



*Project Report On*

**Smart Contract Based Supply Chain Management**

*Submitted in partial fulfillment of the requirements for the  
award of the degree of*

**Bachelor of Technology**

*in*

***Computer Science and Engineering***

**By**

**Eldho Markose (U2103084)**

**Hrishikesh M Sreenivas (U2103103)**

**J K Yaswanth (U2103105)**

**Jeevan James Mathew (U2103107)**

**Under the guidance of**

**Ms. Jisha Mary Jose**

**Department Of Computer Science and Engineering  
Rajagiri School of Engineering & Technology (Autonomous)  
(Parent University: APJ Abdul Kalam Technological University)**

**Rajagiri Valley, Kakkanad, Kochi, 682039**

**April 2025**

# CERTIFICATE

*This is to certify that the project report entitled "**Smart Contract Based Supply Chain Management**" is a bonafide record of the work done by **J K Yaswanth (U2103105), Eldho Markose (U2103084), Jeevan James Mathew (U2103107) and Hrishikesh M Srinivas (U2103103)**, submitted to Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2024-2025.*

Ms. Jisha Mary Jose  
Project Guide  
Assistant Professor  
Dept. of CSE  
RSET

Ms. Anu Maria Joykutty  
Project Coordinator  
Assistant Professor  
Dept. of CSE  
RSET

Dr. Preetha K. G  
Professor and HoD  
Dept. of Computer Science  
RSET

# ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Rev. Dr. Jaison Paul Mulerikkal** CMI, Principal, RSET, and **Dr. Preetha K.G.**, Head of the Department of Computer Science for providing us with the opportunity to undertake our project, "Smart Contract Based Supply Chain Management".

We are highly indebted to our project coordinators, **Ms. Anu Maria Joykutty.**, Assistant Professor, Project Coordinator, Department of Computer Science, for their valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Ms. Jisha Mary Jose**, Assistant Professor, Department of Computer Science, for her patience and all the priceless advice and wisdom she has shared with us.

Last but not least, We would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

**J K Yaswanth**

**Eldho Markose**

**Jeevan James Mathew**

**Hrishikesh M Srinivas**

# **Abstract**

The rapid globalisation of supply chains has led to increased complexity and a pressing need for enhanced transparency, traceability, and efficiency in managing the flow of goods. Traditional supply chain systems are often plagued by issues such as fraud, lack of visibility, delays, and manual inefficiencies. The integration of blockchain technology and the Internet of Things (IoT) offers a transformative solution to these challenges. This project proposes the development of a Smart Contract-Based Supply Chain Management system, leveraging blockchain's decentralized ledger and IoT's real-time data collection capabilities.

Key features of the proposed solution include real-time tracking of goods, automated smart contract execution, secure and transparent record-keeping, and data analytics for supply chain optimization. The platform offers significant benefits such as enhanced transparency, improved efficiency, and reduced operational costs by eliminating intermediaries and streamlining processes. Also to implement different types of encryption algorithms(RSA or Diffie-Hellman Key Exchange Algorithms) for various types of attacks like Sybil or 51% attack.

# Contents

<b>Acknowledgment</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>List of Abbreviations</b>	<b>vii</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem Definition . . . . .	2
1.3 Scope and Motivation . . . . .	2
1.3.1 Scope . . . . .	2
1.3.2 Motivation . . . . .	3
1.4 Objectives . . . . .	3
1.5 Challenges . . . . .	4
1.6 Assumptions . . . . .	4
1.7 Societal / Industrial Relevance . . . . .	4
1.8 Organization of the Report . . . . .	5
1.9 Chapter Summary . . . . .	5
<b>2 Literature Survey</b>	<b>6</b>
2.1 Blockchain-Based Drug Supply Chain Implementation (2023) [1] . . . . .	6
2.1.1 Methodology . . . . .	6
2.1.2 Advantages . . . . .	9
2.1.3 Disadvantages . . . . .	9
2.1.4 Conclusion . . . . .	9

2.2	Recent Advances in Smart Contracts: A Technical Overview and State of the Art(2020) [2]	9
2.2.1	Overview	9
2.2.2	Methodology	10
2.2.3	Social Applications	10
2.2.4	Smart Contract Structure	10
2.2.5	Advantages	11
2.2.6	Disadvantages	12
2.2.7	Conclusion	12
2.3	ECC-Based Authentication Protocol for RFID [3]	13
2.3.1	Overview	13
2.3.2	Methodology	13
2.3.3	Protocol Details	13
2.3.4	Advantages	16
2.3.5	Disadvantages	16
2.3.6	Conclusion	16
2.4	An IoT and Blockchain-Based Secure and Transparent Supply Chain Management Framework in Smart Cities Using Optimal Queue Model [4]	17
2.4.1	Overview	17
2.4.2	Methodology	17
2.4.3	Advantages	18
2.4.4	Disadvantages	19
2.4.5	Conclusion	20
2.5	Blockchain Private File Storage-Sharing Method Based on IPFS[5]	20
2.5.1	Overview	20
2.5.2	Methodology	21
2.5.3	Advantages	23
2.5.4	Disadvantages	24
2.5.5	Conclusion	25
2.6	Summary and Gaps Identified	25
2.7	Chapter Summary	27

<b>3</b>	<b>System Design</b>	<b>28</b>
3.1	System Architecture . . . . .	28
3.2	Data Flow Diagrams . . . . .	29
3.3	Component Design . . . . .	30
3.4	Tools and Technologies . . . . .	30
3.4.1	Softwares . . . . .	30
3.4.2	Hardware . . . . .	31
3.5	Module Divisions . . . . .	31
3.6	Work Break Down . . . . .	31
3.6.1	JK Yaswanth . . . . .	31
3.6.2	Eldho Markose . . . . .	32
3.6.3	Jeevan James Mathew . . . . .	32
3.6.4	Hrishikesh MS . . . . .	32
3.7	Key Deliverables . . . . .	32
3.8	Project Timeline . . . . .	33
3.9	Chapter Summary . . . . .	33
<b>4</b>	<b>Results and Discussions</b>	<b>35</b>
<b>Chapter 4: Results and Discussions</b>		<b>35</b>
4.1	System Implementation . . . . .	35
4.1.1	Smart Contract Deployment . . . . .	35
4.1.2	User Interface and Workflow . . . . .	35
4.2	Outputs . . . . .	36
4.3	Challenges Encountered . . . . .	40
4.4	Comparative Analysis . . . . .	41
<b>5</b>	<b>Conclusion and Future Enhancements</b>	<b>42</b>
<b>Chapter 5: Conclusions &amp; Future Scope</b>		<b>42</b>
5.1	Conclusions . . . . .	42
5.2	Future Enhancements . . . . .	42
<b>References</b>		<b>44</b>

<b>Appendix A: Presentation</b>	<b>46</b>
<b>Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes</b>	<b>74</b>
<b>Appendix C: CO-PO-PSO Mapping</b>	<b>78</b>



## List of Abbreviations

SCM - Supply Chain Management  
IoT - Internet of Things  
ECC - Elliptic Curve Cryptography  
RFID - Radio Frequency Identification  
ECQV - Elliptic Curve Qu-Vanstone  
TTP - Trusted Third Party  
ECDLP - Elliptic Curve Discrete Logarithm Problem  
ECDHP - Elliptic Curve Diffie-Hellman Problem  
FIFO - First-In, First-Out  
RF-PO - Revised Fitness-Based Political Optimizer  
NDN - Named Data Networking  
IPFS - Inter-Planetary File System  
PKI - Public Key Infrastructure  
FIM - Federated Identity Management  
ABAC - Attribute-Based Access Control  
RBAC - Role-Based Access Control  
PES - Public Emergency Services

## List of Figures

2.1	Architecture diagram showing the Local Blockchain Network, Client Communication, and Front End setup. . . . .	7
2.2	Sequence diagram showcasing interactions between stakeholders and the decentralized application. . . . .	8
2.3	Overview of Smart Contracts . . . . .	11
2.4	Steps and computations involved in Protocol 1 . . . . .	14
2.5	Steps and computations involved in Protocol 2 . . . . .	15
2.6	The interest packet (a) and the data packet (b) . . . . .	21
2.7	NDN forwarding model. . . . .	22
2.8	Data communication and forwarding process. . . . .	23
3.1	System Architecture . . . . .	28
3.2	Data storage communication structure of the model, with components: . .	29
3.3	Basic Blockchain Network with interaction functions between them . . . .	30
3.4	Gantt chart . . . . .	33
4.1	Dapp interface for user registration. . . . .	36
4.2	MetaMask Wallet Confirmation . . . . .	37
4.3	Producers's GUI . . . . .	38
4.4	Buyer's Interface . . . . .	39
4.5	BackEnd Codes . . . . .	40

## List of Tables

2.1	Advantages and Disadvantages of Reviewed Works . . . . .	26
4.1	Comparison between Traditional and Blockchain Based SCM . . . . .	41

# Chapter 1

## Introduction

Blockchain technology has revolutionized industries by establishing decentralized, secure, and transparent systems, and supply chain management is one of the key applications. Traditional supply chains are marked by low visibility, high operating costs, and disputes over fraud cases. Blockchain offers the solution that aids in the development of a shared, immutable ledger, which ultimately enhances the transparency and trust among different stakeholders. Besides that, it incorporates smart contracts, which are self-executing contracts with terms that are codified in the electronic form.

Embedded in code—automates processes such as payments and compliance checks, reduces the dependency on intermediaries, and further enhances efficiency. Together with transparency and traceability, blockchain-based smart contracts are, therefore a transformative solution for modern SCM challenges.

### 1.1 Background

Supply chain management (SCM) is a cornerstone of global trade and commerce, ensuring the efficient flow of goods and services from suppliers to end consumers. It involves the complex coordination of multiple stakeholders, including manufacturers, logistics providers, and retailers. However, traditional supply chains often operate on centralized systems, leading to inefficiencies such as delays, errors, and limited traceability. Data silos between stakeholders hinder transparency and real-time decision-making, while manual processes, such as verifying shipments and reconciling transactions, are time-consuming and error-prone. These inefficiencies are further amplified by the scale and complexity of global operations, where different regions must adhere to diverse regulations and standards.

Additionally, modern supply chains face significant challenges related to fraud, counterfeiting, and disputes. Industries like pharmaceuticals, electronics, and luxury goods are particularly vulnerable to counterfeit products, resulting in financial losses and eroded consumer trust. Disputes often arise due to the absence of verifiable records regarding the movement, condition, or ownership of goods. Resolving such issues usually involves costly third-party arbitration, which adds to the inefficiencies. Emerging technologies like blockchain and smart contracts present a transformative opportunity to address these problems by enabling decentralized, secure, and automated solutions that enhance transparency, efficiency, and collaboration across supply chain networks.

## **1.2 Problem Definition**

Traditional supply chain management (SCM) systems face numerous challenges that hinder their efficiency and effectiveness. A significant issue is the lack of transparency among stakeholders, which leads to information silos and hampers trust. This problem becomes critical in global supply chains, where the complexity and scale of operations increase the risk of fraud, counterfeiting, and errors. Manual processes and reliance on intermediaries introduce delays, inflate costs, and create bottlenecks in the system.

Additionally, disputes often arise due to a lack of verifiable data regarding the movement and condition of goods. The absence of real-time tracking and traceability makes it difficult to ensure accountability, especially in cases of damaged or lost products. Current systems also struggle to adapt to modern demands for sustainability, regulatory compliance, and efficient resource utilization. These challenges necessitate a robust and scalable solution to optimize supply chain processes and foster collaboration across all stakeholders.

## **1.3 Scope and Motivation**

### **1.3.1 Scope**

This project focuses on designing and implementing a blockchain-based supply chain management system that leverages smart contracts to enhance operational efficiency and stakeholder trust. The scope includes the development of a framework for automating key

supply chain activities such as procurement, tracking, and payments. The system aims to integrate Internet of Things (IoT) devices to enable real-time monitoring and data collection, ensuring improved traceability and accountability across the supply chain.

### 1.3.2 Motivation

The motivation behind this project stems from the pressing need to address inefficiencies in traditional supply chain systems. As industries strive to meet the growing demands for transparency, security, and automation, blockchain technology presents a transformative opportunity. Smart contracts, with their ability to automate agreements and enforce compliance without intermediaries, offer significant potential to streamline supply chain operations. This project seeks to contribute to the advancement of supply chain management by providing a solution that is efficient, secure, and adaptable to diverse industrial and societal needs.

## 1.4 Objectives

- **To Investigate Challenges:** Analyze the current challenges faced by blockchain-based supply chain systems and identify potential solutions.
- **To Enhance Security:** Address security concerns by implementing robust cryptographic measures and decentralized mechanisms to protect data integrity and confidentiality.
- **To Optimize Processes:** Utilize smart contracts to automate supply chain processes, reducing delays and improving efficiency.
- **To Ensure Transparency:** Develop a transparent system that allows stakeholders to access and verify data in real-time.
- **To Evaluate Attacks:** Examine common security attacks on blockchain networks and propose countermeasures to mitigate their impact on supply chain processes.
- **To Provide Recommendations:** Offer recommendations for the implementation of blockchain technology to improve security, transparency, and efficiency in supply chain management.

## **1.5 Challenges**

Off-chain data remains vulnerable to tampering despite the on-chain security of blockchain. Blockchain transactions may introduce delays, affecting real-time supply chain processes. Additionally, scalability challenges arise as the number of participants and transactions grows, potentially slowing down the network. Security threats like 51 percent attacks, Sybil attacks, and smart contract vulnerabilities pose risks to the system's integrity.

## **1.6 Assumptions**

Successful blockchain implementation in the supply chain requires stakeholders to have access to the necessary hardware, software, and reliable internet connections. Data integrity is crucial, with external sources like IoT sensors providing accurate, real-time inputs. The blockchain platform must be scalable to handle increasing transactions, while also ensuring robust cybersecurity to prevent threats like 51 percent attacks and smart contract vulnerabilities. Additionally, there should be no significant legal or regulatory obstacles to adopting blockchain in the target region.

