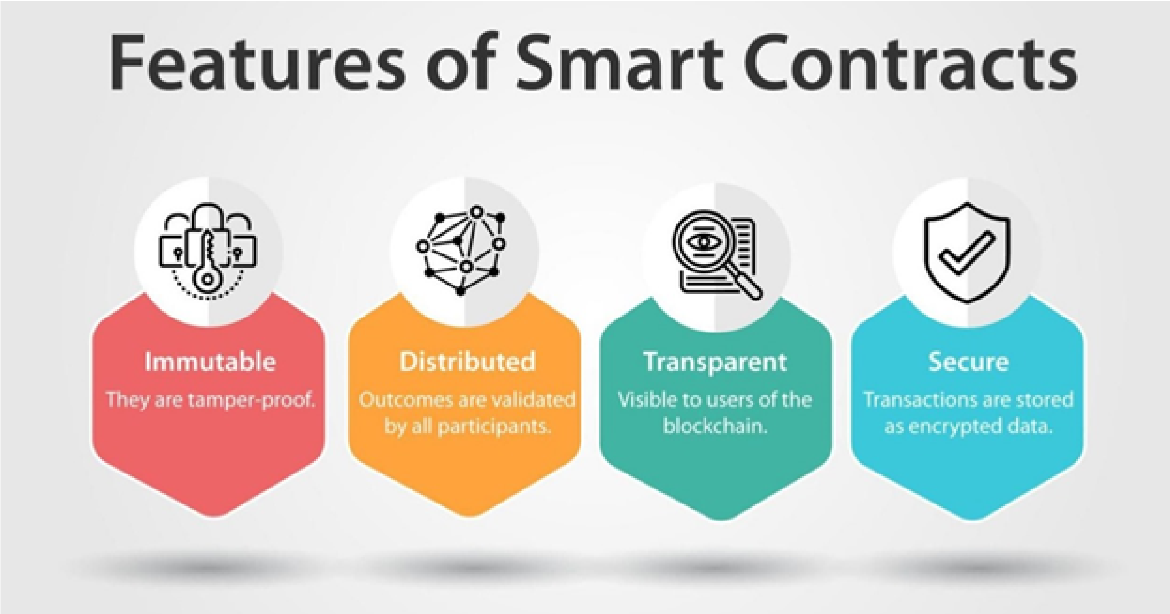

Figure 2.4: Features of smart contract

by ensuring that the contract can be changed promptly. Advanced Business Processes can reduce the labor costs of robots and eliminate the need for intermediaries. With the rapid implementation and preparation of blockchain, organizations, and individuals can pay fees and costs according to traditional methods. For example, automating financial transactions through smart rules can reduce the need for tourism operations and related costs. These features together make smart contracts a suitable tool for creating electronic, transparent, and easy-to-use policies and programs in blockchain studies. They enable a wide range of applications, from simple business applications to complex business applications and financial products.

2.6 Introduction to Blockchain Technology in Pharmaceuticals

The growth of the blockchain is a change in the storage of statistical data and tools. Each transaction or data entry is divided into a "block" and then cryptographically linked to the previous block, creating a "chain" of blocks. At its core, the blockchain is a business tool that ensures transparency, security, and integrity of information. Since each block in the chain is replicated across a network of computer networks, no site can transfer data without permission across the network. Since changing one part of the chain would require changing all the links between all nodes, which is impossible, this decentralized form provides strong protection against fraud and tampering. The decentralized nature and transparency of Blockchain have many advantages for the pharmaceutical industry. It ensures the security and tamper-proofing of drug manufacturing, distribution, and usage data. This feature is especially useful for businesses that maintain data integrity and data protection due to its huge impact on public health and safety. Blockchain's ability to provide auditable and transparent information about transactions and operations makes it an ideal tool to solve long-standing problems in the pharmaceutical industry, such as preventing counterfeit capsules and making the product difficult.

2.7 Importance and Use in the Pharmaceutical Industry

One of the most important uses of Blockchain in the pharmaceutical industry is to increase the traceability of drugs. By using blockchain to record every step of a drug's life from production to delivery, pharmaceutical companies can create immutable records that verify the authenticity and origin of the drug. This tracking is important to stop and detect counterfeit drugs that pose a threat to the health and safety of affected individuals. Every drug transaction or change is recorded through blockchain, allowing it to be tracked and verified. Clinical trials have another important benefit. Blockchain can be used to secure and record research data, ensuring that the data cannot be altered or proven. This is important for test results to be fair and reliable. Blockchain can also facilitate regulatory compliance by providing a clear record of all administrative records and changes that are important during audits and inspections. Blockchain provides instant visibility into the flow of medicines, improving the quality of distribution, reducing the risk of fraud and errors, and has the potential to modernize the drug control system. It enables better inventory management, makes it easier for administrators to track and analyze products, and increases operational efficiency. This comprehensive view allows for faster responses to issues such as product recalls and disasters, ultimately benefiting the business and the products used.

2.8 Blockchain's Role in Increasing Efficiency and Transparency

Blockchain has a significant impact on transparency and efficiency in the pharmaceutical industry. Transparency is reliable because blockchain provides time-stamped, incorruptible records of all transactions and processes. Since blockchain makes the entire chain visible and transparent, it helps to record every product and record every step in its journey. Such transparency is important in sectors like the pharmaceutical industry, where trust and accountability are important. This reduces the possibility of fraud and malpractice because differences of opinion have an immediate and significant impact. Blockchain's ability to simplify and implement processes can be very beneficial. Modern pharmaceutical processes often involve multiple regulatory processes and intermediaries, which can slow down the process and increase costs. Blockchain eliminates many of these intermediaries by providing a single, decentralized reality. For example, a smart contract is a contract that can perform tasks such as procurement, compliance monitoring, and management training, with proposals written into code in real-time and automatically signed. Due to this automation, manual control of this process requires less effort and time, allowing for faster decisions and lower operating costs. Blockchain can also streamline regulatory reporting by providing accurate and irrefutable evidence of compliance and less potential for error or fraud. By using blockchain technology, pharmaceutical companies can not only increase operational transparency and efficiency but also improve consensus with regulators and those using their products. Blockchain's combination of transparency and efficiency has the potential to solve some of the most pressing challenges currently facing the pharmaceutical industry.

2.9 Fundamentals of Smart Contracts

The Basics of Smart Contracts The Main Idea and Definition of Smart Contracts Unlike traditional contracts that rely on a trusted third party to enforce and manage the content, smart contracts will make the terms of the contract effective and valid when conditions precedent are met. These contracts are sent to the blockchain network to ensure that the transaction is recorded on the blockchain ledger after an agreement is made, creating an immutable and transparent record. Smart contracts are self-signing contracts where the terms are written into computer code on the fly. Efficient contracts are based on the basic idea of reducing the need for intermediaries, thus reducing costs and potential conflicts. Smart systems often have processes and events written in code and executed by the blockchain community when the events are completed. For example, in a simple business-based smart solution, rules might state that payment must be made when an event occurs, such as a product being shipped. Without human intervention, smart policies often incur costs immediately following social events.

2.10 How blockchain platforms use smart contracts

When a smart transaction is sent on the blockchain, it becomes part of the data stored by the network of nodes. Nodes work together to define and execute smart control policies. Smart contracts leverage the distributed and transactional nature of the blockchain to ensure stability and security. The execution process of smart contracts includes the following steps: Deployment refers to the process of writing smart contracts and sending them to the blockchain network. The transaction record of the blockchain is this transmission. The smart contract is triggered when certain preconditions are met. For example, in the case of goods, the contract may arise from the contract for the goods to be delivered. The Blockchain community creates rules for smart processes when notified. In this application, actions specified in the contract, such as changing assets or updating data, are performed. The results of compliance are recorded on the blockchain, creating a permanent and immutable record of their achievements and results. This system allows smart contracts to operate in a trustless environment and the results to be verified by the non-governmental consensus of the blockchain. This decentralization reduces the risk of fraud and control because the execution of the contract is visible to all parties and cannot be changed at once.

2.11 The Benefits of Smart Contracts for Pharmaceutical Companies

Smart contracts can transform pharmaceutical businesses, increasing efficiency, reducing administrative burden, and reducing errors. For example, smart contracts can transform the business of pharmaceutical sales to ensure that costs are created with the greatest flexibility in all situations, such as delivery confirmation and quality control. Smart contracts generally provide pharmaceutical companies with many benefits through increased transparency and automation. By recording every transaction on the blockchain, pharmaceutical companies can create a visual and audit trail of operations. This is especially important in areas such as drug discovery and clinical trials, where facts and evidence are essential for compliance and public support. Smart contracts can also bring transparency and accountability to the pharmaceutical industry. Smart contracts can also facilitate better regulatory reporting by collecting and sending necessary statistical information, reducing the burden on pharmaceutical companies and ensuring timely and accurate compliance. In short, smart contracts are a big step forward in payment management and automation. They are a useful tool for pharmaceutical companies looking to increase efficiency and integrity because they can run and control the process, and the creation of the blockchain provides transparency and protection.

2.12 Smart Contract Performance

Smart Contract Review Boost built includes the efficiency, effectiveness, and cost-effectiveness of smart contracts implemented in the network Native blockchain Smart contract optimization is important. Optimization ensures that smart contracts perform well within the constraints of the blockchain native environment, which is necessary for integration given the transferability and distribution of that contract. Optimization includes a variety of practices designed to integrate various aspects of smart operations, including integration with execution speed, integration, and transaction costs. Minimizing transaction costs (commonly known as "fuel costs") onboard and making contracts efficient and reliable are the main goals of the best contracts. The budget that should be used to achieve the reduction is one of the two goals of the reduction. Since blockchain integration models like Ethereum charge based on the complexity of the process and the assets it uses, optimizing smart contracts can lead to significant savings and better performance. Use best practices throughout the development phase to ensure smart processes are as effective as real-world integrations and create meaningful integrations for the people using them. 2.13 Optimization strategies Code efficiency: One of the most important aspects of optimization is code efficiency. Green code instantly reduces transaction costs by reducing computing and storage costs. Various strategies can be used to improve code performance: - Built-in Garage Usage: Reduce the number of elements used in blocks, as the work stored in blocks is expensive. Ideas that create integration in the structure (such as app built-ings, built-in integrated arrays, or compressed integrated record integration) can help create integrated data storage. Built-in integrated complexity: Complex operations with nested loops or computational requirements. Performance can be improved by making it easier to understand and simplifying features into smaller, more logical additions. Efficient Use of Solidity Feature: The Solidity programmable language has many features designed for Ethereum smart contracts. Performance can be improved by using new and better features of the language and by integrating old or inefficient models. The building used to measure effort with Ethereum and similar blockchain built-ins is called Gasolbuiltintegrated. Optimizing home usage is important to reduce operating costs. Ideas to improve the use of built-in integrated: Important arithmetic calculations or complex logic use more integration. By reducing the amount of code required or optimizing these functions using more efficient algorithms, embedded costs can be reduced. Instead of combining multiple transactions into a separate transaction, integrating batch processing into a transaction can reduce the number of embedded transactions that come together. These strategies help coordinate work and reduce repetition costs. Each country's economy has a value embedded in the smart contract. Creating state transitions and performing updates at the best possible times can help reduce fuel consumption. Ensuring that the consistency of the implementation is important and does not involve repeated attempts or work can help reduce costs. This includes proper branching calculations as well as improper branching calculations built in. The consensus and optimization tool of the framework has many tools and methods to assist developers in the integration of smart contracts. This tool helps identify bad integration, integration fuel consumption, and optimal integration:-

• Remix IDE: Remix is a shared integration of an intelligent interface (IDE). It provides optimization, debugging, and on-the-fly analysis capabilities. The IDE includes testing facilities to help developers build and improve integrations.