## **1.7 Societal / Industrial Relevance**

The integration of blockchain technology and smart contracts into supply chain management has significant implications for both society and industry. On a societal level, the proposed system enhances consumer trust by ensuring the traceability and authenticity of goods, allowing individuals to make informed purchasing decisions. This is especially critical in industries such as pharmaceuticals and food, where product quality and safety directly impact public health. Additionally, the immutable nature of blockchain records helps combat counterfeit goods, protecting both consumers and businesses from fraudulent products. By promoting real-time monitoring and efficient resource utilization, the system also supports environmentally sustainable practices, reducing waste and optimizing logistics. Furthermore, it empowers small businesses by lowering barriers to entry, enhancing their credibility, and simplifying compliance with industry standards.

From an industrial perspective, the system significantly boosts operational efficiency by automating key processes such as order verification and payment settlement. This

automation minimizes manual errors and reduces delays, ultimately enhancing productivity. The elimination of intermediaries further contributes to cost savings, improving profitability for organizations. Designed to be scalable, the framework accommodates large-scale global operations while remaining adaptable to diverse industry needs. Additionally, the transparent and immutable nature of the blockchain simplifies compliance with regulatory requirements, providing a verifiable audit trail that enhances governance and accountability.

## **1.8 Organization of the Report**

This report is structured to provide a detailed understanding of the development of a blockchain-based supply chain management system. Chapter 1 introduces the background, motivation, and objectives of integrating blockchain and IoT technologies to address traditional supply chain challenges. Chapter 2 reviews existing research on blockchain applications in supply chain management, highlighting the advantages of decentralization, automation through smart contracts, and the gaps in current systems. Chapter 3 explains the system design, covering the implementation of blockchain, IoT, and IPFS technologies, along with the technical methods and tools used. Chapter 4 presents the results, discussing the system's performance in improving transparency, security, and efficiency. Finally, Chapter 5 concludes with a summary of the project's contributions and suggests future directions for enhancing scalability, security, and usability in blockchain-based supply chain systems.

## **1.9 Chapter Summary**

This chapter introduces blockchain technology to address traditional supply chain management (SCM) challenges like limited visibility, fraud, and high costs. It discusses the potential of smart contracts to automate processes and improve efficiency. The chapter outlines the project's objectives, including improving security, optimizing processes, transparency, and addressing challenges like 51 Percentage attacks. It also highlights blockchain's societal and industrial relevance, emphasizing its role in transparency and boosting operational efficiency.



## Chapter 2

### Literature Survey

#### 2.1 Blockchain-Based Drug Supply Chain Implementation (2023) [1]

##### 2.1.1 Methodology

- **Data Architecture:** Designed a decentralized system with three components:
  1. **Local Blockchain Network:** Includes Ganache and Truffle for blockchain simulation and smart contract deployment.
  2. **Client Communication:** Establishes the link between users and blockchain via tools like Web3.js and MetaMask.
  3. **Frontend Interface:** Built with HTML, CSS, and JavaScript for user interaction.
- **Smart Contract Development:** Key functionalities include account creation, medicine registration, ownership transfer, and transaction recording.
- **Implementation:** Blockchain stakeholders (e.g., manufacturers, wholesalers) interact via secure decentralized applications. Actions like adding medicine, tracking ownership, and purchasing involve state changes recorded immutably.
- **Testing and Validation:** Rigorous testing conducted using Ethereum-based tools. Performance measured in terms of gas cost for contract deployments and transactions.

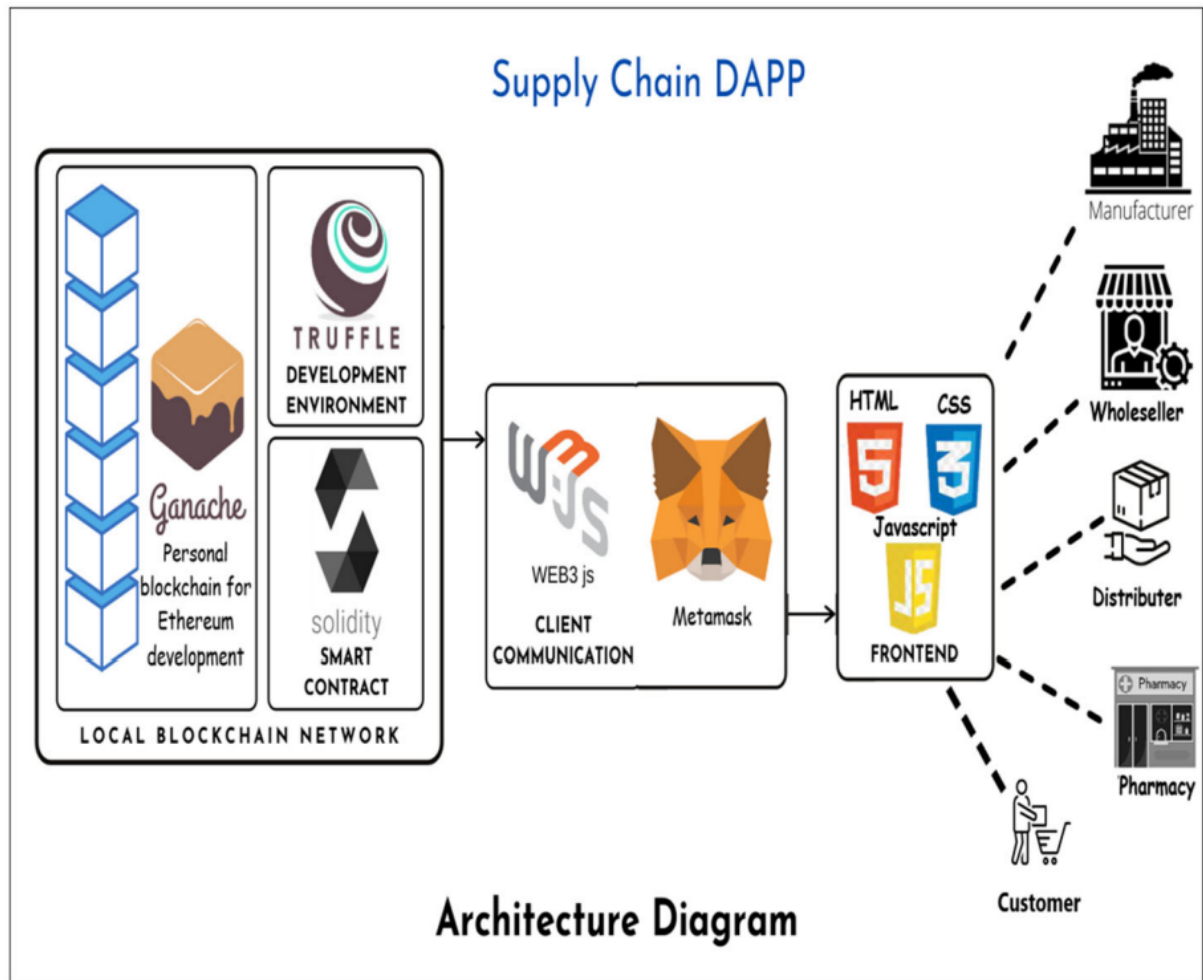


Figure 2.1: Architecture diagram showing the Local Blockchain Network, Client Communication, and Front End setup.

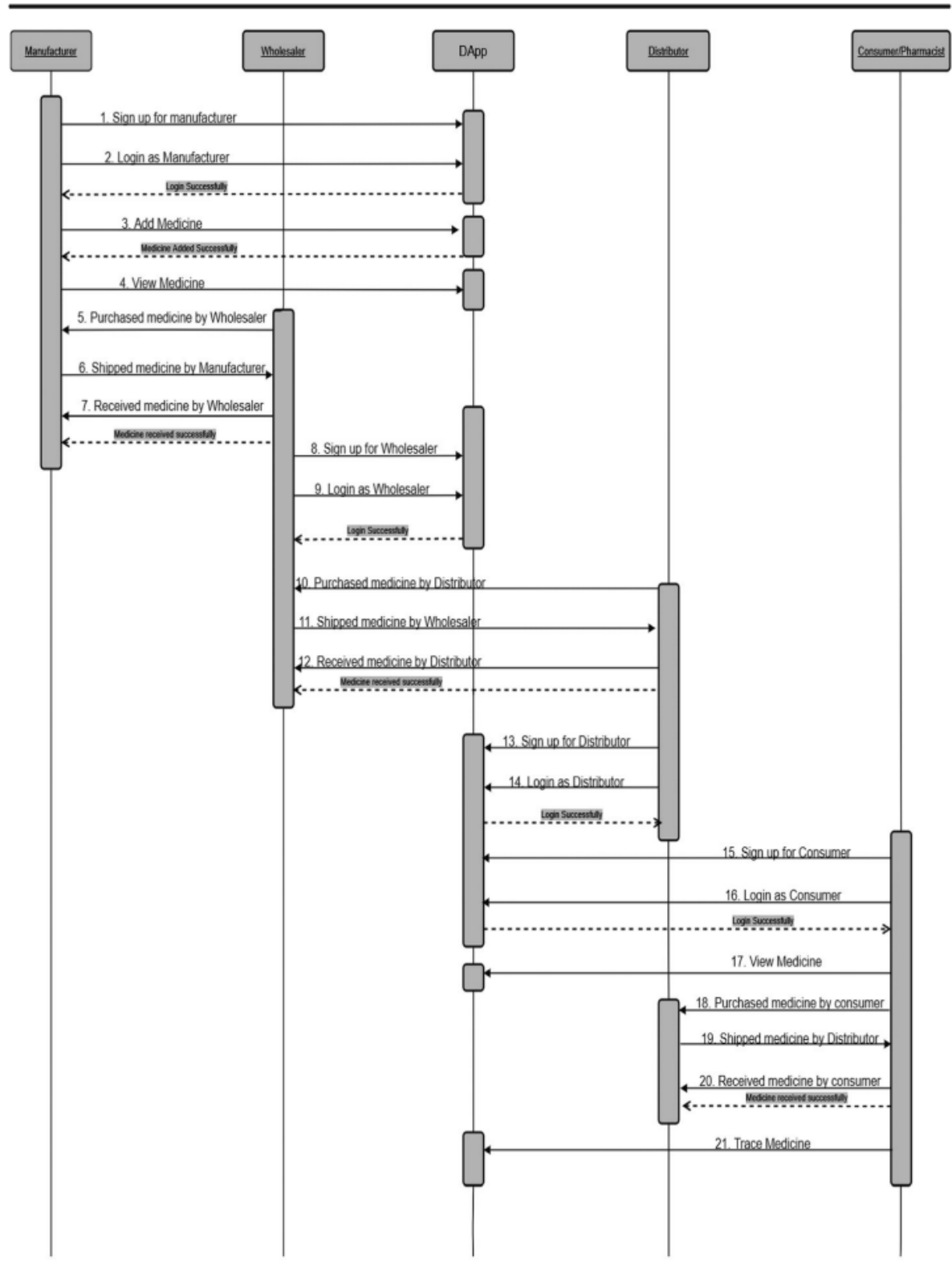


Figure 2.2: Sequence diagram showcasing interactions between stakeholders and the decentralized application.

### **2.1.2 Advantages**

- Enhanced Traceability: Tracks medicine origin and ownership at every stage.
- Improved Security: Blockchain immutability ensures data reliability.
- Efficient Transactions: Automates and secures transactions using smart contracts.

### **2.1.3 Disadvantages**

- Resource Intensity: High computational and storage costs.
- Access Limitations: Dependence on crypto wallets and blockchain understanding.
- Data Integrity Risks: Errors in initial data input can persist without easy correction.

### **2.1.4 Conclusion**

The proposed blockchain model significantly enhances the security and traceability of the pharmaceutical supply chain. However, challenges like computational overhead and public adoption require attention. Future research should integrate IoT for real-time monitoring and address standardization for broader applicability.

## **2.2 Recent Advances in Smart Contracts: A Technical Overview and State of the Art(2020) [2]**

### **2.2.1 Overview**

This paper explores the advancements in smart contract technologies, presenting a comprehensive technical overview and classification into four categories: cryptography, access management, social applications, and smart contract structure. It highlights the role of blockchain platforms as the underlying technology that facilitates trust, transparency, and tamper-resistance for these contracts. The study addresses both the capabilities and challenges of smart contracts in emerging domains.

### 2.2.2 Methodology

#### Access Management

- **IoT Access Management:** Smart contracts allow secure access management for IoT devices in decentralized environments.
- **Federated Identity Management (FIM):** Decentralized systems provide cross-domain identity management independent of central authorities.
- **Attribute-Based Access Control (ABAC):** Utilization of smart contracts to assess attributes and provide dynamic resource access.
- **Role-Based Access Control (RBAC):** Allocation of access rights based on roles, augmented with decentralization for increased security.

### 2.2.3 Social Applications

- **Electoral Systems:** Safe and honest e-voting systems to guarantee electoral integrity.
- **Auctions:** Decentralized marketplaces to provide for fairness and responsibility in bidding procedures.
- **Payment Systems:** Smart contracts facilitate and render secure cashless payments traceable.
- **Energy Distribution:** Peer-to-peer energy trading and distribution handled effectively via smart contracts.

### 2.2.4 Smart Contract Structure

- **Microservices:** Smart contract modular design, with reusability and flexibility.
- **Service-Oriented Computing:** Interoperability of smart contracts with service-based designs for successful operation.
- **Research Frameworks:** Methodologies for crafting and assessing smart contract enhancement.

- **High-Performance Computing:** Techniques to enhance the scalability and efficiency of smart contract execution.

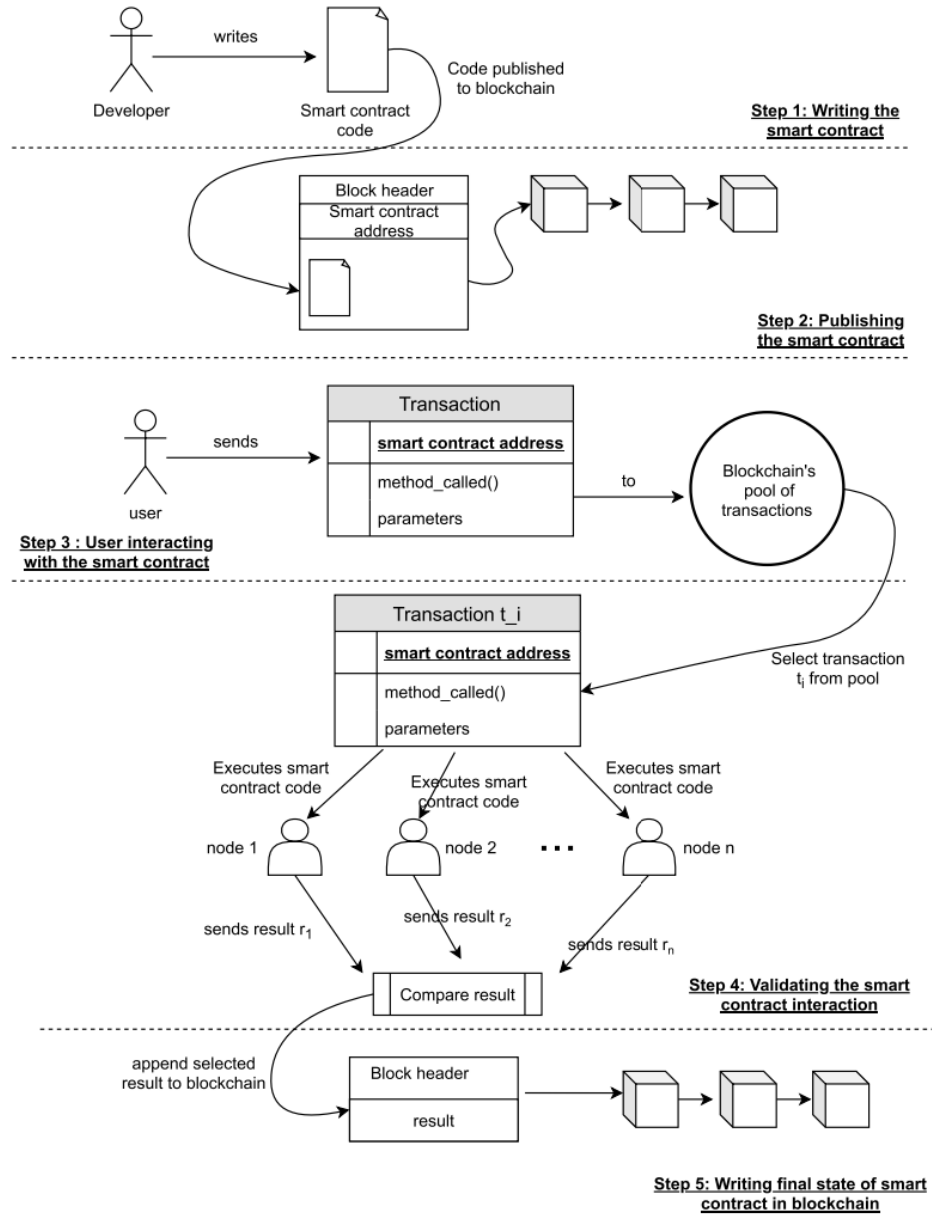


Figure 2.3: Overview of Smart Contracts

### 2.2.5 Advantages

- **Transparency and Trust:** Smart contracts operate on blockchain, ensuring transactions are visible and traceable.

- **Decentralization:** Elimination of central authorities reduces single points of failure.
- **Automation:** Self-executing code streamlines operations and reduces human intervention.
- **Enhanced Security:** Blockchain's cryptographic features secure data and operations.
- **Scalability of Applications:** Wide-ranging use cases from IoT to social domains.

#### 2.2.6 Disadvantages

- **Advanced Development:** Writing secure smart contract code is skill-intensive, and errors can lead to serious vulnerabilities.
- **Immutability Issues:** On platforms like Ethereum, contracts are immutable once deployed, which complicates bug fixes.
- **Scalability and Performance:** Current blockchain platforms are sluggish on transaction speed and quantity in comparison to traditional systems.
- **Cost:** Outrageous computational and transaction fees on mainstream platforms.
- **Limited Privacy:** Public blockchains expose data to all participants, making them unsuitable for sensitive applications.

#### 2.2.7 Conclusion

This research highlights the utility of AI and machine learning to classify smart contracts for improving the security of blockchains. The suggested model holds a lot of promise but needs improvement to adapt more across platforms.

Future development will concentrate on feature set expansion and computational efficiency optimization.

## 2.3 ECC-Based Authentication Protocol for RFID [3]

### 2.3.1 Overview

This paper introduces two highly efficient elliptic curve cryptography (ECC)-based authentication protocols for Radio Frequency Identification (RFID) systems. The protocols offer security solutions to resource-constrained environment challenges while ensuring scalability and resistance to regular attacks.

### 2.3.2 Methodology

- **Architecture:** The protocol is made up of a tag, a reader, and a backend server. ECC enables lightweight, secure communication despite having limited computational power.
- **Key Operations:**
  - **Tags** perform elliptic curve multiplications to encrypt and authenticate information.
  - **Readers** authenticate tags against stored credentials or a revocation list of tags.
- **Protocol Extensions:** Multireader authentication is established using time-based mechanisms and trusted third-party (TTP) signatures to secure initial stages of communication.

### 2.3.3 Protocol Details

#### Protocol 1: ECC-Based Authentication with Storage at Reader Side

##### Initialization Phase:

- The tag receives a unique identity  $id_n$  and keys  $K_{n1}$  and  $K_{n2}$ , along with the public key  $Q_r$  of the reader. These are securely stored on the tag.
- The reader stores the corresponding  $id_n, K_{n1}, K_{n2}$  tuples for all legitimate tags in its database.



### Authentication Phase:

1. **Reader Initialization:** The reader generates a random value  $r_r$  and computes  $R_r = r_r G$ , where  $G$  is the ECC base point.  $R_r$  is sent to the tag.
2. **Tag Response:**
  - The tag generates  $R_n = (r_n + K_{n2})G$ , where  $r_n$  is a random number.
  - It calculates  $H((r_n + K_{n2})(Q_r + R_r))$  and derives parts  $h_{r1}$  and  $h_{r2}$ .
  - The tag encrypts  $K_{n1}$  using  $h_{r1}$  as  $A_1 = K_{n1} \oplus h_{r1}$  and computes a hash  $h_2$ .
  - The message  $R_n, A_1, h_2$  is sent to the reader.
3. **Reader Validation:**
  - Using its private key and  $R_r, R_n$ , the reader computes  $h_{r1}, h_{r2}$ , decrypts  $K_{n1}$ , and verifies  $h_2$ .
  - If valid, the reader sends a confirmation  $h_1$  to the tag.
4. **Tag Confirmation:** The tag compares the received  $h_1$  with its computed value to authenticate the reader.

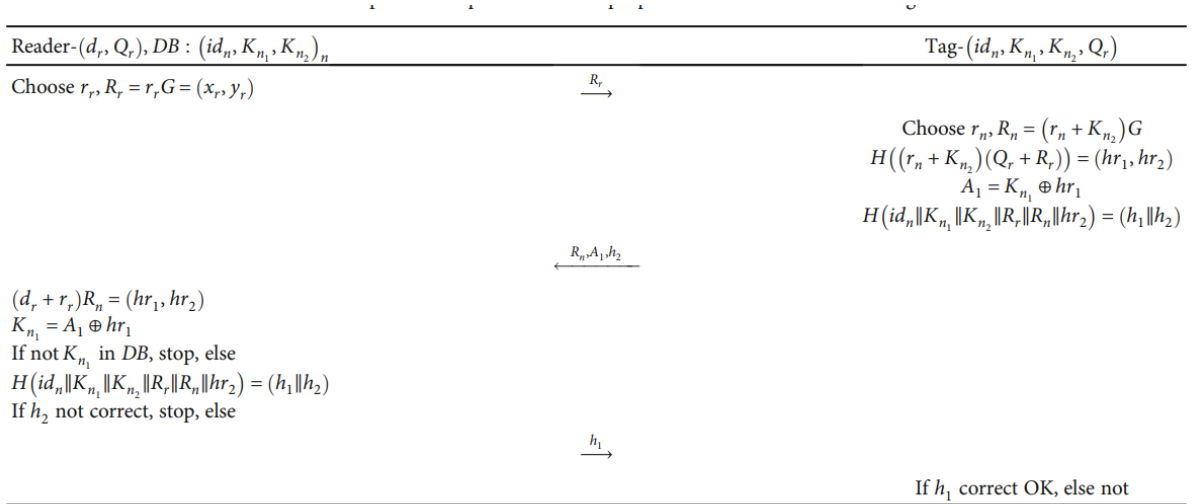


Figure 2.4: Steps and computations involved in Protocol 1

## Protocol 2: ECC-Based Authentication Without Storage at Reader Side

### Initialization Phase:

- The tag uses the ECQV (Elliptic Curve Qu-Vanstone) mechanism to derive a key pair  $d_n, Q_n$  and obtains a certificate  $Cert_n$ .
- The reader stores only the list of revoked tags.

### Authentication Phase:

1. **Reader Initialization:** The reader generates  $R_r = r_r G$  and sends it to the tag.
2. **Tag Response:**
  - The tag computes  $R_n = (r_n + d_n)G$ .
  - A Diffie-Hellman key  $K = (r_n + d_n)(Q_r + R_r)$  is derived.
  - The tag encrypts  $id_n, Cert_n, r$  using  $K$ , computes a hash  $h_2$ , and signs  $R_n$  with its private key  $d_n$ .
  - It sends  $C, R_n, s_n$  (ciphertext, computed value, signature) to the reader.
3. **Reader Validation:**
  - The reader derives  $K$ , decrypts the ciphertext, and verifies the tag's signature.
  - If valid, the reader sends  $h_1$  to confirm authentication.
4. **Tag Confirmation:** The tag checks  $h_1$  to authenticate the reader.

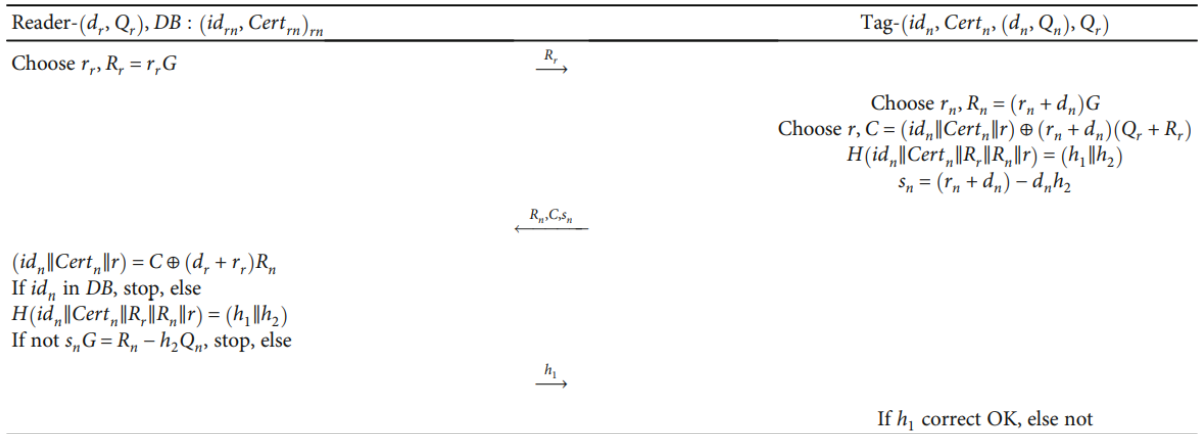


Figure 2.5: Steps and computations involved in Protocol 2

#### **2.3.4 Advantages**

- **Protocol 1:**
  - Minimal computational load on tags, requiring only two ECC multiplications.
  - Simple database storage for readers, containing all valid tag credentials.
- **Protocol 2:**
  - Eliminates the need for large reader-side storage, relying on a list of revoked tags.
  - Supports multireader authentication with additional mechanisms.

#### **2.3.5 Disadvantages**

- **Protocol 1:**
  - Reader scalability is limited by the size of stored credentials.
  - Assumes the trustworthiness of the reader.
- **Protocol 2:**
  - Computationally heavier due to ECQV operations.
  - Requires secure handling of certificates and signatures.

#### **2.3.6 Conclusion**

Both protocols demonstrate strong security, high efficiency, and adaptability. Protocol 1 is suited for scenarios with stable and limited tag-reader pairs, while Protocol 2 excels in scalable, multireader environments. These protocols provide robust solutions for securing RFID systems and can be adapted for diverse applications with further optimizations.

## **2.4 An IoT and Blockchain-Based Secure and Transparent Supply Chain Management Framework in Smart Cities Using Optimal Queue Model [4]**

### **2.4.1 Overview**

This paper, "An IoT and Blockchain-Based Secure and Transparent Supply Chain Management Framework in Smart Cities Using Optimal Queue Model," presents a new paradigm for addressing inefficiencies inherent in traditional supply chain management (SCM). Through the integration of IoT for real-time data gathering, blockchain for secure and transparent record-keeping, and an optimal queue model to minimize latency, the framework significantly enhances SCM operations in smart cities. The system includes a Revised Fitness-based Political Optimizer (RF-PO) to optimize such important parameters like queue length and resource usage for efficient service request processing and better management of scalability problems.

### **2.4.2 Methodology**

#### **1. IoT-Based Data Collection**

- IoT is utilized to obtain real-time information, such as environmental factors like temperature, humidity, and smoke intensity.
- Secure preprocessed data is sent to edge servers for analysis and storage, facilitating quick decision-making in emergencies such as fire brigades or other public emergency services (PES).