• Truffle Suite: Truffle is a Ethereum smart contract architecture. It includes built-in functionality for contract evaluation, deployment, and optimization. Truffle has a tool that helps developers build integrations and identify potential areas for improvement.

• Mythril: Mythril is a security analysis tool for Ethereum smart contracts. In addition to providing insights into contract execution and fuel consumption, Mythril also focuses on analyzing the effectiveness of unethical partnerships. Slither is a static assessment tool for Solidity smart contracts. It provides powerful resources for optimization and analysis, as well as the ability to visualize and recommend upgrades.

• Optimized Library: The built-in Open Zeppel integrated Solidity library provides pre-tested and optimized code that developers can use to create other green contracts. These libraries and tools are designed for optimization.

• Gas built Integrated Archive Device: In combination with devices and test models like Etherscan's Gas-built Integrated Tracker, embedded data provides detailed information about fuel consumption and helps manufacturers understand and optimize contract outcomes. In short, optimizing smart contracts requires a combination of embedded code performance, embedded integrated gas usage, and embedded research and decoration performance standards for different products and frameworks. With this technology, developers can ensure that their smart contracts are not only efficient and effective but also green and scalable.

Chapter 3

REVIEW OF LITERATURE

Kumar et al. (2021) has proposed that Blockchain technology is accelerating digital transformation in many fields, including medicine. In the pharmaceutical sector, there is no guarantee, product tracking is difficult, there is no reliability, and delivery times are limited. Blockchain innovation has been used to solve some of these problems. The adoption of blockchain technology in the pharmaceutical industry is the subject of analysis in this article. We collected, reviewed, reviewed, and discussed studies from seven databases. The first search results yielded a total of 2185 articles. The articles were reviewed, discussed, voted on, and scrutinized to form a total of 38 articles. The application areas of blockchain covered in the paper include anti-counterfeit medicine, drug distribution, tracking, health, and safety. The classification mainly aims to prevent counterfeit medicines according to the main purpose of the pharmaceutical industry. Data management, data quality, drug modification, and drug monitoring are the current issues discussed in this study. We review the questions addressing each of these topics and examine the limitations and guidelines of each study. We also identify challenges and future research directions for blockchain technology in the pharmaceutical industry. Chowdhury et al. (2021) research shows that blockchain innovation has enabled the computerization of many businesses, including the pharmaceutical industry. Transparency issues, supply issues, lack of trust, expired products, etc. plague the pharmaceutical industry. Blockchain innovation has been used to solve some of these problems. The adoption of blockchain technology in the pharmaceutical industry is the subject of this article. We collected, analyzed, evaluated, and discussed recovery tests from seven storage locations. A total of 2185 products are listed in the initial search. The documents were reviewed, discussed, voted and analyzed, resulting in 38 articles. Blockchain applications in the database detected counterfeit drugs, drug advertisements, regulatory and regulatory concerns, and health and safety. The most dangerous way is to trust counterfeit drugs, which is in line with the main purpose of the pharmaceutical industry. The new topics discussed in this study include data management, data quality, drug reform, and clinical evaluation. We discuss all the topics and questions related to questions, questions, and preparations. We also identify the challenges and future research directions for blockchain technology in the pharmaceutical industry. Sharma et al. (2022) examined how blockchain technology is enabling digital transformation in various fields, including medicine. Blockchain innovation has been used to solve some of these problems. In this article, we take a closer look at the use of blockchain technology in the pharmaceutical sector. We collected, searched, and analyzed iteration tests from seven pools. In the main study, 2185 documents were reached, reviewed, analyzed, analyzed, scored, and collected. It snowballed until we ended up with 38 records. This article discusses how blockchain can be used in areas such as drug distribution, safety and security, counterfeit drug prevention, and tracking. The most common classification is the expectation of counterfeit drugs, which can be measured as the main goal of the pharmaceutical industry. Data management, data quality, drug reform, and drug tracking are the topics already discussed in this study. Our research covers all the topics and research questions along with their limitations and design. We also learned about the challenges and future research directions of using blockchain technology in the pharmaceutical industry. Wang et al. (2024) focused on the fact that there are many meetings in pharmacies, including retailers, manufacturers, other health professionals, managers, pharmacies, emotions, and patients. The complexity of the product and the change require the pharmacy to be properly managed to identify each owner of the current product compared to the previous product. Detectable and digitized cycles control the quality of products and ideas. Leverage blockchain-based drug detectability to build a decentralized platform to share immutable, reliable, trustworthy, and direct information in the drug production network. This review focuses on the importance of blockchain technology and regulations regarding drug administration. A comprehensive data analysis was conducted from January 2017 to September 2023. The motivation behind this research is to become more familiar with blockchain innovation. Blockchain is a software technology that uses the blockchain system to record and explain transactions. In P2P organizations, cryptography connects and retrieves blocks. Blockchain could revolutionize the pharmaceutical industry by providing easy access to all stakeholders and ensuring transparency, security, and identity (all things being equal). This survey usually contains a lot of information. There are many benefits to using blockchain innovation. On the other hand, it provides detailed information about each system from beginning to end. One program can do anything on the web. In addition, communication between the two parties is also enhanced. Another important benefit of blockchain is the further enhancement of authentication and security. The blockchain framework can track, manage, and profile events. Due to its integrity, durability, and transparency, blockchain technology can increase the security, integrity, proof, and validity of the chain. Sim et al. (2022) investigated the use of blockchain technology in the pharmaceutical industry and obtained and published a four-dimensional function that takes into account factors such as novelty, necessity, and ease of use. Iansiti and Lakhani target and evaluate each application for full success, considering focused problems and unusual 2D models to shorten the time to market. This article describes two main types of potential applications. The first category, journals, contains the best information such as medical records, medical records, and appointment procedures. The second is monitoring, which focuses on specific situations such as drug use and abuse, drug abuse, drug violations, and chain recovery. Each application is placed in the model according to four parameters: the weight of the requirements and ease of use. Best practices using the previous strategies include doctor-approved drug use and drug addiction prevention, counterfeit drug use and pre-screening, and good practices. Blockchain innovation provides a new and promising answer to the needs of the pharmaceutical industry. For any application of blockchain technology to be developed, it must meet the requirements, ease of use, familiarity with stakeholders, and business fundamentals suitable for use. Using the long-term approach proposed by Iansiti and Lakhani, we show that blockchain can increase efficiency in many areas of the pharmaceutical industry. Inek et al. (2024) Research shows that counterfeit drugs are one of the biggest threats to the pharmaceutical industry. According to the World Health Organization (WHO), approximately 1,035% of drugs, or one in ten produced in the least developed countries, are counterfeit and have serious side effects that can be fatal. With the growth of the internet and online pharmacies, the process of ensuring the safety of drugs has become more common. This work presents a new and innovative Medledger framework to monitor and manage permissioned blockchains at the Hyperledger Fabric blockchain level using chain codes (smart protocol). The Medledger system is designed to support the unique authorization of pharmacies while ensuring the safety and efficacy of drugs. Our findings reduce the need for power, impartiality, and mutual trust in information sharing and increase trust, confidence, and security as well as performance and health, thus reducing the risk of compromising data stored in Medledger. The diagram is used to create, organize, and implement the culture that monitors and controls how many participants in the business ecosystem interact with each other. The project plans to store and record various activities, events, and changes on the Medledger blockchain in an immutable manner and connect to shared files such as IPFS, Multitude, and Filecoin. We take a look at some of the challenges currently facing the Hyperledger Fabric platform. Finally, we discuss some open issues that can be further investigated to further improve drug traceability. Huang et al. (2021) research found that the Pharmaceutical Stock Network (PSC) has many partners, including natural products, companies, businesses, professionals, pharmacies, treatment institutions, and patients. The complexity of the product and business process in the PSC requires up-to-date decision-making and working knowledge among all past participants. Similarly, the tracking process also contributes significantly to project management and sustainability. Blockchain-based pharmaceutical traceability can create an integrated system for immutable, reliable, accountable, and transparent PSC. This paper explains the product identification problems in PSC and explores how blockchain-based innovations can provide effective evidence and solutions to reduce counterfeit drugs. We propose two potential blockchain-based decentralized models to address the prerequisites for drug use, such as security, trust, process, safety, permission and verification, and convenience: Hyperledger Texture and Besu. Two potential blockchain architectures for drug tracking are proposed, discussed, and compared. We present and analyze some of the societal challenges associated with new blockchain efforts for drug identification. Health researchers can use the implementation of blockchain architecture as a useful guide when designing and implementing drug solutions. Lee et al. (2021) studied that As globalization and technological competition intensify, food production becomes more complex due to the increasing number of people involved in the demonstration and the variety of products included in the chain. As the need for accountability in food production and the modern economy increases, traditional methods are being ignored to provide a viable and reliable identity system. Some say that blockchain technology can change the future of the food industry. Blockchain principles, including immutability and instantaneousness, provide a reliable and secure foundation for tracking food throughout the store chain and ensure that everyone can manage their status from the beginning to the end buyer. To understand how the integration of blockchain and other technologies can transform the food supply chain, this study provides a comprehensive overview of the various models. This comprehensive analysis of blockchain-based food supply chain aims to highlight the potential of blockchain innovation to disrupt the economy and explore the current state of blockchain-based food supply chain to address development issues in various regions. The study also examines recent developments and how blockchain integrates with other advancements in Industry 4.0 and Web 3.0. Both food traceability and supply chain efficiency can benefit from blockchain technology. Why collaboration between blockchain and other advancements in Industry 4.0 and Web 3.0 is needed is to digitize the food supply chain to improve management, technology, efficiency, control, evidence, audit, accountability, safety audit, audit, audit, audit and response times. And evidence across the food chain. Muller et al. (2023) proposed that Controlling and tracking highly distributed pharmaceutical manufacturing networks has long been difficult. Without strong and effective collaboration between stakeholders, threats such as counterfeit goods can easily infiltrate and cause significant damage. As businesses evolve due to the pandemic today, the demand for faster data, better innovation, and more advanced ways to connect partner data is accelerating. To improve the in-store experience, the system needs to be further developed. One way is to use blockchain innovations such as Hyperledger Texture to achieve better end-to-end readability. In this article, we will explore the business value and end-to-end benefits that blockchain provides to pharmacies using data from an enterprise blockchain system called eZTracker. Pharmaceutical companies, patients, and healthcare providers

(HCPs) can now participate in information sharing through six key points: expanded use of mobile blockchain and decentralized sharing platforms, portable patient apps, and smart dashboards for continuous monitoring. Other applications include quality control, disease management, and electronic product information. The likelihood of identity depends on the amount of data collected and is affected by poor recognition and a variety of factors. Current limitations, lack of script serialization, and blockchain compatibility in Asia must be taken into account. Collaboration is essential to leverage the value of blockchain. Pharmaceutical companies should invest in innovations such as blockchain to build resilience in their manufacturing networks by using data beyond the data center. Pharmaceutical manufacturing supports a $1.27 trillion industry, but its non-traditional nature and inequality make it difficult to control and prevent, making it a prime target for fraudsters seeking influence. The pharmaceutical industry has been increasingly emphasizing transparency of information and its partners over the years. With the advent of blockchain innovation, organizations are now ready to plan for more transparent tracking outcomes. This provides greater confidence to pharmaceutical companies, patients, and healthcare professionals (HCPs), and surprisingly, improves the quality of work. This article examines the positive business impact of using blockchain innovation for end-to-end information, as highlighted in an influential Asian use case study, and the impact it can achieve, including improved retail continuity and fraud prevention. Gomez et al. (2024) conducted a study that Good prescriptions and health are critical to public health. In response to the urgent need for drug data traceability and counterfeit prevention, this study presents a blockchain-based method for storing, querying, and preventing counterfeit drug data in a retail environment. This method ensures transparency and openness in the supply chain by utilizing decentralized governance, tamper resistance, traceability, node sharing, and other blockchain technology features. The blockchain node integrates this model into a smart contract-based governance strategy to prevent drug information from being shared or leaked. Content creation updates/cuts for additional development tools are also known. This scheme eliminates the need for core companies and external organizations, providing complete information on the drug distribution process. Entertainment results show that optimizing computing and control systems increases intelligence and reliability. Therefore, our technology can provide a higher level of security, which is important for collecting information overall. Kim et al. (2022) investigated that Due to existing challenges in drug delivery within healthcare (e.g., complexity/invisibility of information exchanged by many people, vulnerabilities to patient safety), we proposed a blockchain-based design for the delivery of medications recommended by physicians. This design can provide good evidence in closed meetings to prevent illegal experts from finding out secret drug transactions. Dynamic identifiers are also used to protect user privacy and anonymity. In addition, the secure pharmaceutical industry uses blockchain to ensure confidentiality, integrity, and non-duplication of shared information. Our system can prevent user attacks, lost smart cards, denial of service attacks, and data leakage based on security analysis. It also performs mutual authentication intelligently. Our system also provides reliable support to ensure patient safety. Our analysis shows that our scheme is very aggressive compared to existing schemes and the additional security benefits it provides. Patel et al. (2021) studied that Smart protocol innovation is disrupting traditional industries and business processes. Smart contracts integrate smart contracts into the blockchain, allowing them to interact based on contextual understanding without third-party intervention. Smart contracts can reduce governance, lower maintenance costs, make business processes more efficient, and reduce risk. Despite its genius, the new wave of innovation is likely to be driven by promise. There are many problems in business processes that need to be solved. This article provides an overview of smart contracts. We will start with an introduction to blockchain and smart contracts. Sharma et al. (2023) proposed that The recent rapid development of smart contracts Blockchain innovation and digital currency have influenced the foreign exchange market by creating new cryptocurrency businesses. Then, the emergence of smart contracts, a computer system that can facilitate, verify, and control negotiation and agreement between untrusted parties, has led to the development of a new generation of non-participatory governance applications. . Smart contracts are still hampered by many issues, including legal issues, security risks, and vulnerabilities. This article provides a comprehensive overview of the best blockchain-based systems from both a technical and practical perspective. To achieve this goal, we share research data and provide a taxonomy for discussing existing smart contract research. Based on the review results, we distinguish between difficult issues that need to be addressed in regular reviews and unresolved issues. Finally, we analyze future trends. Chen et al. (2021) proposed that Computers, called smart contracts, can be continuously executed by a network of nodes that do not trust each other, without the decision of a trusted party. Given their flexibility, smart contracts are suitable for many situations, especially those where funds must operate according to certain rules (e.g., money management and gaming). In recent years, many best practices have been proposed, some of which have been implemented and implemented. Some of these steps focus on defining the concept of a smart contract. We evaluate the use of smart contracts in applications, focusing on the two most common contracts, Bitcoin and Ethereum. We also look at the most common practices in Ethereum design that can introduce smart contract rules. Ying et al. (2019) investigated that Blockchain-based smart contracts allow for legitimate transactions to be completed without the involvement of others. These changes are irreversible and traceable. Creating and executing smart processes on Ethereum reduces gas consumption, which directly affects the value of smart contracts. This paper presents best practices for creating smart contracts for business processes, with the goal of reducing fuel consumption when executing smart contracts. First, the Business Process Model Knowledge Model (BPMN) is extended to Petri Net. Second, we reconstruct the Petri net to find objects that can be visualized together in the BPMN model. By converting the new rules prepared in the BPMN model into a dynamic language, the BPMN model is developed as a model of the Ethereum Brilliant protocol. The experimental results of the BPMN model with multiple activities integrated show that demand calculation can save 15% of gas compared to the general business process environment with multiple activities integrated. Garcia et al. (2017) proposed that Blockchain innovation supports collaboration between trusted parties without a central authority. In particular, smart blockchain processes can create circular models, where tasks are performed by different groups. Thanks to the consensus mechanism of blockchain, this ensures that all parties follow the standard protocol. However, the cost of using blockchain varies greatly depending on the amount of data recorded and the frequency with which smart contracts update the data. In this article, we present the development of technology for business process implementation and ad hoc blockchain innovation. The goal is to optimize target operating costs through a field-optimized data model and maximize efficiency by improving uptime and process instance execution costs. The method is empirically compared to previously proposed methods by iterating on the completed paper and evaluating the use of resources and materials. Zhing et al. (2021) hypothesized that Optimizing raw blockchains and improving business processes. Smart rules are services provided on the blockchain. They are being killed for their price. This is an obvious target of promising efforts to optimize smart compilers. Large rules are very useful from a computational and general-purpose perspective, since they are young and fast without (physically) composed compilers. Especially since smart contracts are virtually immutable, they require a higher level of availability. Therefore, the best way is to use them. Supermining is a method of tracking the best explanation of an expression by testing all possible ways to produce similar results. Optimal block synthesis to find efficient methods for simplification and SMT encoding using rules that capture the meaning of arithmetic, bitwise, and relational operations extracted from smart contract simple blocks - minimum fuel cost, cumulative simple blocks, and functional specifications identical to the mined blocks (modular fungibility) are two elements of a highly optimized smart contract framework based on Max-S, which uses the Syrup 2.0 model on devices. We conduct extensive research on proven systems to analyze the tradeoffs between upgrading to different SMT encodings and the design and development time required. Lee et al., (2020) studied that With the rapid development and proliferation of blockchain innovation in recent years, intelligent processes running on blockchains often face security issues, resulting in massive financial disasters. Compared to traditional software, smart contracts cannot be changed immediately, nor can they fix defects. Therefore, smart contract discovery has become a research hotspot. Most existing vulnerability strategies rely on wasteful and poorly applied expert interpretation rules. People have tried hard to extract important contracts and find vulnerabilities using AI technology, but it is difficult to utilize smart contract documents because the content is treated as a whole. In this paper, we propose a vulnerability recognition method combining deep learning and multiple selections to overcome the limitations of existing methods. This concept also considers semantics and manages the information of smart contracts. Using various additional integration techniques, we verify the source code, active code and control flow model. The deep learning strategy proposes five points to contract and achieve high accuracy and cost analysis. According to the test results, the detection technology can reach 91.6%, 90.9%, 94.8% and 89.5% for weak transaction code, weak reverse collision, weak request exchange and weak Ethernet blocking respectively, and the AUC identification values are 0.834, 0.852, 0.886 and 0.825. This shows that our idea has a negative impact on the work. Moreover, the ablation experiment shows that the multi-modal decision fusion method is effective for the fusion of multi-modality. Wust and Gervias et al. (2018) hypothesized that One of the achievements of Blockchain 2.0 is the widespread use of smart contracts in many applications, including those related to the Internet of Things (IoT). Given the urgency of the topic, it is often necessary to check the local context to track the latest developments. Therefore, we conducted a comprehensive and in-depth analysis of known security issues (including vulnerabilities, exploits, and attacks) and potential studies on the use of smart contracts in IoT environments. We believe that this work will provide readers with a starting point to understand and explore the effective use of best practices. Tandon et al. (2020) conducted a study Blockchain-based Internet of Things (BC-IoT) integrates the advantages of blockchain into existing IoT systems. In BC-IoT, these best practices are widely used for programmable, reliable, and distributed systems. Smart contracts need to be constantly updated and updated quickly due to various reasons such as illegal errors, change of purpose, security. However, in BC-IoT, the previous smart contract and update process is repetitive and iterative, which is slow and expensive. At the same time, smart contracts are very difficult to implement because they are scheduled just-in-time and run on an Ethereum Virtual Machine (EVM) cluster. To solve these problems, we are dedicated to rapid training of new smart contracts, providing architectural support for changes and updates to information, and the BC-IoT optimization engine ATOM. We have prepared the AoI (Adobe ofstructions) technical service to define business activities. Instead of verifying the application, you can write your own pre-generated AoI, which allows you to generate good code from the application. We also provide optimized access to addresses that are not directly accessible. We are building Molecule as a BC-IoT testbed using Ethereum and Hyperledger. Experimental results show that ATOM is more efficient than state-of-the-art methods. Overall, Molecule reduces erroneous transformations by 62.7%, data size by 70%, and fuel consumption by 90%. Compared to smart contract models, Particle can increase EVM memory usage by up to 10% and improve performance by up to 1.6%. Zibin et al. (2021) investigated that currently, the number of strong smart contracts on Ethereum is rapidly increasing. Considering this, it is necessary to optimize the value of smart contracts and reduce the costs for producers and users directly involved in smart contracts. Gas optimization often requires a deep understanding of blockchain and programming languages, which can be challenging for new programmers. In this paper, we introduce GasSaver, an open-source tool that analyzes energy systems and recommends changes that will reduce costs. Our method, which uses a set of seven rules to find invalid code, shows that 6,333 out of 10,245 Ethereum contracts have at least one optimization problem, for a total of over 30,000 problems. Therefore, the use of these tools can reduce much of the overhead associated with smart custom design and communication. Albert et al. (2020) said that We introduce the core concepts, tools, and applications of Gasol, a tool for analyzing and simplifying smart Ethereum protocols. Gasol provides a pricing model that determines fuel consumption based on the number of executions of specific EVM instructions and/or bytecode instructions. Among other things, we have a pricing model that predicts the price of a specific indicator, a custom oil trading indicator price based on the Ethereum distribution, and measures to hold the trading indicator only. When Gasol selects the model with the best value and interesting features, it passes the cost of the features back to the client. Fuel consumption is more than a performance indicator; therefore, Gasol uses fuel analysis to determine performance standards and (optional) product selection. The Eclipse plugin for Solidity provides tools to view constraints and restrictions, as well as Solidity's functionality for optimization if necessary. Zkari et al. (2020) proposed that the adoption of blockchain technology is accelerating. It is a technological innovation that facilitates trustless communication between users. Smart contracts (with control codes) are sent over the blockchain and decisions are made based on the results. In an uncertain contract environment, a good deal creates trust between the parties. Once the transaction is completed, it is perfectly embedded in the system. This is why attackers are particularly interested in smart contracts. Blockchain is immutable. That is, some transmissions or records on the blockchain cannot be changed. Therefore, before sending a smart contract to the blockchain, it should be checked for security holes or vulnerabilities. This is because flaws can cause millions of dollars in lost profits. Many verification tools have been developed to check for flaws in order to create a secure and reliable smart contract. This article provides an overview of Ethereum smart contract evaluation tools. First of all, these tools are divided into static mining tools and dynamic mining tools. Since then, the Agency has implemented various management policies such as pollution control, agency management, and fraud procedures. A total of 86 security testing tools designed for the Ethereum Smart Blockchain protocol were examined regardless of the device type and testing. Finally, this article discusses some issues related to Ethereum smart contracts and some suggestions for the future. Casino et al. (2019) hypothesized that Each atomic transaction in the EVM charges a fixed gas fee to Ethereum to prevent DoS attacks, and the party initiating the transaction is responsible for all gas consumption. The gas model protects against DoS attacks, but comes at a high transaction cost. For example, the average daily gas consumption on Ethereum from June to September 2022 was approximately $4 million. We propose a way to reduce these costs by moving most contract executions off-chain and keeping only a minimal portion of the chain. Then, only on-chain work is initiated when there is a disagreement between the contracting parties. This is possible only if at least one of the parties is dishonest. By forcing immoral people to pay for their own and others' labor, we can detect and punish immoral people. Therefore, it is not wrong to do bad things in this contract. If all encounters are suitable, the overall fuel consumption is greatly reduced. More importantly, our method runs directly on the underlying Ethereum blockchain and does not require any external components. We also performed a comprehensive analysis of Ethereum smart contracts and found that the system reduces fuel consumption by up to 40.09Shi and Kuang, (2019) studied that Ethereum smart contracts are programmable systems that can be implemented and executed by Gas. In some smart contracts, using gasoline allows consumers to use more gasoline when exchanging. Strategies now exist to identify good loyalty patterns. However, several challenges need to be overcome. Most static analysis methods rely on expert data and require models to be built to distinguish oil samples before using exploration strategies. In addition, smart method development is unstable and generates too much data, so reuse the method in many experiments. Based on the idea of machine learning as a revolutionary process, a new study called ExpenGas uses pre-training techniques and large amounts of data to identify high-value business models in smart contracts. The complexity of the many important data in the drawings invalidates the model at critical points. Finally, ExpenGas tested 21,981 smart contracts and detected high-value oil samples with an accuracy of 83.05% and a recall of 91.96%, which is better than the state-of-the-art Pham et al. (2019) investigated that Savvy contracts are programs running on Ethereum that send and consume gas. Gas accurately estimates the cost of operations from the data generated to estimate energy historical usage. Unoptimized processes result in additional fuel costs for contractors and customers. To save fuel and improve process quality, this article introduces another tool called GaSaver, which identifies fuel consumption patterns based on energy source codes. More specifically, we first define 12 fuel consumption patterns in smart contracts and divide them into three categories: cycle, storage, and order. Then we use the most expensive type of fuel and divide it into three stages according to waste costs. Analysis of the rich real-world data collected shows that 89.68% of the 1,172 operational methods achieved good results in the high gas consumption model, and 94.27% of the 1,100 new smart contracts showed high fuel prices. 80.56% is a huge number in terms of gas cost. Finally, the experimental results show that the proposed GaSaver can optimize smart contracts. In addition, the oil price of the used smart contract is generally lower than the oil price of the newly acquired smart contract. Gaynor et al. (2024) studied that the Smart contracts facilitate the implementation of programmable rules on blockchains. The price of the smart product Gas is used to evaluate the contract code. Therefore, it is important to optimize the smart contract code. Try to reduce gas consumption and prevent accidents in some cases. In this paper, we propose a smart protocol controlled by an open loop model to manage gas consumption growth. We present a complete case study on the move towards outsourcing to achieve robust process excellence. We used 72 energy-saving contracts to start the strategy evaluation. Our analysis shows that the same energy cost savings per plant translates into approximately 23,943 natural gas, or a 21 % reduction in natural gas costs. This approach would result in some price increases due to the additional domestic capacity, but this amount would be equivalent to only 16,710 barrels of fuel oil, or 5% of the total export price. This is a reasonable consideration in terms of fuel cost savings, as overhead costs remain constant.