#### **2. Blockchain Integration**

- Blockchain technology provides secure, immutable, and transparent records.
- Smart contracts automate processes such as PES request processing, tracking of supply chain transactions, and verification to ensure regulatory compliance, reducing human intervention and improving reliability.
- A solution based on Hyperledger Fabric is used due to its permissioned access control and high-security features.

### 3. Optimal Queue Model

- A queuing model prioritizes service requests using a first-in, first-out (FIFO) system to ensure fairness and reduce waiting times.
- The system calculates key metrics such as response time, service rates, and queue lengths to enhance performance.
- Actual arrival time, queue length, and end-to-end delay are measured and controlled effectively.

### 4. Revised Fitness-Based Political Optimizer (RF-PO)

- The RF-PO algorithm enhances the traditional Political Optimizer by introducing a fitness-based parameter randomization technique.
- It fine-tunes the queue model parameters, such as temperature, smoke, and humidity thresholds, ensuring optimized response times and minimal delays.
- The algorithm dynamically adapts to changes in supply chain demands, improving scalability and system performance.

### 5. Simulation and Performance Validation

- The framework is implemented using Python and tested under various conditions.
- Comparative analysis with existing methods demonstrates the superiority of the proposed system in reducing delays, optimizing resource utilization, and ensuring transparency.

#### 2.4.3 Advantages

##### 1. Enhanced Security

- Blockchain technology ensures immutable and tamper-proof records, reducing the risk of data breaches and unauthorized access.
- Distributed ledger technology improves data reliability and prevents single-point failures.

## **2. Transparency**

- The framework provides a clear and traceable record of all supply chain transactions, increasing accountability and trust among stakeholders.

## **3. Efficiency**

- The optimal queue model minimizes delays and optimizes the processing of service requests, reducing waiting times and improving customer satisfaction.
- Smart contracts automate routine tasks, eliminating manual errors and reducing processing time.

## **4. Scalability**

- The RF-PO algorithm enhances the system's ability to handle increased transaction volumes and IoT devices in expanding smart city environments.

## **5. Cost Savings**

- Automation through blockchain and smart contracts reduces administrative overhead and intermediary costs, streamlining supply chain operations.

## **6. Real-Time Monitoring**

- IoT integration provides continuous data collection, enabling real-time decision-making and proactive responses to emergencies.

### **2.4.4 Disadvantages**

#### **1. High Computational Demand**

- The complex computations required by blockchain and the RF-PO algorithm necessitate substantial processing power, making implementation expensive.

#### **2. Initial Costs**

- The setup of IoT devices, blockchain infrastructure, and edge computing servers can be prohibitively expensive for smaller businesses or organizations.

#### **3. Interoperability Issues**

- Combining various IoT devices and blockchain systems requires solving compatibility and standardization issues, which are time-consuming and costly.

#### **4. Scalability Restrictions**

- Existing blockchain systems find it difficult to process big data and transaction volumes efficiently with the improvements being implemented.

#### **5. Energy Use**

- The high energy demand of blockchain operations may be a reason for sustainability issues, especially in the case of big applications.

#### **6. Complexity of Deployment**

- Implementing the framework entails large-scale blockchain, IoT, and optimisation algorithms, which can limit adoption in less technically resourced organisations.

#### **2.4.5 Conclusion**

The architecture promotes security, transparency, and efficiency by leveraging blockchain's immutability and IoT's real-time monitoring. RF-PO is most appropriate for dynamic supply chains because it can enhance decision-making and scalability. While it is limited by high computational expense and initial investment, it offers a strong solution for modern supply chain management with potential future enhancements in scalability, interoperability, and cost-effectiveness.

### **2.5 Blockchain Private File Storage-Sharing Method Based on IPFS[5]**

#### **2.5.1 Overview**

The research article, "Blockchain Private File Storage-Sharing Method Based on IPFS," advocates a novel approach to addressing safe and effective file storage and sharing issues. Implementing blockchain technology with Named Data Networking (NDN) and the InterPlanetary File System (IPFS), the novel model ensures decentralization, immutability and knowledge file traceability. It is a secure, decentralized system capable of addressing

challenges such as unauthorized access, duplication, and alteration of files and improving network speed and storage capability.

### 2.5.2 Methodology

#### 1. NDN-Based Encryption

- NDN is employed to encrypt and sign files, separating data storage security from transmission.
- The content-centric addressing is implemented by the NDN network to ensure secure and efficient routing of encrypted data packets.

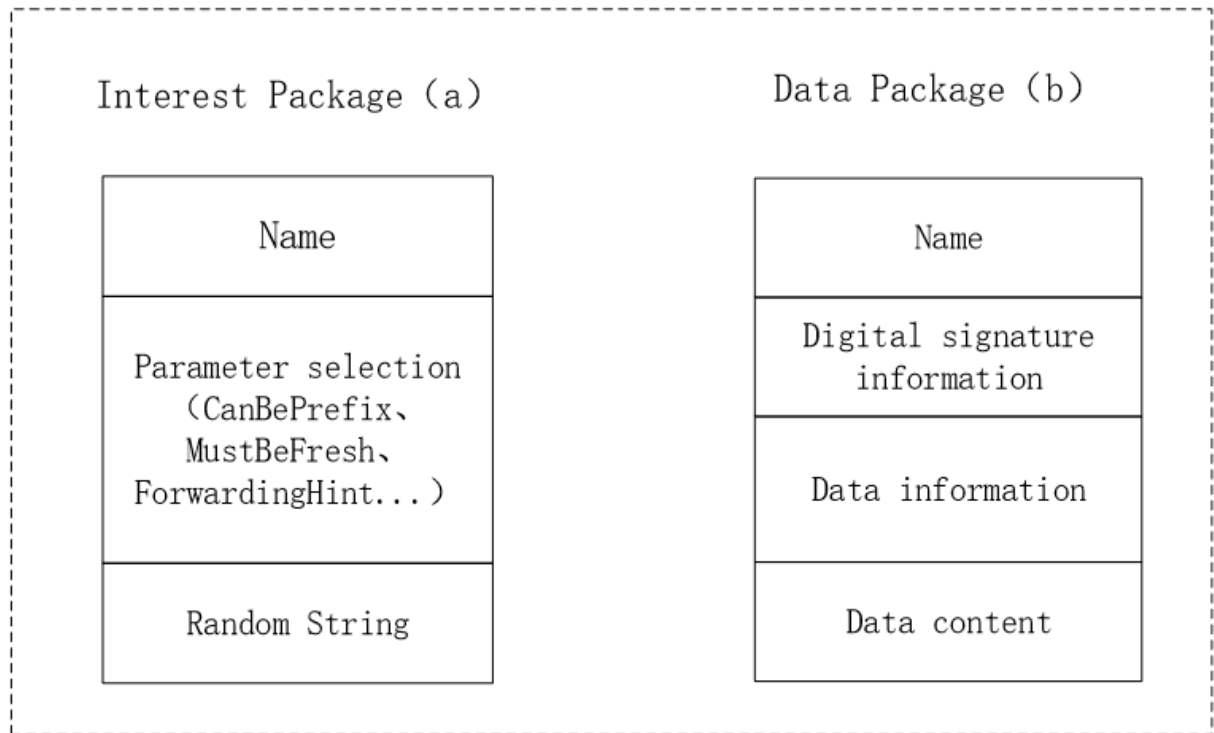


Figure 2.6: The interest packet (a) and the data packet (b)



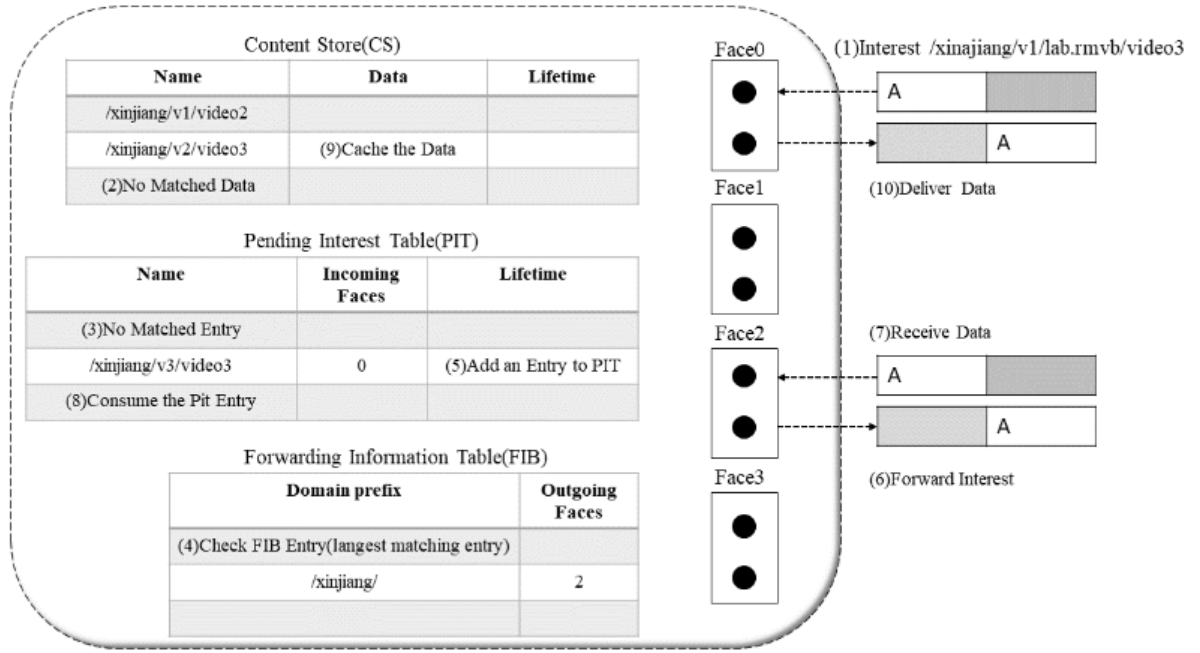


Figure 2.7: NDN forwarding model.

## 2. Blockchain Integration

- Blockchain serves as the backbone for immutability and traceability by storing metadata, including file hash values and transaction details.
- Smart contracts enable automation of file access permissions, ensuring compliance with the publisher's rules.

## 3. Private IPFS Network

- File content is stored in a private IPFS network, reducing redundancy and alleviating storage pressure on the blockchain.
- IPFS provides a content-based addressing mechanism, allowing for efficient retrieval and distributed file storage.

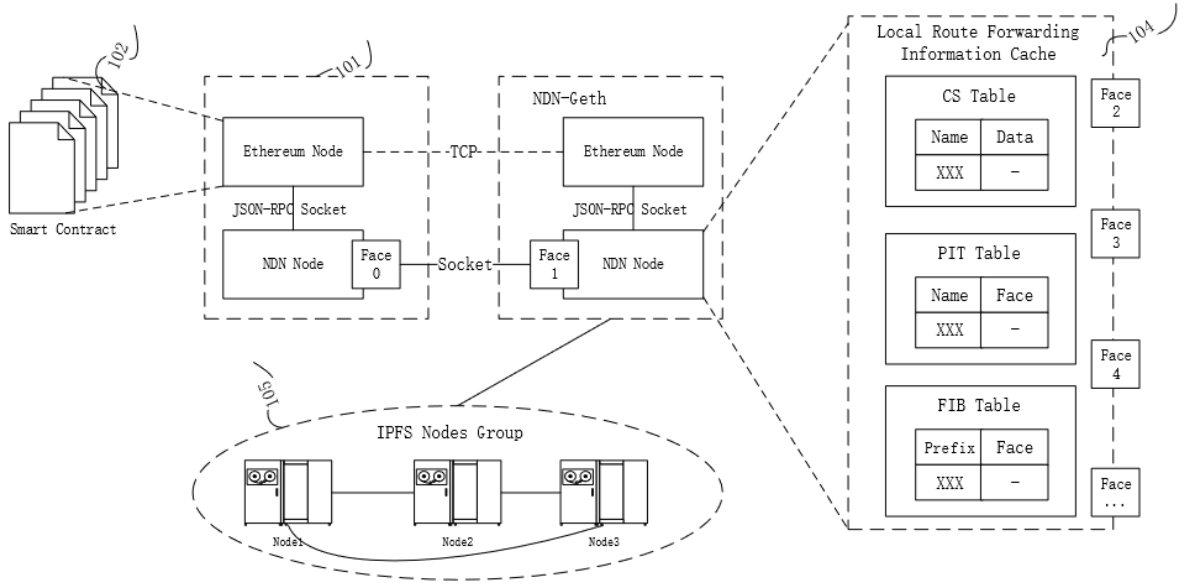


Figure 2.8: Data communication and forwarding process.

#### 4. Reverse Path Forwarding in NDN

- NDN's reverse path forwarding mechanism optimizes file transfer by caching intermediate routing information for future requests.
- This improves forwarding efficiency and reduces data transmission delays.

#### 5. Performance Evaluation

- Simulations compare the proposed system with traditional networks, demonstrating superior throughput, lower packet loss, and reduced latency.
- Storage tests validate the model's ability to securely and efficiently manage files of various sizes.

##### 2.5.3 Advantages

###### 1. Security

- Blockchain ensures tamper-proof storage of metadata, while NDN's signature-based encryption secures file content.

## **2. Transparency**

- All transactions are traceable, enabling accountability and trust in file-sharing processes.

## **3. Efficiency**

- IPFS reduces redundancy, while NDN's caching and reverse path forwarding improve file retrieval performance.

## **4. Scalability**

- The integration of IPFS alleviates storage constraints on the blockchain, enabling handling of large datasets.

## **5. Network Performance**

- NDN's content-centric approach minimizes transmission delays and enhances data reuse.

### **2.5.4 Disadvantages**

#### **1. Technical Immaturity**

- NDN technology is still evolving, with incomplete development and implementation of its components.

#### **2. Simulation-Only Validation**

- The framework has been tested in controlled simulation environments, lacking real-world deployment and testing.