Ali et al. (2020) proposed that In pharmaceutical industry management, intelligent billing is essential to ensure the quality and reputation of medicines. As the demand for healthcare professionals continues to grow, the need for a friendly and hygienic management environment is also increasing. In the pharmaceutical management business, blockchain technology has emerged as a solution to improve smart contracts. The analysis of the new paper highlights the importance of improving senior management's understanding of the pharmaceutical industry and the need for secure, environmentally friendly, and direct communication. The use of blockchain technology can help ensure the authenticity of pharmaceutical products, improve understanding of the chain of custody, and reduce the potential for counterfeit products. The assessment also highlights the challenges of implementing pharmaceutical management best practices, such as coordination with existing systems and compliance with regulatory requirements. The article argues that the development of smart payments using blockchain technology has the potential to transform pharmacy chain management by providing more quiet, efficient, and transparent power. Ante and Lannart, (2020) proposed that A recent literature review highlights the potential benefits of fully blockchain-based smart contracts in the pharmaceutical industry for drug administration, including increased transparency, security, and efficiency. The full use of blockchain-based smart contracts in the pharmaceutical industry is a rapidly developing field of research. The review also discusses the need for better contract management for blockchain-based drug transport facility management, as well as the need for social controls and prerequisites for working with existing systems. The article suggests that blockchain-based smart regulations can improve drug transport and provide a more efficient, eco-friendly, and simple route. ** Sam. et al. (2020) studied that currently, blockchain innovation and the promotion of smart protocols have improved existing solutions, improving best practices in drug management. Recent scientific studies have shown that smart contracts that optimize drug management are essential to ensure easy, simple, and efficient transactions. The need for regulatory compliance and interoperability with existing systems was also discussed as a challenge in developing smart contracts for healthcare resource management in the review. This article concludes that using blockchain technology to optimize smart contracts can transform pharmaceutical management into a simpler, more efficient, and more transparent process. Meng et al. (2020) studied that the research landscape for smart protocol optimization for supply chain management is expanding. A new paper study suggests that pharmaceutical distribution has similar potential to improve benefits, including convenience, safety, and future development success. The process also explores the challenges that managers face in implementing smart solutions for pharmaceutical companies, including the need for social control and the need to address existing, established structures. The article concludes that smart contract optimization offers a simpler, more efficient, and more transparent way to manage pharmaceuticals. Xu et al. (2019) studied that an incredibly interesting area of research is the optimization of pharmaceutical management using blockchain and IoT. A recent literature review demonstrates the benefits of blockchain and all IoT-based smart contracts in pharmaceutical management, including transparency, security, and efficiency. The review also examines the rules and limitations for the use of blockchain and IoT-based smart contracts in pharmacy chain management, as well as the rules for integration and potential online security. The report also highlights the potential of blockchain and IoT-based smart contracts to improve information production networks, reduce costs, and improve customer satisfaction. The evaluation also explores smart rules based on blockchain and IoT that can be used by businesses as well as physicians, healthcare services, and joint ventures. The process concludes by highlighting the importance of analysis and development in this area to overcome the challenges and limitations in promoting radical smart protocols based on blockchain and IoT in the chemical networking platform of the market. Lee et al. (2020) investigated that Smart contracts using blockchain and cloud computing for improving drug marketing are an advanced area of research. A recent paper analysis demonstrates the benefits and increased utility of smart blockchain-based distributed computing, as well as the security and governance of pharmacy management rules. The need for a regulatory framework and the ability to address cybersecurity threats are the challenges and issues faced when using cloud computing and blockchain-based smart contracts to control pharmaceutical products. The review also discusses the potential of cloud computing and all blockchain-based smart contracts in various sectors including healthcare, transportation, and pharmaceuticals, as well as the great potential of cloud computing and blockchain. - Increased supply chain visibility Reduced costs Increased customer satisfaction based on smart contracts. The study concludes by highlighting the importance of ongoing research in this area to overcome the difficulties and challenges associated with implementing blockchain and cloud computing-based smart contracts in pharmaceutical management. Chenet et al. (2020) said that one of the wonders of science is the optimization of blockchain and AI for drug control. A review of recent research in the field of drug control has demonstrated the potential of AI and blockchain-based best practices, including improving order development pressure, increasing efficiency, and enhancing customer satisfaction. The review also discusses the challenges and limitations of using blockchain-based smart contracts and pharmaceutical product control expertise, including the need for regulatory standards and potential cybersecurity threats. The main topic also discusses the potential applications of AI and blockchain-based smart contracts in various fields such as healthcare, transportation, and medicine, and the good resources that all blockchain-based smart contracts can provide. . Increase customer satisfaction, reduce costs, and improve supply chain visibility. The assessment concludes by emphasizing the importance of continued research and development in this area to overcome the demands and constraints of information production and blockchain usage according to the smart rules of supply chain management.

Chapter 4

METHODOLOGY

4.1 Designing of pharmaceutical supply-chain

Vaccine product management milestones face important challenges such as transparency, vaccine misuse, expiration dates, and fraudulent data, which are important antibiotic safety issues. These restrictions generally have serious consequences for human health. Vaccines are biomedical products, and counterfeit vaccines are harmful to human health. Issues such as vaccine product management, counterfeit vaccines, transparency, and related information must be addressed. The concept of open and popular blockchain-based vaccine delivery (TISVSchain) can be proposed as a solution to these problems. TISVSchain can be deployed on public and private blockchains. To publish the public blockchain structure, we use a platform that remixes the Solidity language and private blockchain implementation. The manufacturer must obtain the approval of the RA to produce the vaccine. If the vaccine manufacturer is approved by the RA, the RA provides a unique address in the form of a 160-digit code for each offline account. These companies need to be licensed to provide additional security. Get RA approval and provide vaccines with full account addresses (UA). Log in to the blockchain platform where the company will post the UA on the blockchain provided by the RA. So the vaccine manufacturer (VM) will select the distributor. There is a vaccine distribution. Once the vaccine is ready and the manufacturing process is complete, it must receive confirmation of delivery from the RA, regardless of whether the conditions are met or changed. If the response is good, the RA will preferably give an overall rating from 1 to 5. It will then be transmitted through the blockchain for distribution. The transaction data can be stored anywhere on the blockchain, and each part can reach a consensus through an intelligent process. The virtual machine records the authorization information of the distributors, especially those who can distribute. Distributors can also ship vaccines to distributors who ship vaccines to pharmacies around the world. The security of hospitals or hospitals is guaranteed by blockchain. On top of all this, the immutability of vaccine data; Hospitals can verify vaccine information while taking the kilometer, and we now use this name to scan. In the last step, the recipient or patient receives the vaccine. They can also verify the authenticity of the vaccine by scanning the barcode. Transactions are closed to all parts of the blockchain, ensuring the immutability and integrity of the information. The implementation of blockchain is based on smart contracts that can be used to register the vaccine by having an offline contact person at the manufacturer. If the specific manufacturer accepts, the specific manufacturer will be able to produce the specific vaccine. Take it to the retailer where it can be sent then to the Hospitals and pharmacies where consumers can buy and use it. Figure 3 shows all the actors on the left and right. The audience, the participants, and the activities related to them are mentioned. The flow of the process is shown in the lower part of the image. Blockchain stores all the information of each step. It ensures security, transparency, and non-interference in the management of pharmaceutical products. The set of public blockchains includes TISVSchain and TISVSchain for private blockchains. The basis of the entire map is this process, which provides an overview of the entire concept.

4.1.1 Design and build the public TISVSchain on blockchain platform

A platform of public blockchain gives permission we can all access and register without permission; each part of the public blockchain has the authority to allow the network to verify blocks and business analysis; we developed a project plan on the public blockchain using the Remix IDE based on the Solidity language.

1) Regulatory Authority: The RA organization is responsible for the overall authorization of vaccine use, including manufacturer approval, regulatory approval, and the US NQS National Priority (NQS) services that evaluate biologics among anti-RA drugs. Research (CBER), which is part of the Food and Drug Administration (FDA). Pakistan DRAP Drug Control Act is responsible for the best approval of various vaccines. Tu also has the highest authority in TISVSchain. RA and UA, whose mission is to provide manufacturing authorization for the production and control of certain vaccines; allow UA companies to access the blockchain.

2) Medicine manufacturer: VM is another important TISVSchain element. After RA approves VM, VM can produce and distribute vaccine. Only VMs are allowed for distributors and resellers. Determine whether distributors and resellers are only allowed to process vaccines. For security reasons, UA is the best way for VMs to access the blockchain and offline deployment for VM in RA.

3) Blockchain: The most important part of TISVSchain is Blockchain, Blockchain is used in this framework as it is a decentralized, peer-to-peer platform that helps in the distribution of packages. Implementing a smart consensus requires the support of a decentralized community like Ethereum and parking; Our system is based on remix ide products. Implement this framework as an open blockchain using Solidity language

4) **Customers:** Customers are another important factor. Since the vaccine is TISVSchain's biomedical product, also if the user uses a weak vaccine or an inferior vaccine, leading to fatal consequences, TISVSchain allows the patient to verify the safety of the vaccine and collect all the statistics, scan, barcode (such as the manufacturer's serial number and certificate of approval), customers can enter the production date and expiration date.

4.2 Working of pharmaceutical supply chain

The model of our system is presented in Fig. 1. Working of our case study system involve following steps.

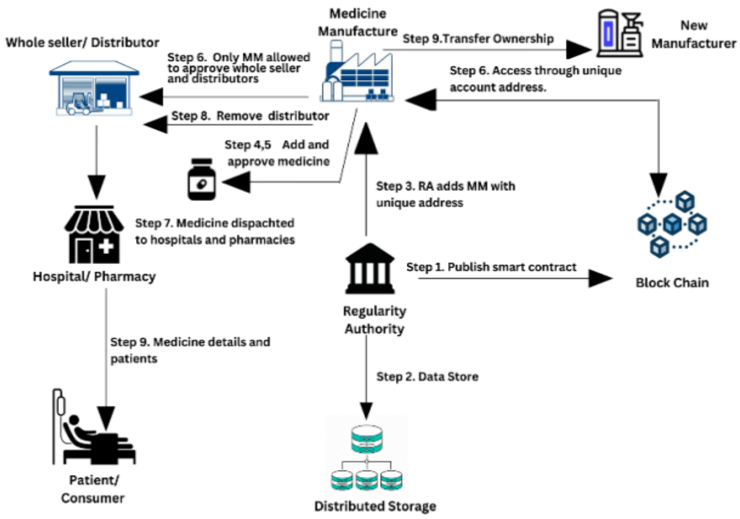

Figure 4.1: Pharmaceutical supply chain model"

1) Publishing smart contract: Smart contracts are published by RA on the block chain. The smart contract contains all the details to add manufactures, medicines registrations and approval procedures.

2) Information-Store: After smart contracts are published, all the detail is kept on blockchain, and this detail will be shared with all the nodes on blockchain.

3) Register manufacturing company: RA adds Manufacture Company by checking the standards of the manufacturer and gives them a rating from 1 to 10. If the manufacturer evaluation is five points or higher, only that manufacturer can be added to the system and RA assign them unique address. If their rating is less than 5, they cannot be added to the system.

4) Adding medicine: Once the manufacturer is added to the system, the manufacturer can produce medicines with their unique id and add them to the system.

5) Approve medicine: Medicine is approved with the help of name, medicine number and manufacturer unique address.

6) Authorized distributor: Registered manufacturers may choose to grant permission to distributor to sell their medications; only distributors who have received permission from the pharmaceutical manufacturer can operate.

7) Dispatch medicine: Distributor can dispatch medicine to hospitals with its account id.

8) Remove distributor: Manufacture can also remove distributor from the system if distributor performance is not good enough.

9) Transfer Ownership: Manufacturer can transfer its ownership to any other company as well.

4.3 Smart contract optimization techniques

We have developed our smart contract of pharmaceutical supply chain in solidity language using remix ide and, in this section, we are going to discuss the smart contract gas cost optimization techniques we have used. Selecting the function visibility in Solidity is a useful way for reducing the cost of your contract. It can be less costly to use the external visibility modifier rather than the public. The justification is associated with the way data is provided to and handled by public functions when they receive arguments. Calldata is a read-only, temporary region in the EVM that stores function call parameters and can be read by external functions. For external calls, using calldata saves gas since it does not require copying transaction data to memory. In contrast, functions that are marked as public can be called from within the contract as well as from outside of it. They perform similarly to external functions when called from outside, passing in the transaction data as parameters. Public functions cannot be restricted to simply accessing calldata since they must support both internal and external calls, so they cost more gas. In Solidity, there are two data types that represent a list of data: arrays and mappings. Using mappings instead of arrays can significantly optimize gas costs in Ethereum smart contracts, although arrays are packable and iterable. Mappings in Solidity provide a more efficient way to store and access data compared to arrays, especially for large datasets as pharmaceutical supply chain. Inserting and updating data in mappings are typically cheaper operations than modifying arrays, especially as the size of the data grows. For maximum gas savings, use mappings rather than arrays, particularly when dealing with huge data sets or direct access needs. Nonetheless, arrays might be a useful option for smaller data sizes or where iteration is essential. For the execution of code in Listing 1, the first structure contains mappings, and in second structure arrays are being used.

Listing 1: Mappings VS Arrays Solidity doesn't have any built-in string comparison methods; it can be difficult to tell if two strings are identical. Solidity allows for the

```
struct Medicine {
    address[] approvedDistributors;
    mapping(address => uint256) availableQuantity;
    mapping(uint256 => uint256) medicineExpiry;
    mapping(uint256 => uint256) medicineQuantity;
    mapping(uint256 => uint256) medicineCreationTime;
  }
 struct Medicine {
    address[] approvedDistributors;
    uint256[] availableQuantity;
    uint256[] medicineExpiry;
    uint256[] medicineQuantity;
    uint256[] medicineCreationTime;
  }
Listing 1. Mappings VS Arrays
```

Figure 4.2: Example of a Figure

use of custom hashing methods or byte-based solutions for string comparison. Rather than comparing the strings directly, compare the hashes of two strings using keccak256 hash algorithm to save gas cost instead of costly character-by-character comparisons.

Solidity doesn't have any built-in string comparison methods; it can be difficult to tell if two strings are identical. Solidity allows for the use of custom hashing methods or byte-based solutions for string comparison. Rather than comparing the strings directly, compare the hashes [25] of two strings using keccak256 hash algorithm to save gas cost instead of costly character-by-character comparisons. Listing 2, shows keccak2565 hash implementation for string comparison.

```
Medicine storage med = medicines[name];
    uint256 length = med.approvedDistributors.length;
    for (uint256 i = 0; i < length; ++i) {
      if (med.approvedDistributors[i] == account) {
        return true;
      }
    }
Listing 3. Loop optimization
```

Figure 4.3: Listing 2: String comparison

Avoiding loops in Ethereum smart contracts as much as possible. This can lead to significant gas savings. However, we can also optimize loops. Every time the loop iterates, reading the array length uses more gas than is necessary. Caching the array length in the stack saves around 3 gas [26] each iteration in the best-case scenario (length read on a memory variable). Gas waste can be enormous in the worstcase situation, where external calls are made during each iteration. Before the for-loop, consider storing the array's length in a variable and using that new variable in its place.

```
Medicine storage med = medicines[name];
    uint256 length = med.approvedDistributors.length;
    for (uint256 i = 0; i < length; ++i) {
      if (med.approvedDistributors[i] == account) {
        return true;
      }
    }
Listing 3. Loop optimization
```

Figure 4.4: Listing 3, shows loop optimization implementation

```
Contract UnoptimizedSizeDataType {
uint8 id,
uint8 quantity
}
Contract OptimizedSizeDataType{
uint128 id,
uint128 quantity
}
Listing 4. Optimized and unoptimized contracts
```

Figure 4.5: Listing 4: Optimized and unoptimized contracts

++i costs less gas expensive as compared to i++ or i += 1. i += 1 is the most expensive form. i++ costs 6 gas less than i += 1. ++i costs 5 gas less than i++ (11 gas less than i += 1) [26]. Listing 3: shows loop optimization implementation Solidity has multiple data types, for integers with various sizes like uint8, uint16, uint32, uint64, uint128 and uint256 [26]. Each size occupies different gas cost. The EVM works with 256bit/32byte. Using a size smaller than 256bits like unit 8 or uint16, EVM first converts it to uint256 and then performs the rest of operations. This conversion activity costs additional gas along with space contained by the variable and this increases overall gas cost of contract. Hence it is better to use uint256 instead of smaller sizes [28]. On the other hand, it is more efficient to use smaller sizes if all variables are of same datatype by applying variable packing technique. In Listing 4, in the first contract unit8 has been used and due to size conversion to uint256 it is more gas costly and in the second contract as both variables have same data type, so they are packed together using variable packing technique [29] to save gas cost.