#### **3. Initial Setup Complexity**

- Implementing the private IPFS network and configuring the NDN system require significant technical expertise and resources.

#### **4. Private Blockchain Dependence**

- The model relies on trusted nodes in a private blockchain environment, which limits its applicability in open, public systems.

### **2.5.5 Conclusion**

This paper introduces an innovative method for secure and efficient knowledge file storage and sharing by integrating blockchain, NDN, and IPFS technologies. The framework enhances security, transparency, and performance while addressing limitations of traditional systems. However, further development and real-world validation are required to overcome challenges such as technical immaturity, deployment complexity, and reliance on private blockchain environments. The study lays a strong foundation for future research in decentralized file storage systems.

## **2.6 Summary and Gaps Identified**

### **Summary:**

The following table outlines the advantages and disadvantages of the works reviewed:

<b>Paper/Work</b>	<b>Advantages</b>	<b>Disadvantages</b>
<b>Blockchain-Based Drug Supply Chain Implementation (2023)</b>	Enhanced traceability, improved security, and automated transactions through smart contracts.	High computational costs, limited accessibility due to blockchain technical requirements, and risks associated with incorrect initial inputs.
<b>Recent Advances in Smart Contracts (2020)</b>	Transparency, trust, decentralization, automation, and enhanced security with scalable applications across various domains.	Complex development processes, issues with contract immutability, and high transaction fees.
<b>ECC-Based Authentication Protocol for RFID (2021)</b>	Lightweight, efficient authentication with high security for constrained environments, scalable protocols for multireader setups.	Scalability issues in reader-side storage and increased computational overhead for certificate-based protocols.
<b>IoT and Blockchain-Based Supply Chain Management (2024)</b>	Real-time monitoring, secure record-keeping, and optimized queue models for enhanced efficiency and scalability.	High computational demands, energy-intensive operations, and initial deployment complexity.
<b>Blockchain Private File Storage-Sharing Method (2022)</b>	Secure, decentralized file storage with IPFS integration, ensuring immutability, scalability, and efficient data retrieval.	Immature NDN technology, reliance on simulations for validation, and challenges in interoperability.

Table 2.1: Advantages and Disadvantages of Reviewed Works

### Gaps Identified:

- **Scalability Issues:** Existing blockchain systems struggle to handle large-scale data and transactions efficiently, limiting their applicability in global supply chains.
- **High Computational Costs:** Many solutions require significant processing power, making them less viable for resource-constrained environments.
- **Interoperability Challenges:** Integration with existing systems, especially IoT devices, lacks standardization and compatibility.
- **Limited Real-World Validation:** Many proposed frameworks have only been tested in simulations, lacking real-world deployment and practical insights.
- **Privacy and Security Concerns:** Public blockchains expose data to all participants, posing challenges for sensitive applications. Improved mechanisms for privacy are needed.

## 2.7 Chapter Summary

Chapter 2 provides a review of existing work related to blockchain and supply chain management. It examines various applications, such as using blockchain for tracking pharmaceuticals, integrating IoT for real-time monitoring, and leveraging IPFS for secure and efficient data storage. While these studies highlight advantages like improved traceability, transparency, and automation, they also reveal challenges like high computational costs, lack of standardization, and limited real-world testing. The chapter concludes by identifying gaps in current systems, including scalability issues and privacy concerns, and emphasizes the need for solutions that are practical and efficient for real-world applications.

# Chapter 3

## System Design

The "Smart Contract based Supply Chain Management" project is divided into several key modules, each handling a specific aspect of the network's design and functionality.

### 3.1 System Architecture

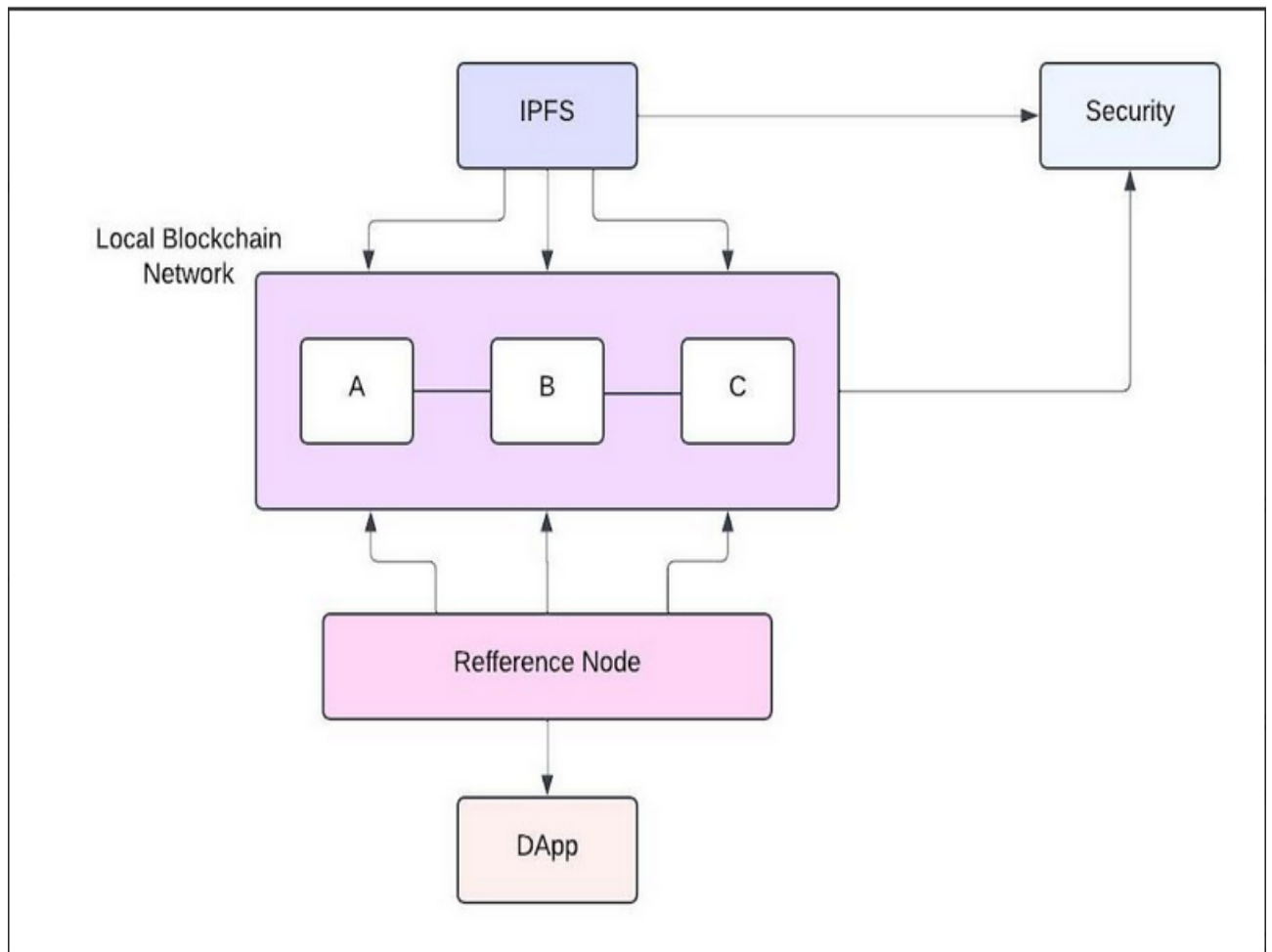


Figure 3.1: System Architecture

3.2 Data Flow Diagrams

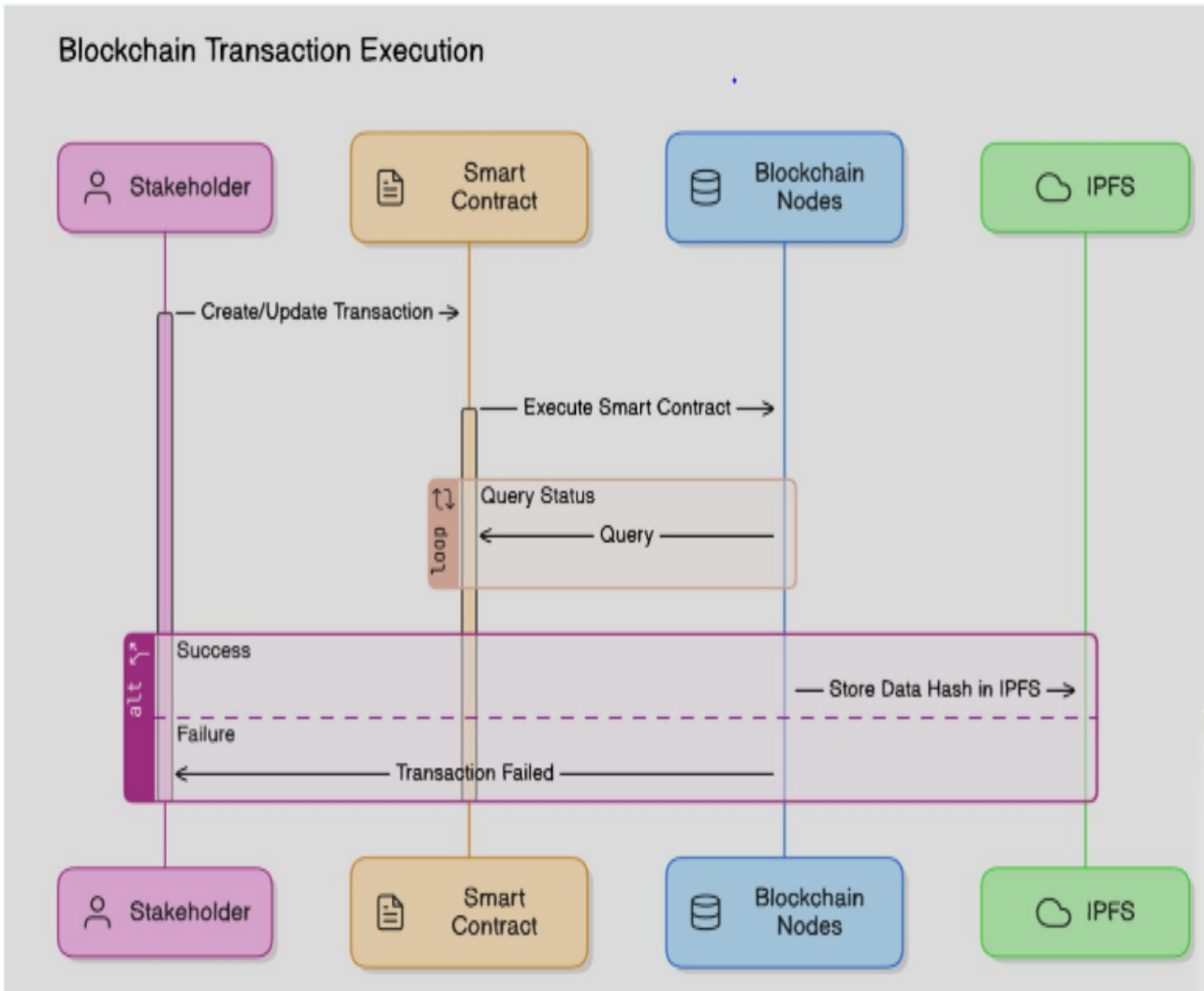


Figure 3.2: Data storage communication structure of the model, with components:



### 3.3 Component Design

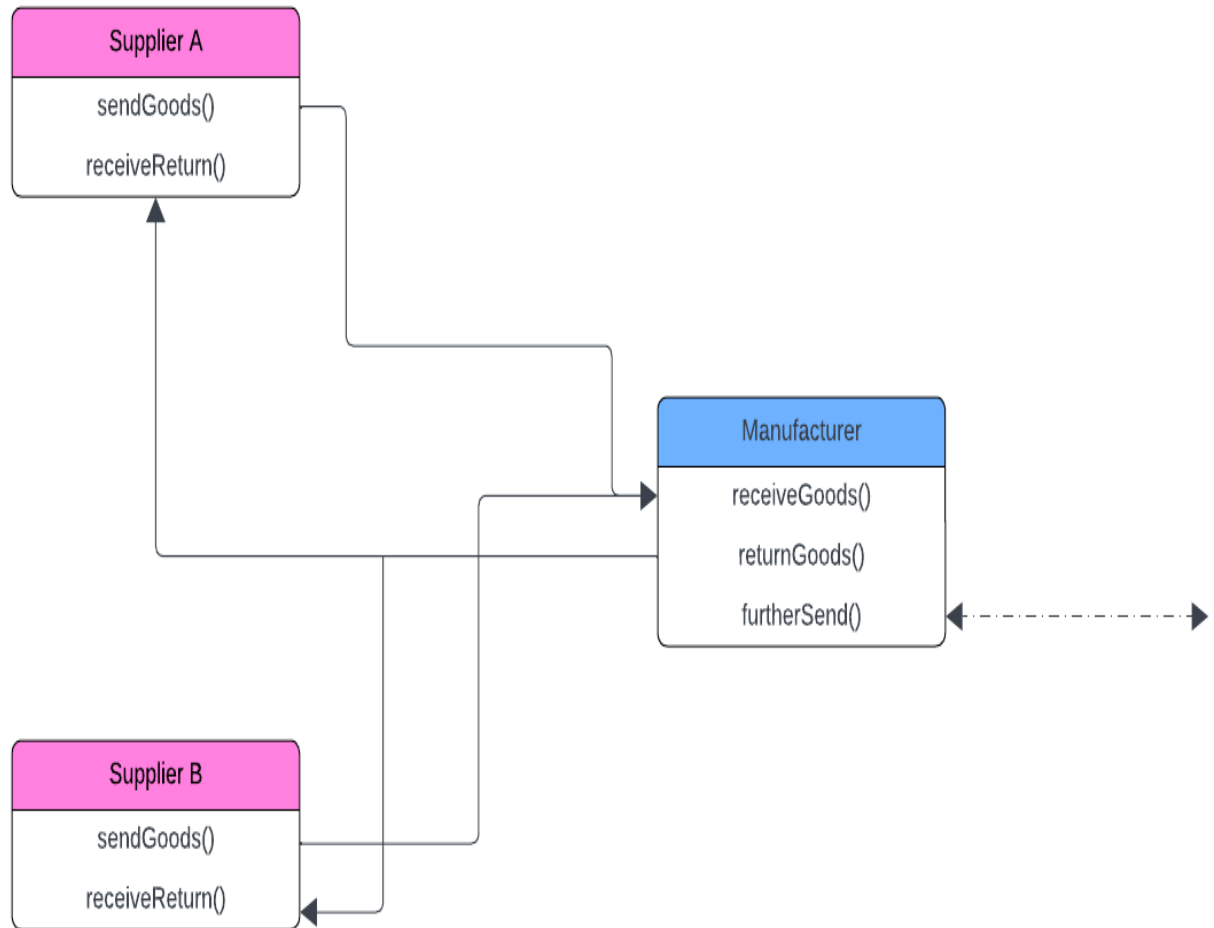


Figure 3.3: Basic Blockchain Network with interaction functions between them

### 3.4 Tools and Technologies

#### 3.4.1 Softwares

- Ganache
- Truffle Suite
- Node JS
- Solidity

### **3.4.2 Hardware**

- Processor: i7 or equivalent AMD.
- Cores: 2 or more cores
- Memory (RAM):8 GB or more.
- Disk Space: At least 10 GB of free disk space
- Disk Type: SSD
- Local Network: A basic home network Wi-Fi or Ethernet.
- Internet: Internet access for downloading dependencies and updates.