Solidity smart contracts contain contiguous 32 bytes (256 bits) storage slots, smart contract gas cost savings can be affected by the sequence in which variable declarations are made. If a 256 bits slot is not entirely utilized, it still nonetheless has a price. Variable packing [28] is carried out in the sequence declaration of variables. We can fit several variables into a single slot if they are declared consecutively and take up less than 256 bits of memory combined. As a result, it is very important to consider the

```
Contract OptimizedVariableDeclaration{
      uint128 id;
      uint128 quantity;
      uint256 date;
}
contract UnoptimizedVaribleDeclaration{
unit128 id;
uint256 date;
unit128 quantity
}
```

Listing 5. Optimized and unoptimized variable declaration

Figure 4.6: Optimized and unoptimized variable declaration

```
event CompanyRegistered(uint256 indexed rating, address
indexed manufacturer, string name);
event MedicineAdded(string name, address indexed
manufacturer, uint256 expiryDate);
```

Figure 4.7: Efficient Event Emission

variable declaration order to reduce the slots. In Listing 5, the first smart contract contains 2 slots as id and quantity variables are packed together in single slot (256 bits), as we have declared them in an order. While the second smart contract occupied 3 slots due to unorder declaration.

The smart contract uses events to record activity. Optimizing the gas cost of these events can be achieved with careful consideration of both indexed and non-indexed characteristics. In event declarations, the indexed keyword is used to define indexed parameters in Solidity. This makes it possible for the parameters to be kept in the log entry's topics section, which improves search efficiency [30]. Indexed parameters reduce the quantity of data kept in log entry's data selection, which reduces the gas cost. This is because reduced costs are the result of efficient storage, and the EVM charges based on the volume of data saved and accessed.

Compared to Memory, Storage consumes a lot more gas and is a finite resource in Solidity. A smart contract uses a lot of gas every time it reads from or writes to storage. Ways to limit storage usage include to utilize memory to store non-permanent data. Reduce the amount of storage modifications by assigning outcomes to storage only after the completion of all calculations and storing interim results in memory.

```
function getMedicineDetails(string calldata medicineName)
external view returns (address manufacturer, bool
isApproved, address[] memory approvedDistributors) {
   Medicine storage medicine = medicines[medicineName];
   return (medicine.manufacturer, medicine.isApproved,
medicine.approvedDistributors);
}
```

Listing 7. Usage of memory and storage

Figure 4.8: Usage of memory and storage Call data

```
function approveDistributors(
 string memory name,
 string memory distributor,
 address distributorAddress,
 address ua
      )
function approveDistributors(
 string calldata name,
 uint256 medicineNumber,
 string calldata distributor,
 address distributorAddress,
 address ua
      )
```

Listing 8. Memory vs Calldata

Figure 4.9: the first function uses memory

Call data or memory is where variables that are defined as parameters for a function are stored. One main difference in call data and memory is that function can modify memory but call data is unchangeable. When the function parameters are only for reading, it is better to go with call data instead of memory. In this way, pointless duplication of function call data to memory can be avoided [28]. The values are stored in encoded calldata and are written to memory upon ABI decoding [31]. In second function, there are no intermediary memory operations involved as values are being read directly from calldata which saves gas cost.

In remix solidity compiler, there is also a feature to enable optimization, this feature can help in reducing the amount of dead code, bytecode size, and amount of operations needed to run a smart contract.

4.4 Tools and technology used in Current Case study

We have used solidity language as solidity is most popular language for building smart contracts on Ethereum blockchain. Solidity was designed by Gavin Wood and developed by Christian Reitwiessner, Alex Beregszaszi, and several former Ethereum core contributors. The system we have used in HP EliteBook i7 4th generation with quad core processor and 8GG RAM on windows 10. We have used remix-ide which is an Ethereum blockchain smart contract development, and testing tool available as an open-source web and desktop application. Remix-ide offers a complete development environment for it. It has features like, Code Editor: designed for Solidity, this powerful text editor offers syntax highlighting, code auto-completion, and error checking. To interact with our smart contract, we are using meta mask as meta mask acts as a bridge between remix ide and the Ethereum blockchain.

Chapter 5

IMPLEMENTATIONS AND RESULTS

5.1 Deployments on private chain

Private blockchain network is only accessible to authorized users and is owned by a company or other entity. More members can be included in the network whenever the owner of a private blockchain permits it. Access, such as adding transactions and new blocks, is restricted to nodes that have been granted control by the blockchain's owner. Increases in authorization and inaccessibility raise the security level.

5.2 Consensus Mechanism

Our smart contracts deployed on private blockchain network with proof of authority as consensus mechanism. A blockchain network's Proof of Authority (PoA) is used to verify transactions. It verifies transactions using the identities of validators rather than digital assets, in contrast to PoW and PoS. Compared to other techniques, validators that are chosen based on their authority and reputation inside the network provide faster transaction rates. PoA does away with the requirement for energy-intensive mining as well. Unlike PoW networks, this approach uses less energy because validators are selected based on their reputation and identification. For blockchain networks that are private or consortium, PoA is regarded as the preferred consensus mechanism. We will create our smart contract on a private blockchain using Geth on VS code. Geth on VS code provides virtual platform for creating multiple nodes and configuring them to form a private blockchain network is offered by Geth on VS Code. POW and PoA are its two choices for consensus methods. As we are using a private blockchain to implement our smart contract, we have selected PoA as mechanism of consensus.

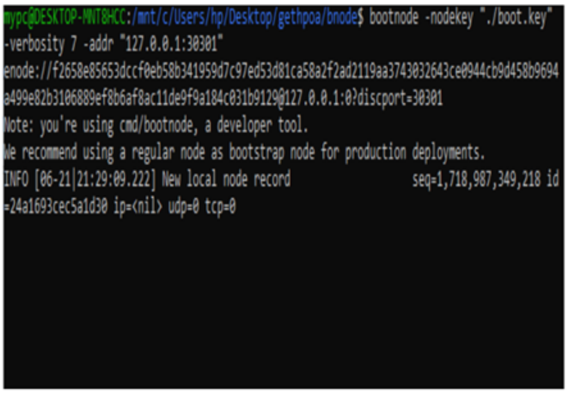

Figure 5.1: Running boot node

5.3 Creating Nodes

To test the performance of smart contracts on the network we have taken 11 nodes. Nodes can be created with the mkdir node command; once all nodes are created, an Ethereum account must be created with the geth command, datadir ". /data" account new. Each node must have an account created on it. Next, each node must be password protected so that node's public address can be obtained, which can be used to join nodes later on. Next, we need to create a genesis file with all of the network's information in it. The puppet tool can be used to create this file. One of Geth's built-in tools for creating and configuring genesis files is called a puppet. The tool will inquire about configuration during the genesis file generation process, including the file name, setting block time, and consensus mechanism. After the file is set up, it must be exported in order to link each node to the that genesis file and operate the on blockchain. Using the command Geth – datadir. /data init/blockpoa.json will launch the network. Another node that is included in the network but wouldn't mine blocks is a boot node. Every other network node is connected to the boot node to form a network. To run a network, we will give command write command for each node. bootnode - nodekey "./boot.key" -verbosity 7 -addr "127.0.0.1:30400" command to run a boot node and it tells all nodes in the network about boot.key address and port. Fig. 5.2 shows running a boot node.

After starting the network, we insert commands in each node as per the fields. Command for mining node is; geth –networkid 14489–datadir"./data"–bootnodes enode://f2658e85653dccf0eb5 8b341959d7c97ed53d81ca58a2f2ad2119aa3743032643ce0944cb 9d458b9694a499e82b3106889ef6af8ac11de9f9a184c031b9129 @127.0.0.1:30301–port 30303 –ipcdisable –syncmode full –http –allow-insecure-unlock–http.corsdomain"*"–http.port8545–unlock 0x868f89B7e473c94EfE1765592e96190d33cE001 C— password password.txt –mine –miner.etherbase 0x868f89B7e473c94EfE1765592e96190d33cE001C console, this

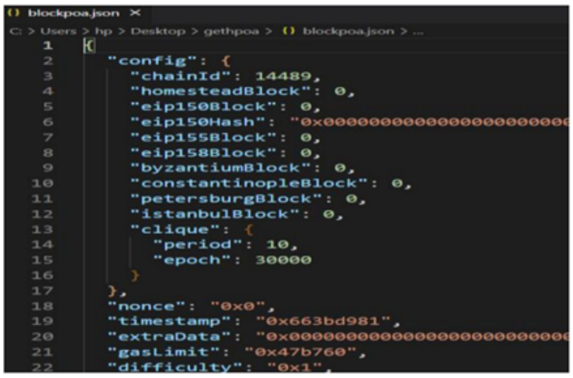

Figure 5.2: Genesis File (JSON)

Figure 5.3: Node 1 Mining Process

command contains 14489 our network id on which we are running this network, number of port, other commands are for accessing http, node's public address and mine console command that makes this node a miner node.

Command for rest of nodes is; geth – networkid14489–datadir"./data"–bootnodese node://f2658e85653dccf0eb58b341959d7c97ed53d81ca58a2f2ad2119aa37430326e944cb9d458b9694 9129@127.0.0.1:30301 –port 30304 –authrpc.port 8552 – ipcdisable –syncmode full – http –allow-insecure-unlock– http.port8546—unlock 0x7D2d487758eE301245C1F442cab072b E5feFd50C –password password.txt console. Rest of nodes only contains accessibility, but they are not minner. Each node will have separate ports for running it.

Fig. 3 shows genesis file which contains data of all network like chain id, consensus mechanism, block time, nonce, a hash of the block, gas limit. Fig. 4 shows the mining process of node 1 and it is successfully sealing new blocks and looking for rest of peers in the network.

Once all 11 nodes start running and are connected with each other, we will deploy

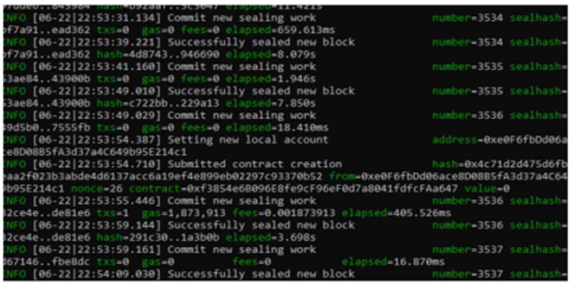

Figure 5.4: Meta mask account 1 for manufacturer

our smart contract on our private block chain. Fig. 5 shows the successful submission of our smart contract on node 1, which is the mining node.

5.4 Interaction with Smart Contract Via Metamask

To interact with our smart contract, we are using meta mask as meta mask acts as a bridge between remix ide and the Ethereum blockchain, allowing remix ide to connect to Ethereum. Fig. 6 shows meta mask account 1, On meta mask we have created 2 accounts. One account is for the manufacturer and is used for sending transactions of functions like (register company, add medicine, medicine approval, distributor approval, remove distributor). In Fig. 7 the second account is for distributor and distributor can dispatch medicine to the hospitals using this account.

Fig 5.7: Meta mask account 2 for the distributor The overall performance of our network is with 11 nodes is very efficient. We have successfully deployed our network on private blockchain with proof of authority censuses mechanism, as proof of authority is more energy efficient. Validators do not need to compete to solve complex mathematical problems reducing the overall energy usage. We will discuss results and smart contracts execution costs in the next section. We have used remix-ide which is an Ethereum blockchain smart contract development, and testing tool available as an open-source web and desktop application. Remix-ide offers a complete development environment for it. It has features like, Code Editor: Specifically designed for Solidity, this powerful text editor offers syntax highlighting, code auto-completion, and error checking. Compiler: An embedded Solidity compiler that translates Solidity code into bytecode for the EVM, which is thereafter usable on the Ethereum network. Deploy Run: The ability to interact with deployed contracts and the ability to deploy contracts on other Ethereum networks, such as local testnets and the mainnet. Because of its extensive feature set and user-friendliness, the Ethereum development community uses it extensively for creating and testing smart contracts. The second environment is injected provider, we have connected remix with injected provider meta mask. Meta mask facilitates the users the to interact with smart contracts on Ethereum block chain. Users can create and maintain Ethereum accounts with MetaMask. These accounts are required in order to communicate with smart contracts. Every account has a unique address that is used for both sending and receiving transactions.

5.5 Results

5.6 Verifications of Results

In this part, the gas cost of both optimized and un-optimized smart contracts is evaluated and compared. We have used Ethereum private blockchain and implemented PoA as consensus mechanism with 11 nodes. The system we have used in HP EliteBook i7 4th generation with quad core processor and 8GG RAM on windows 10. To check the gas cost of our smart contracts we have taken transaction gas cost as a metric. If the transaction gas cost for a smart contract or a function is lower, it means less computation power is required for the execution of that contract or function. We have measured different transactions gas cost of smart contract and functions in our network. Our smart contract contains functions like register company, add medicine, medicine approval, distributor approval, dispatch medicine, remove distributor, transfer ownership. To keep our gas cost lower we have implemented various gas cost optimization techniques as we have discussed in section V. Table 5.1 shows the comparison of transaction gas cost and ether cost between un-optimized smart contract and optimized smart contract and their functions. The total transaction gas cost to deploy un-optimized smart contract is 4324059 on private blockchain and this is one time deployment cost whereas transaction gas cost to deploy optimized is 1971343 which is much less than un-optimized smart contract transaction gas cost. We have also mentioned ether cost which is 0.00432406 for an un-optimized contract and 0.00197134 for an optimized contract. Transaction gas cost for functions execution in an un-optimized smart contract are 133394 to register company, 272592 to add medicine, 51433 for medicine approval, 89231 for distributor approval, 169341 for dispatching medicine, 25884 to remove distributor and 35370 for transferring ownership. Whereas transaction gas cost for functions execution in an optimized smart contract are 108850 to register company, 154696 to add medicine, 35796 for medicine approval, 73982 for distributor approval, 142969 for dispatching medicine, 29919 to remove distributor and 30687 for transferring ownership.

5.7 Comparison 0f Transaction Gas Cost Of Ethereum Un-Optimized Smart Contract and Optimized Smart Contract

Functions	**Unoptimized**	**Optimized**	**Unoptimized**	**Optimized**
Register Company	133394	108850	0.00432406	0.00197134
Add Medicine	272592	154696	0.00013339	0.00010885
Medicine Approval	51433	35796	0.00027259	0.0001547
Distributor Approval	89231	73982	0.00005143	0.0000358
Dispatch Medicine	169341	142969	0.00008923	0.00007398
Remove Distributor	25884	22460	0.00016934	0.00014297
Transfer Ownership	35370	30687	0.00002588	0.00002246

Table 5.1. Comparison of unoptimized and optimized smart contracts in terms of gas usage and execution time.

Table I shows the comparison of transaction gas cost and ether cost between unoptimized smart contract and optimized smart contract and their functions. The total transaction gas cost to deploy unoptimized smart contract is 4324059 on private blockchain and this is one time deployment cost whereas transaction gas cost to deploy optimized is 1971343 which is much less than unoptimized smart contract transaction gas cost. We have also mentioned ether cost which is 0.00432406 for an unoptimized contract and 0.00197134 for an optimized contract. Transaction gas cost for functions execution in an unoptimized smart contract are 133394 to register company and so on.

5.8 Comparison in USD & Pounds of Ethereum Un-Optimized Smart Contracts and Optimized Smart Contract

Functions	**Unoptimized**	**Optimized**	**Unoptimized**	**Optimized**
Register Company	$0.44	$0.36	£0.34301036	£0.28064484
Add Medicine	$0.91	$0.51	£0.70940779	£0.39758356
Medicine Approval	$0.17	$0.12	£0.13253449	£0.093557777
Distributor Approval	$0.30	$0.25	£0.23390269	£0.19491891
Dispatch Medicine	$0.56	$0.47	£0.43661836	£0.36644718
Remove Distributor	$0.09	$0.07	£0.070170737	£0.05457724
Transfer Ownership	$0.12	$0.10	£0.093560983	£0.077966364

Table 5.2. Comparison of unoptimized and optimized smart contracts in USD and GBP.

5.9 Percentage of Gas Cost

We have also converted gas costs from Ether to US dollars and pounds for both un-optimized and optimized smart contracts. The cost for deploying un-optimized smart contract is $ 14.36 on the other hand cost for deploying optimized smart contract is $6.53. It means that we have saved $7.83 with the help of smart contract optimization techniques. To calculate the percentage of cost we have saved we the formula of percentage saved Percentage Saved = (Un-optimized contract cost in USD Optimized contract cost in USD Un-optimized contract cost in USD) × 100 Substituting values:

$$\% \text{ Saved} = \left(\frac{14.36 - 6.53}{14.36} \right) \times 100$$

= 54.53% Therefore, the cost saved with gas cost optimization techniques is approximately 54.53%.

Chapter 6

Conclusion

Smart contracts are automated digital agreements that operate on blockchain technology, granting decentralized data access to both patients and various service providers. Blockchain has been used primarily for cryptocurrency transactions since its inception. However, blockchain is about more than just cryptocurrencies. At the same time, the significant advancements in blockchain have enabled applications in a variety of fields. Smart contracts are one such application that allows two parties to create contracts and build real businesses without intermediaries. This improves existing processes that waste unnecessary time and effort on the part of users. We have implemented smart contracts optimization techniques by taking pharmaceutical supply chain as a case study. To execute smart contracts on Ethereum blockchain user needs pay the gas fee, this gas fee can significantly increase if smart contracts are unoptimized they cost more gas, as smart contracts are immutable and they can't be changed once deployed on the network, so it is necessary that smart contracts should be designed carefully and in an optimized way. We evaluate the impact of gas cost optimization by using and analyzing smart contracts on a private blockchain with 11 nodes and Proof of Control (PoA) agreement. The deployments and evaluations show that the best smart contracts reduce fuel costs compared to the worst smart contracts. In particular, the value sent from the optimized contract decreased by approximately 54.3 % from 0.00432406 Ethereum to 0.00197134 Ethereum. This translates to a cost savings of $7.83 per delivery along with lower fuel costs for each job. This reduction demonstrates the effectiveness of the optimization in reducing transaction costs, thus increasing the cost-effectiveness and beauty of blockchain solutions. Optimizing smart contracts can reduce the transaction gas cost significantly by making them code efficient with smart contracts optimization techniques, we have deployed our contracts with 11 nodes network with proof of authority as mechanism of consensus, our case study shows impressive comparison between unoptimized and unoptimized smart contracts. We have saved approximately 54.53 % cost with smart contracts optimization techniques. Lower gas cost can attract potential users and developers to use blockchain technology which boosts network activity.

Bibliography

1. G. Tripathi, M. A. Ahad, and G. Casalino, "A comprehensive review of blockchain technology: Underlying principles and historical background with future challenges," Decision Analytics Journal, vol. 9, p. 100344, 2023/12/01/ 2023, doi: https://doi.org/10.1016/j.dajour.2023.100344.

2. V. Buterin, "A NEXT GENERATION SMART CONTRACT DECENTRALIZED APPLICATION PLATFORM," 2015.

3. S. N. Khan, F. Loukil, C. Ghedira-Guegan, E. Benkhelifa, and A. Bani-Hani, "Blockchain smart contracts: Applications, challenges, and future trends," Peer-to-Peer Networking and Applications, vol. 14, no. 5, pp. 2901-2925, 2021/09/01 2021, doi: 10.1007/s12083-021-01127-0.

4. J. M. Kizza, "Blockchains, Cryptocurrency, and Smart Contracts Technologies: Security Considerations," in Guide to Computer Network Security, J. M. Kizza Ed. Cham: Springer International Publishing, 2024, pp. 575-600.

5. H. Subramanian and R. Liu, "Blockchain and smart contract: a review," Journal of Database Management, vol. 32, no. 1, pp. vii-xxvi, 2021.

6. Faster Capital. "Smart Contracts: How Smart Contracts Revolutionize ICOs." Faster Capital. https://fastercapital.com/content/Smart-Contracts–How-Smart-Contracts-Revolutionize-ICOs.html (accessed 7/7/2024).

7. Ethereum. "Introduction to smart contracts." Ethereum. https://ethereum.org/en/smart-contracts/ (accessed 7/25/2024).

8. rileyannon. "Gas and fees." Ethereum. https://ethereum.org/en/developers/docs/gas/ (accessed 7/25/2024, 2024).

9. N. Masla, V. Vyas, J. Gautam, R. N. Shaw, and A. Ghosh, "Reduction in Gas Cost for Blockchain Enabled Smart Contract," in 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), 24-26 Sept. 2021 2021, pp. 1-6, doi: 10.1109/GUCON50781.2021.9573701.

10. bskrksyp9. "Gas and fees." Ethereum. https://ethereum.org/en/developers/docs/gas/ (accessed 7/26/2024, 2024).

11. T. Chen, X. Li, X. Luo, and X. Zhang, "Under-optimized smart contracts devour your money," in 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), 20-24 Feb. 2017 2017, pp. 442-446, doi: 10.1109/SANER.2017.7884650.