### **3.5 Module Divisions**

- Network Creation and Automation.
- Security and Encryption.
- Real World Data Simulation.
- IPFS.

### **3.6 Work Break Down**

#### **3.6.1 JK Yaswanth**

- Creation of smart contracts for each node.
- Deployment and Interaction between the nodes.
- Standard template smart contracts automation.
- Interaction and deployment automation.
- Automation with real-time data(for a target SCM)
- Integrating with Web3 for a Frontend GUI.

### **3.6.2 Eldho Markose**

- Encryption
- Decryption
- Automate File transfer to IPFS
- Mutual Authentication using public keys
- Key Management
- Integration of module

### **3.6.3 Jeevan James Mathew**

- Hashing
- User Verification
- File storage and Retrieval
- IPFS Implementation
- IPFS Integration

### **3.6.4 Hrishikesh MS**

- Creation of Smart Contract
- Documenting Network Creation
- Checking Redundancy
- Blockchain management

## **3.7 Key Deliverables**

The Project aims to deliver a working prototype to mitigate the problems faced in a modern Supply Chain of an Industry. It would contain a Website developed for the end-user to view the complete industry at a glance giving all Information in a single view.

### 3.8 Project Timeline

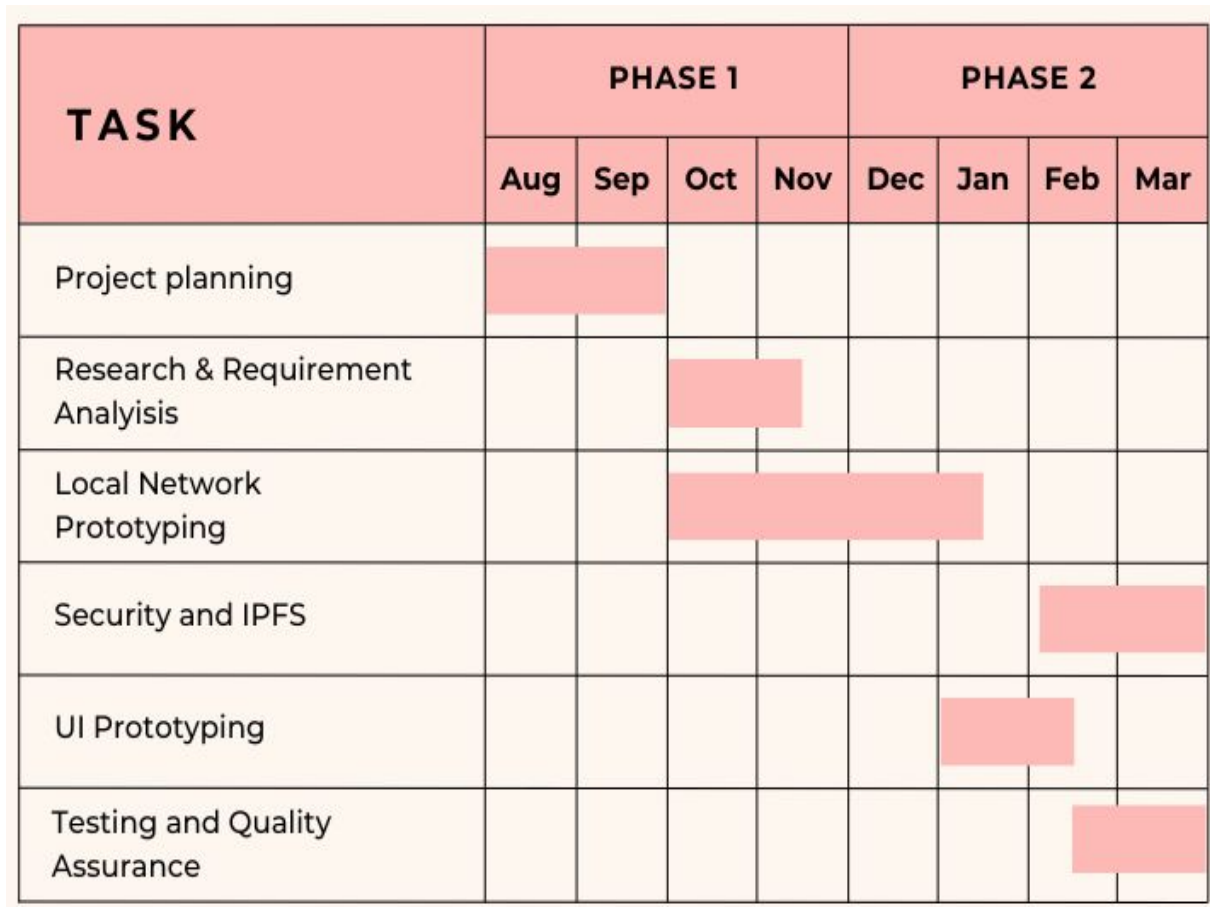


Figure 3.4: Gantt chart

### 3.9 Chapter Summary

This chapter outlines the design of the blockchain-based supply chain management system, focusing on creating a secure, transparent, and efficient platform. The system integrates blockchain for decentralized and tamper-proof data storage and IoT devices for real-time data collection and monitoring. It automates processes like tracking goods and verifying transactions through smart contracts, reducing manual errors and delays.

The chapter describes the flow of data within the system, from IoT devices to blockchain storage, ensuring traceability and accountability at every step. Tools such as Ganache and Truffle are used for blockchain simulation, Solidity is employed for writing smart contracts, and Node.js handles backend operations. To manage large files efficiently, IPFS (Inter-Planetary File System) is used for off-chain storage, reducing the load on the blockchain.

network.

The system is divided into modules, including network creation, encryption for data security, real-world supply chain simulation, and integration with IPFS. Each team member is assigned specific responsibilities, such as creating smart contracts, ensuring data security, and managing storage. The chapter concludes by presenting a project timeline, detailing the progress and milestones in developing a reliable and scalable supply chain solution.

## Chapter 4

### Results and Discussions

#### 4.1 System Implementation

This project was implemented by using a modular approach, with each component: blockchain network, smart contracts, IoT integrations, and IPFS.

##### 4.1.1 Smart Contract Deployment

- **Transaction Confirmation:** Smart contracts were successfully deployed on a local blockchain network (Ganache) and transactions recorded and were immutable. For wallet integration we used MetaMask.
- **Gas Costs:** Deployment and execution were cost optimized by making the average gas fee of 0.0003 GO for each transactions.

##### 4.1.2 User Interface and Workflow

- **Role Based Access:** The Front end interface allowed Users to register on bases of roles they play in the supply chain for example like producer, distributor or retailer.
- **Real Time Tracking:** IoT enabled data logged on blockchain provided visibility into goods' conditions.

## 4.2 Outputs

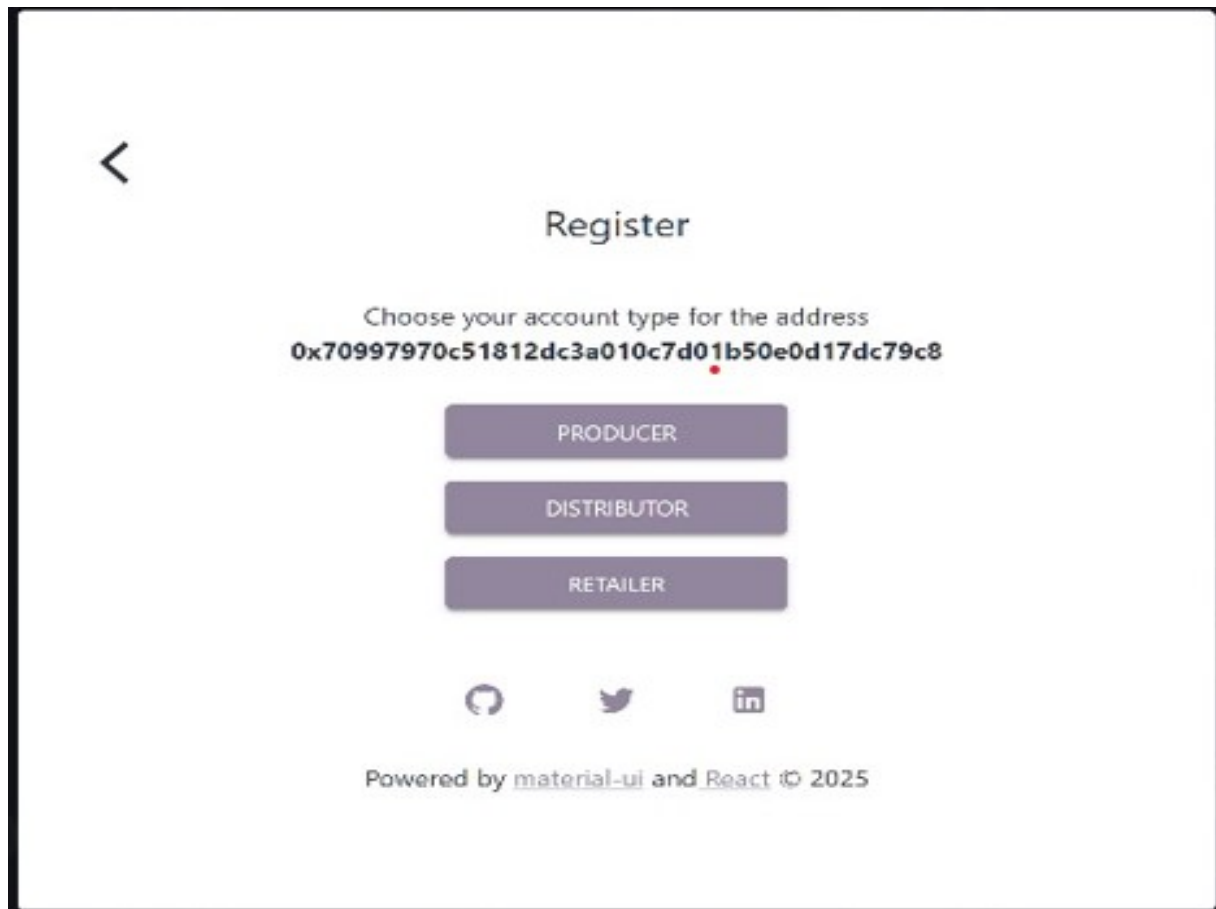


Figure 4.1: Dapp interface for user registration.

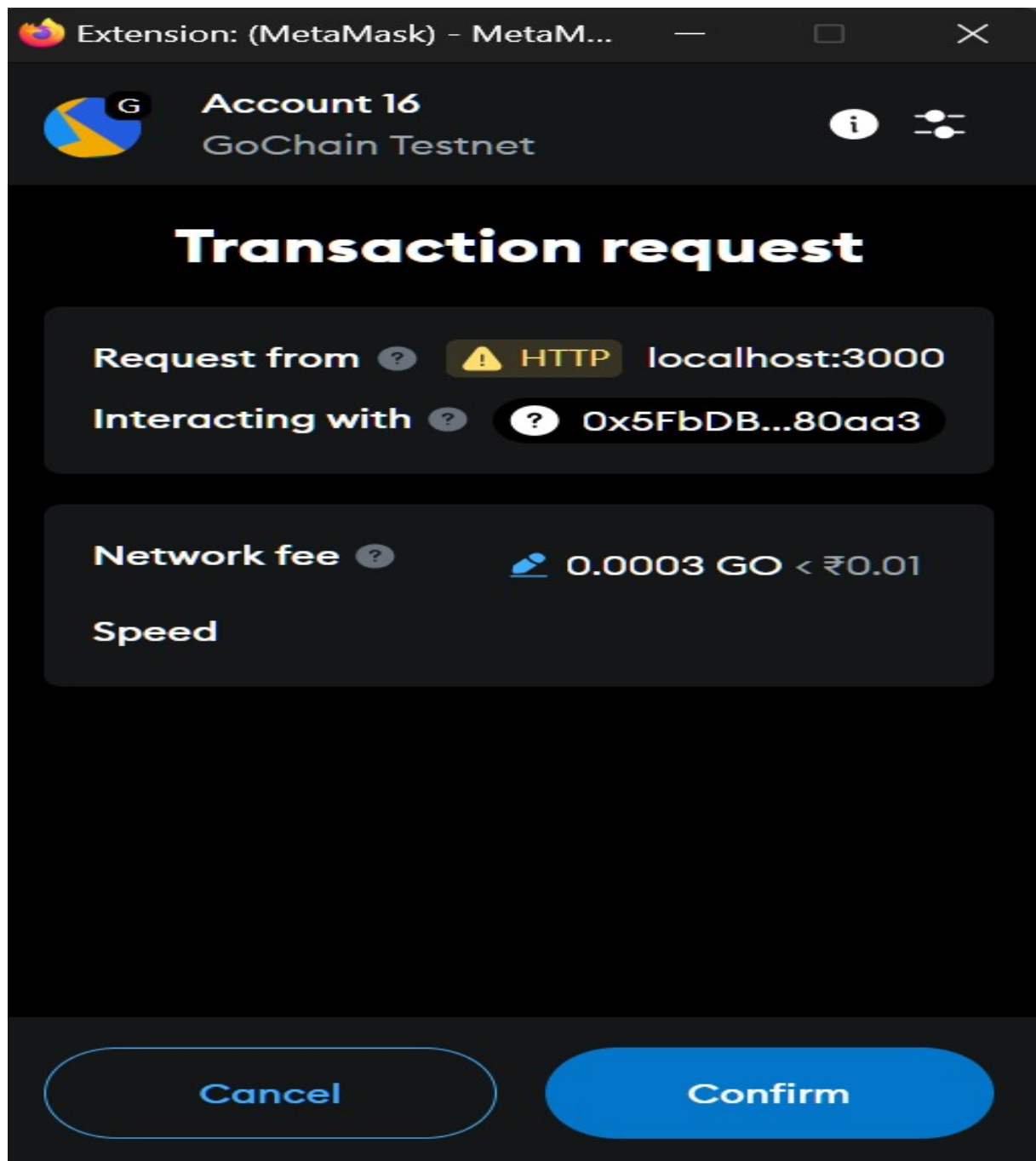


Figure 4.2: MetaMask Wallet Confirmation



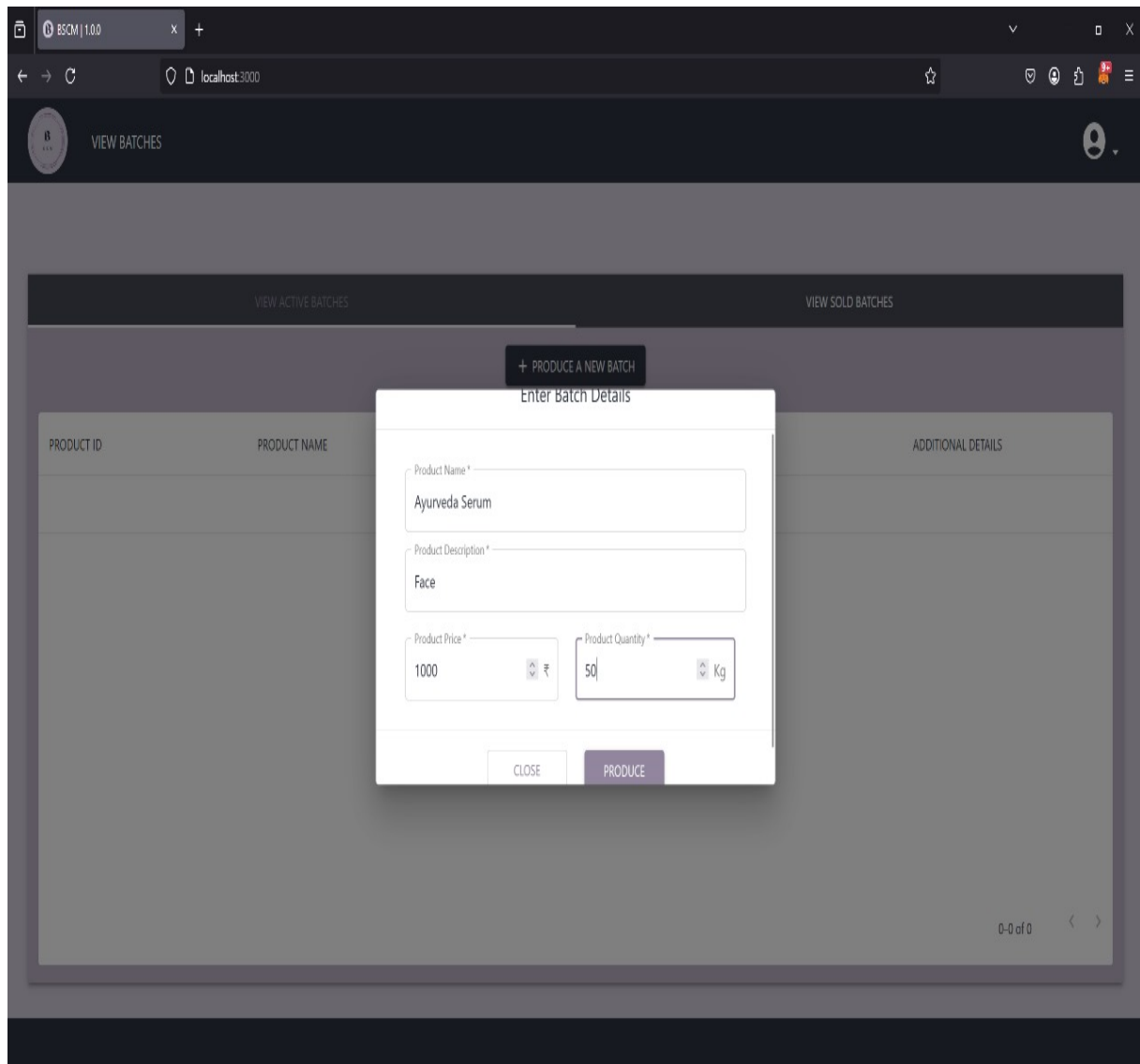


Figure 4.3: Producers's GUI

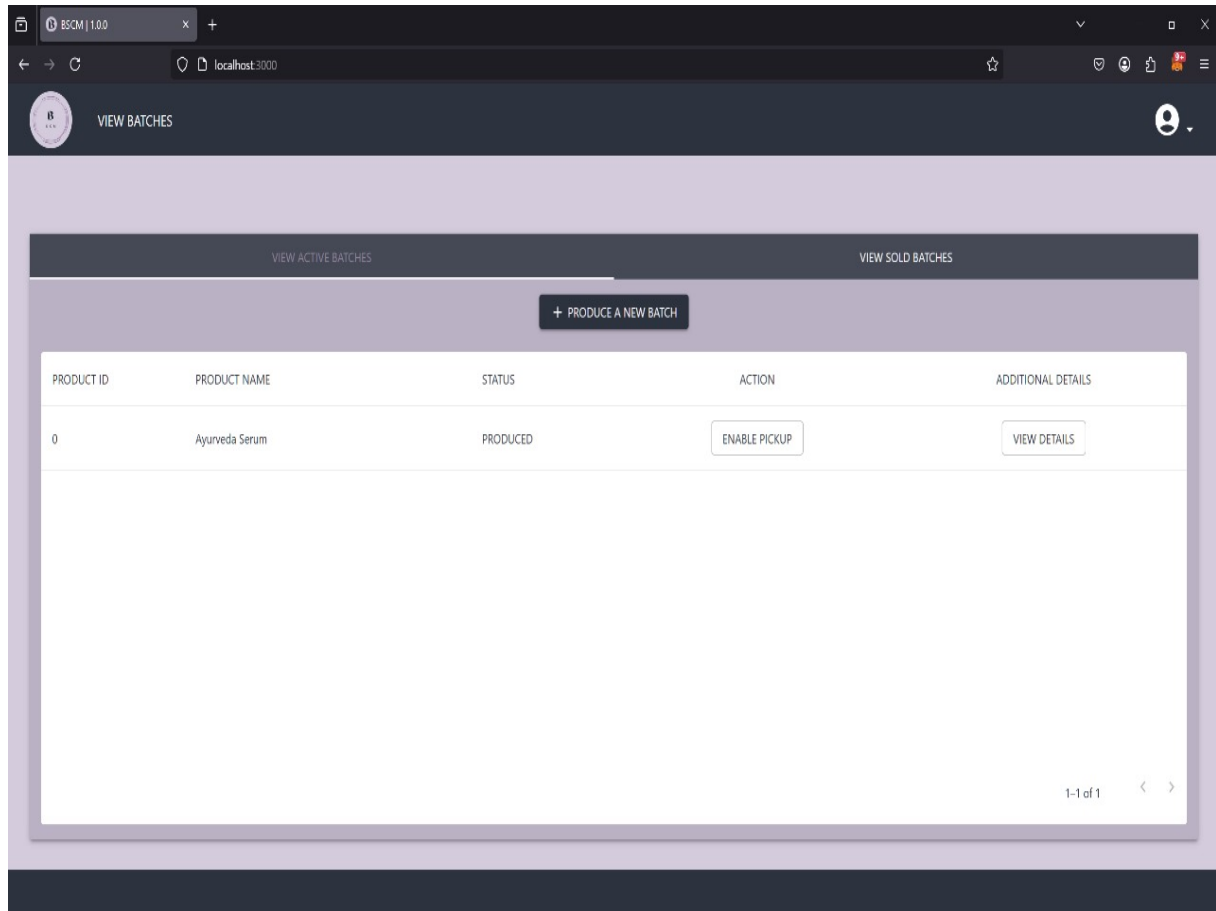


Figure 4.4: Buyer's Interface

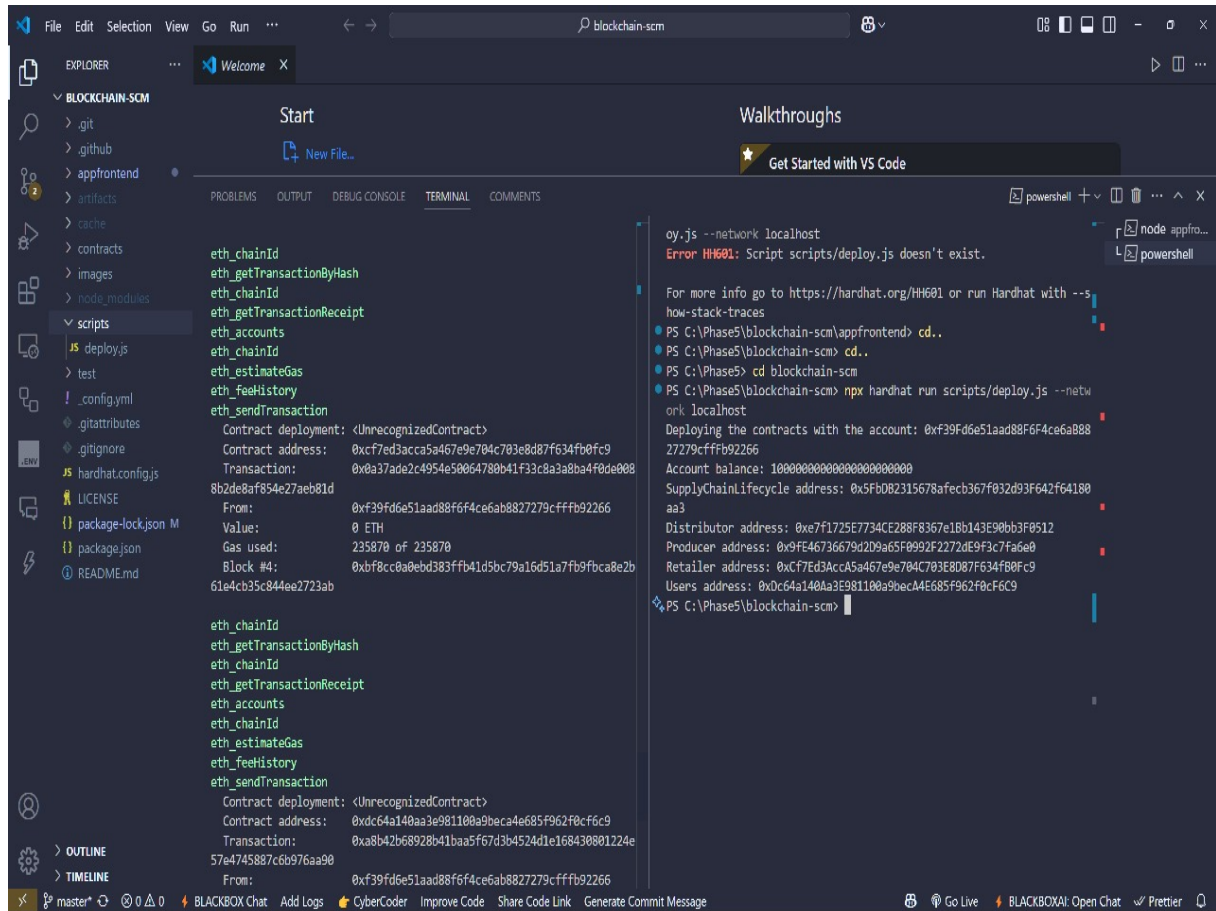


Figure 4.5: BackEnd Codes

### 4.3 Challenges Encountered

- Scalability issues with high transaction volumes
- IoT data inconsistencies requiring validation layers

#### 4.4 Comparative Analysis

Table 4.1: Comparison between Traditional and Blockchain Based SCM

<b>Metric</b>	<b>Traditional SCM</b>	<b>Blockchain Based SCM</b>
Transparency	Low (data silos)	High (shared ledger)
Traceability	Manual, error prone	Automated, immutable
Cost Efficiency	High (intermediaries)	Low (smart contracts)
Security	Vulnerable to fraud	Tamper proof, encrypted

## Chapter 5

### Conclusion and Future Enhancements

#### 5.1 Conclusions

The main aim of this project is to present a blockchain based supply chain management system that integrates smart contracts and IoT devices. IPFS is used for data storage, enhancing security, transparency, and efficiency across all stages of the supply chain. By automating processes through smart contracts, the system reduces dependency on manual intervention and minimizes the risk of fraud or data manipulation. The integration of IoT devices allows for real time tracking of goods, while IPFS ensures secure and decentralized storage of critical documentation. Overall, this provides a more trustworthy, traceable, and efficient supply chain that can be useful for industries where product authenticity and logistics visibility are essential.