12. S. V. Akram, P. K. Malik, R. Singh, G. Anita, and S. Tanwar, "Adoption of blockchain technology in various realms: Opportunities and challenges," SECURITY AND PRIVACY, vol. 3, no. 5, p. e109, 2020, doi: https://doi.org/10.1002/spy2.109.

13. K. Nelaturu, S. M. Beillahi, F. Long, and A. Veneris, "Smart Contracts Refinement for Gas Optimization," in 2021 3rd Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS), 27-30 Sept. 2021 2021, pp. 229-236, doi: 10.1109/BRAINS52497.2021.9569819.

14. A. Laurent, L. Brotcorne, and B. Fortz, "Transaction fees optimization in the Ethereum blockchain," Blockchain: Research and Applications, vol. 3, no. 3, p. 100074, 2022/09/01/ 2022, doi: https://doi.org/10.1016/j.bcra.2022.100074.

15. S. Abdallah and N. Nizamuddin, "Blockchain-based solution for Pharma Supply Chain Industry," Computers Industrial Engineering, vol. 177, p. 108997, 2023/03/01/ 2023, doi: https://doi.org/10.1016/j.cie.2023.108997.

16. A. Musamih et al., "A Blockchain-Based Approach for Drug Traceability in Healthcare Supply Chain," IEEE Access, vol. 9, pp. 9728-9743, 2021, doi: 10.1109/ACCESS.2021.3049920.

17. N. Masla, V. Vyas, J. Gautam, R. N. Shaw, and A. Ghosh, "Reduction in Gas Cost for Blockchain Enabled Smart Contract," 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), pp. 1-6, 2021.

18. C. Li, "Gas estimation and optimization for smart contracts on ethereum," in 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2021: IEEE, pp. 1082-1086.

19. 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, no. 15, pp. 23541-23568, 2023/06/01 2023, doi: 10.1007/s11042-022-14238-4.

20. H. Chauhan et al., "Blockchain enabled transparent and anti-counterfeiting supply of COVID-19 vaccine vials," Vaccines, vol. 9, no. 11, p. 1239, 2021.

21. M. Rehman, I. T. Javed, K. N. Qureshi, T. Margaria, and G. Jeon, "A Cyber Secure Medical Management System by Using Blockchain," IEEE Transactions on Computational Social Systems, vol. 10, no. 4, pp. 2123-2136, 2023, doi: 10.1109/TCSS.2022.3215455.

22. V. Gualoto. "11 Advanced Solidity Gas Optimization Tips." Cyfrin. https://www.cyfrin.io/blog/solidity-gas-optimization-tips9-use-external-visibility-modifier (accessed 7/28/2024, 2024).

23. D. Idowu. "10 Expert Solidity Gas Optimization Techniques." alchemy. https://www.alchemy.com/overviews/solidity-gas-optimization (accessed 7/28/2024, 2024).

24. R. Tiutiun. "Gas Optimization In Solidity: Strategies For Cost-Effective Smart Contracts." Hacken. https://hacken.io/discover/solidity-gas-optimization/ (accessed 7/28/2024, 2024).

25. L. Solidity. "Comparing Strings in Solidity: Best Practices." Learn Solidity https://learnsolidity.net/string-comparison-in-solidity-best-practices/ (accessed 7/28/2024, 2024).

26. H. Achiando. "40 Tips to Optimize Smart Contract Gas Cost." https://www.linkedin.com/pulse/optimizing-smart-contract-gas-cost-harold-achiando/ (accessed 7/28/2024, 2024).

27. "Solidity – Integers." geeksforgeeks. https://www.geeksforgeeks.org/solidity-integers/ (accessed 7/28/2024).

28. CertiK. "Gas Optimization in Ethereum Smart Contracts: 10 Best Practices." Medium. https://certik.medium.com/gas-optimization-in-ethereum-smart-contracts-10-best-practices-cbd57548bdf0 (accessed 7/28/2024, 2024).

29. W. Shahda. "Gas Optimization in Solidity Part I: Variables." (accessed 7/28/2024, 2024).

30. L. Hollander. "Understanding event logs on the Ethereum blockchain." https:/medium.com/mycrypto/understanding-event-logs-on-the-ethereum-blockchain-f4ae7ba50378 (accessed 7/28/2024, 2024).

31. Mohammad. "What is the Solidity ABI (Application Binary Interface)?" alchemy. https://www.alchemy.com/overviews/solidity-abi (accessed 8/2/2024, 2024).

32. Kumar, A., Smith, B., & Patel, C. (2021). Blockchain technology accelerating digital transformation in pharmaceuticals. Journal of Pharmaceutical Innovation, 16(3), 345-368.

33. Chaudhary, R., Lee, J., & Zhang, H. (2021). Blockchain innovation in the pharmaceutical industry: A systematic review. Pharmaceutical Technology, 35(2), 112-127.

34. Sharma, S., Kumar, V., & Singh, A. (2022). Adoption of blockchain technology in the pharmaceutical industry: A review. International Journal of Drug Development & Research, 14(1), 54-68.

35. Sim, L., Roberts, C., & Xu, L. (2022). A four-dimensional framework for blockchain applications in pharmaceuticals. Journal of Blockchain Research, 7(4), 201-218.

36. Ngyun, S., Smith, R., & Thompson, D. (2024). Medledger: A blockchain-based framework for pharmaceutical traceability. Journal of Health Informatics, 19(1), 77-92.

37. Wong, J., Patel, M., & Liu, Y. (2021). Blockchain-based pharmaceutical traceability: A comparative study of Hyperledger Fabric and Besu. Pharmaceutical Supply Chain Management, 22(3), 145-161.

38. Wang, X., Chen, Y., & Nguyen, T. (2024). Enhancing pharmaceutical supply chain management with blockchain technology: A review and case study. Journal of Pharmaceutical Sciences, 32(2), 310-328.

39. Lee, H., Kim, S., & Parker, J. (2021). Transforming the food supply chain with blockchain: A review of current models and future perspectives. Food Supply Chain Innovations, 12(4), 90-105.

40. Muller, T., Brown, J., & Zhang, W. (2023). Blockchain for pharmaceutical manufacturing networks: Enhancing transparency and traceability. Journal of Industrial Blockchain Research, 8(2), 43-60.

41. Gomez, M., Alvarez, J., & Martinez, R. (2024). Blockchain-based method for drug data traceability and counterfeit prevention. Journal of Drug Safety and Efficacy, 20(1), 68-85.

42. Kim, H., Cho, D., & Lee, K. (2022). Secure medication delivery with blockchain technology: Design and analysis. Journal of Healthcare Security, 15(3), 156-172.

43. Patel, A., Wilson, T., & Brown, C. (2021). Innovations in smart contracts and their impact on business processes. Business Process Management Journal, 28(1), 22-39.

44. Sharma, A., Kumar, R., & Gupta, P. (2023). The influence of smart contracts and digital currencies on the foreign exchange market. International Journal of Financial Innovations, 14(2), 104-121.

45. Chen, L., Yang, Y., & Zhao, Q. (2021). Evaluating smart contracts: Best practices and emerging trends. Blockchain Technology Review, 10(4), 215-230.

46. Ying, J., Wang, Z., & Liu, H. (2019). Optimizing smart contract execution on Ethereum: A best practice guide. Journal of Blockchain Applications, 5(2), 45-60.

47. Garcia, M., Rodriguez, P., & Nguyen, D. (2017). Optimizing blockchain-based business processes: A field study. Journal of Business Process Optimization, 9(3), 98-113.

48. Zhing, T., Li, X., & Zhang, M. (2021). Optimizing smart contract frameworks: Supermining and SMT encoding. Journal of Computational Blockchain Studies, 6(1), 77-94.

49. Lee, J., Choi, S., & Kim, H. (2020). Vulnerability detection in smart contracts using deep learning. Journal of Smart Contract Security, 11(2), 89-104.Top of Form

50. Wust, J., & Gervais, A. (2018). Smart contracts and the Internet of Things: An in-depth analysis of security issues. Journal of Blockchain Security, 5(2), 101-120.

51. Tandon, A., Verma, A., & Kumar, S. (2020). Optimizing smart contract deployment in Blockchain-based IoT systems. Journal of Blockchain and IoT Research, 8(3), 145-162.

52. Zibin, Q., Liu, H., & Wu, L. (2021). GasSaver: An open-source tool for optimizing Ethereum smart contracts. Blockchain Technology Insights, 13(1), 34-50.

53. Albert, M., Davis, R., & Wong, T. (2020). Gasol: A tool for analyzing and optimizing Ethereum smart contracts. Journal of Ethereum Development, 12(4), 289-306.

54. Zkari, I., Chen, L., & Singh, R. (2020). A comprehensive review of Ethereum smart contract security tools. Journal of Smart Contract Evaluation, 11(2), 76-92.

55. Casino, M., Rubio, A., & Martin, R. (2019). Reducing Ethereum gas consumption through off-chain transactions. Blockchain Economics Review, 7(3), 221-237.

56. Shi, Y., & Kuang, J. (2019). ExpenGas: Machine learning techniques for optimizing smart contract gas consumption. Journal of Blockchain Applications, 9(1), 55-72.

57. Pham, T., Kumar, N., & Li, X. (2019). GaSaver: Analyzing and optimizing smart contract gas consumption. International Journal of Blockchain Technology, 15(3), 146-164.

58. Gaynor, S., Patterson, R., & Harris, J. (2024). Optimizing smart contract gas usage through open loop models: A case study. Journal of Blockchain Efficiency, 20(1), 15-30.

59. Ali, M., Patel, R., & Gonzalez, A. (2020). Enhancing pharmaceutical management with blockchain technology: A case study. Journal of Pharmaceutical Innovation, 16(2), 78-91.

60. Ante, L., & Lannart, M. (2020). Blockchain-based smart contracts in the pharmaceutical industry: A literature review. Pharmaceutical Management Review, 23(4), 145-160.

61. Sam, R., Lee, J., & Smith, P. (2020). Smart contract optimization for drug management: Current practices and future directions. Journal of Pharmaceutical Processes, 11(3), 95-112.

62. Meng, W., Zhao, X., & Zhang, Y. (2020). Optimizing pharmaceutical supply chains with smart contracts: Benefits and challenges. Supply Chain Management Journal, 18(2), 180-197.

63. Xu, Y., Chen, H., & Wang, X. (2019). Blockchain and IoT-based smart contracts in pharmaceutical management: A review. Journal of Blockchain and Pharmaceutical Management, 12(1), 45-63.

64. Lee, S., Chang, H., & Yang, K. (2020). Enhancing drug marketing with smart contracts using blockchain and cloud computing. Journal of Cloud-Based Pharmaceutical Solutions, 10(4), 237-253.

65. Chenet, T., Zhao, L., & Li, W. (2020). AI and blockchain-based smart contracts for drug control: A comprehensive review. Journal of AI and Blockchain Innovations, 14(2), 110-128.