#### 5.2 Future Enhancements

While the current system successfully demonstrates the potential of blockchain in supply chain management, there are several areas that can be improved and expanded in the future to make the solution more robust and scalable.

- **Mobile Application Integration:** Develop a cross platform mobile app to allow users to access and interact with the system on the go.
- **AI-Based Predictive Analytics:** Integrate AI to forecast demand, optimize inventory, and detect anomalies in the supply chain.
- **Multichain Interoperability:** Enhance the system to support multiple blockchain networks for greater flexibility and compatibility.

- **Advanced IoT Integration:** Include support for more sensor types (e.g., humidity, shock, light exposure) to monitor fragile or sensitive goods.
- **User Role Expansion:** Add more user roles like customs authorities, quality inspectors, or insurance agents for broader real-world usage.
- **Automated Dispute Resolution:** Implement smart contract logic to resolve disputes automatically based on predefined rules and evidence.
- **Regulatory Compliance Module:** Add a feature to ensure and verify that all processes comply with international regulations and standards.

## References

- [1] K. C. Bandhu, R. Litoriya, P. Lowanshi *et al.*, “Making drug supply chain secure traceable and efficient: a blockchain and smart contract based implementation,” *Multimedia Tools and Applications*, vol. 82, pp. 23 541–23 568, 2023.
- [2] V. Y. Kemmoe, W. Stone, J. Kim, D. Kim, and J. Son, “Recent advances in smart contracts: A technical overview and state of the art,” *IEEE Access*, vol. 8, pp. 117 782–117 801, 2020.
- [3] H. Lamrani Alaoui, A. El Ghazi, M. Zbakh, A. Touhafi, and A. Braeken, “A highly efficient ecc-based authentication protocol for rfid,” *Journal of Sensors*, vol. 2021, pp. Article ID 8 876 766, 1–16, July 2021.
- [4] A. Y. A. B. Ahmad, N. Verma, N. M. Sarhan, E. M. Awwad, A. Arora, and V. O. Nyangaresi, “An iot and blockchain-based secure and transparent supply chain management framework in smart cities using optimal queue model,” *IEEE Access*, vol. 12, pp. 51 752–51 771, 2024.
- [5] P. Kang, W. Yang, and J. Zheng, “Blockchain private file storage-sharing method based on ipfs,” *Sensors*, vol. 22, no. 14, p. 5100, 2022.
- [6] W. Zou *et al.*, “Smart contract development: Challenges and opportunities,” *IEEE Transactions on Software Engineering*, vol. 47, no. 10, pp. 2084–2106, Oct 2021.
- [7] K. Christidis and M. Devetsikiotis, “Blockchains and smart contracts for the internet of things,” *IEEE Access*, vol. 4, pp. 2292–2303, 2016.
- [8] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, and G. Danezis, “Chainspace: A sharded smart contracts platform,” *arXiv preprint arXiv:1708.03778*, 2017. [Online]. Available: <https://arxiv.org/abs/1708.03778>

- [9] H. Taherdoost, "Smart contracts in blockchain technology: A critical review," *Information*, vol. 14, p. 117, 2023. [Online]. Available: <https://doi.org/10.3390/info14020117>
- [10] Y. Nakamura, Y. Zhang, M. Sasabe, and S. Kasahara, "Exploiting smart contracts for capability-based access control in the internet of things," *Sensors*, vol. 20, no. 6, p. 1793, 2020.
- [11] S. Namasudra, G. C. Deka, P. Johri, M. Hosseinpour, and A. H. Gandomi, "The revolution of blockchain: State-of-the-art and research challenges," *Archives of Computational Methods in Engineering*, vol. 28, pp. 1497–1515, 2021, [CrossRef].
- [12] M. A. Alqarni, M. S. Alkatheiri, S. H. Chauhdary, and S. Saleem, "Use of blockchain-based smart contracts in logistics and supply chains," *Electronics*, vol. 12, no. 6, p. 1340, 2023. [Online]. Available: <https://doi.org/10.3390/electronics12061340>
- [13] M. Wang, Y. Wu, B. Chen, and M. Evans, "Blockchain and supply chain management: A new paradigm for supply chain integration and collaboration," *Operations and Supply Chain Management: An International Journal*, vol. 14, pp. 111–122, 2020.
- [14] S. E. Chang and Y. Chen, "When blockchain meets supply chain: A systematic literature review on current development and potential applications," *IEEE Access*, vol. 8, pp. 62 478–62 494, 2020.
- [15] A. Saini, Q. Zhu, N. Singh, Y. Xiang, L. Gao, and Y. Zhang, "A smart-contract-based access control framework for cloud smart healthcare system," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5914–5925, April 2021.
- [16] S. Majumder, S. Ray, D. Sadhukhan *et al.*, "Ecc-coap: Elliptic curve cryptography based constraint application protocol for internet of things," *Wireless Personal Communications*, vol. 116, pp. 1867–1896, 2021.
- [17] C. Patel, A. K. Bashir, A. A. AlZubi, and R. Jhaveri, "Ebake-se: A novel ecc-based authenticated key exchange between industrial iot devices using secure element," *Digital Communications and Networks*, vol. 9, no. 2, 2023.



- [18] S. Ullah, Z. Jiangbin, N. Din, M. T. Hussain, F. Ullah, and M. Yousaf, “Elliptic curve cryptography: Applications, challenges, recent advances, and future trends: A comprehensive survey,” *Computer Science Review*, 2022, school of Software, Northwestern Polytechnical University, Xi’an, PR China.
- [19] A. Braeken, P. Kumar, and A. Martin, “Efficient and provably secure key agreement for modern smart metering communications,” *Energies*, vol. 11, no. 10, p. 2662, 2018.
- [20] K. Sowjanya, M. Dasgupta, and S. Ray, “An elliptic curve cryptography based enhanced anonymous authentication protocol for wearable health monitoring systems,” *International Journal of Information Security*, vol. 19, no. 1, pp. 129–146, 2020.
- [21] N. Dinarvand and H. Barati, “An efficient and secure rfid authentication protocol using elliptic curve cryptography,” *Wireless Networks*, vol. 25, no. 1, pp. 415–428, 2019.
- [22] F. Merabet, A. Cherif, M. Belkadi, O. Blazy, E. Conchon, and D. Sauveron, “New efficient m2c and m2m mutual authentication protocols for iot-based healthcare applications,” *Peer-to-Peer Networking and Applications*, vol. 13, no. 2, pp. 439–474, 2020.

# Appendix A: Presentation



# **Smart Contracts in Supply Chain Management**

## **100% Evaluation Presentation**

**Guide: Ms. JISHA MARY JOSE**

**J K Yaswanth(U2I03I05)  
Eldho Markose(U2I03084)  
Jeevan James(U2I03I07)  
Hrishikesh M Sreenivas(U2I03I03)**

# Introduction

1

- Today any industry contains multiple agents working together to make the whole industry viable.
- Synchronizing all these agents using the conventional Supply Chain Management uses too much energy and they lack transparency, security and integrity midst the different agents present.
- Hence, Integrating Blockchain technology with appropriate smart contracts with on-chain or off-chain data encryption helps to resolve these problems to an extent.

# PROJECT OBJECTIVE

2

- **To Investigate Challenges:** Analyze the current challenges faced by blockchain-based supply chain systems and identify potential solutions.
- **To Enhance Security:** Address security concerns by implementing robust cryptographic measures and decentralized mechanisms to protect data integrity and confidentiality
- **To Optimize Processes:** Utilize smart contracts to automate supply chain processes, reducing delays and improving efficiency.

- **To Ensure Transparency:** Develop a transparent system that allows stakeholders to access and verify data in real-time.
- **To Evaluate Attacks:** Examine common security attacks on blockchain networks and propose countermeasures to mitigate their impact on supply chain processes.
- **To Provide Recommendations:** Offer recommendations for the implementation of blockchain technology to improve security, transparency, and efficiency in supply chain management.



# MODULE DIVISION

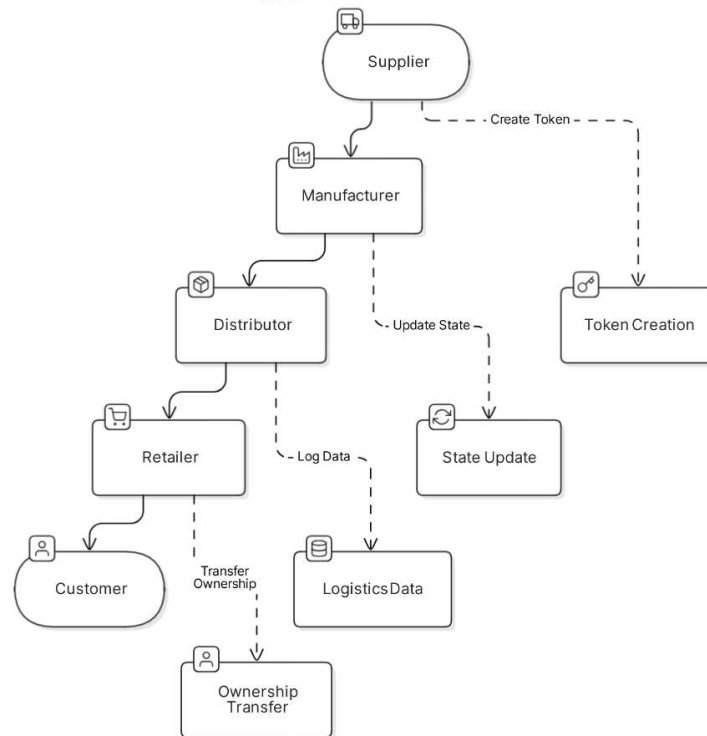
4

- **Network Creation and Automation**
  - **Security and Encryption**
  - **IPFS**
- 
- 
-

# WORK FLOW DIAGRAM

5

Supply Chain Process





# MODULES



## Network Creation and Automation Module

7

- **Purpose :-** Creation of Smart Contracts for the Local Simulation for various nodes incoming into the network and automating the process.
- **Key Features :**
  - **Automated Blockchain Setup:** Installs and configures Truffle and Hardhat for local Ethereum simulation.
  - **Smart Contract Deployment:** Automatically compiles and deploys supply chain smart contracts.
  - **Order Management:** Enables order creation, status tracking (Created, Shipped, Delivered), and order querying.
  - **Simulation Automation:** Uses Node.js to automate contract interactions and simulate supply chain events.
  - **End-to-End Automation:** Single script to run the entire process (setup, deployment, simulation).
  - **Extensible:** Easily customizable and scalable to include more features and actors.
  - **Transaction :** Transaction simulation is done through using MetaMask

# ENCRYPTION MODULE DESIGN

## 1. Key Generation (ECC):

- The module uses Elliptic Curve Cryptography (ECC) to generate a public-private key pair.
- The public key encrypts a randomly generated AES symmetric key, while the private key is used later for decryption.

## 2. File Encryption:

- The content of the input file is encrypted using AES (Advanced Encryption Standard) with a 256-bit key in CBC (Cipher Block Chaining) mode.
- AES ensures fast and secure content encryption.

## 3. Data Packaging:

- The encrypted AES key, initialization vector (IV), and file ciphertext are combined into a single binary file with metadata to facilitate secure transfer and storage.

## 4. File Decryption:

- The private ECC key decrypts the AES key.
- AES decryption reverses the encryption process to retrieve the original plaintext file.

# ENCRYPTION MODULE WORKFLOW

## ENCRYPTION WORKFLOW

### 1. Input:

- Prompts the user for:
  - The file to encrypt.
  - Paths to save the private key, public key, and the combined encrypted file.

### 2. Process:

- Generates ECC key pair (public and private keys).
- Encrypts the file using the public key and AES encryption.
- Combines the encrypted\_symmetric\_key, iv, and ciphertext into a single binary file.

### 3. Output:

- Saves the private key, public key, and the encrypted file.

## IPFS MODULE STORAGE

- **IPFS** serves as a decentralized, **off-chain data storage** system. Its main purpose is to store large files efficiently without burdening the blockchain, which primarily records only the **file's hash** (Content Identifier or CID). This approach ensures data integrity, availability, and scalability while leveraging blockchain for traceability and security.

### Key Features:

- **Decentralization:** Distributes files across multiple nodes, reducing reliance on a single point of failure.
- **Content-Based Addressing:** Uses unique CIDs (hashes) to reference files, ensuring data integrity.
- **Efficient Storage:** Large data is stored off-chain, reducing blockchain storage costs.
- **In-Network Caching:** IPFS caches data for faster future retrievals.
- **Scalability:** Handles large amounts of data without overloading the blockchain.
- **Security & Integrity:** Files can be verified against their on-chain hashes, ensuring they haven't been tampered with.

# Software and Hardware Requirements

## Hardware Requirements

- **Processor:** Intel Core i7 or AMD Ryzen 7
- **RAM:** 16GB or higher
- **GPU:** NVIDIA RTX 3050 or higher
- **Storage:** SSD (512GB or higher)

## Software Requirements

- **Solidity:** for writing smart contracts on the Ethereum blockchain, enabling automation of supply chain processes.
- **HardHat:** For managing smart contract development and testing
- **React.js & Tailwind CSS:** For building user friendly and responsive front end
- **Ethereum:** Ethereum ensures transparency, security and decentralized data storage
- **Ether.js:** Interact with Ethereum smart contracts more efficiency.
- **MetaMask:** Enables secure digital wallet transaction.
- **IPFS:** Stores large large files off-chain while keeping data integrity.

## Gantt Chart

TASK	PHASE 1				PHASE 2			
	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar
Project planning								
Research & Requirement Analysis								
Local Network Prototyping								
Security and IPFS								
UI Prototyping								
Testing and Quality Assurance								

# RESULTS



## Front-End First Interface

14

<




### Register

Choose your account type for the address  
**0x70997970c51812dc3a010c7d01b50e0d17dc79c8**

PRODUCER

DISTRIBUTOR

RETAILER

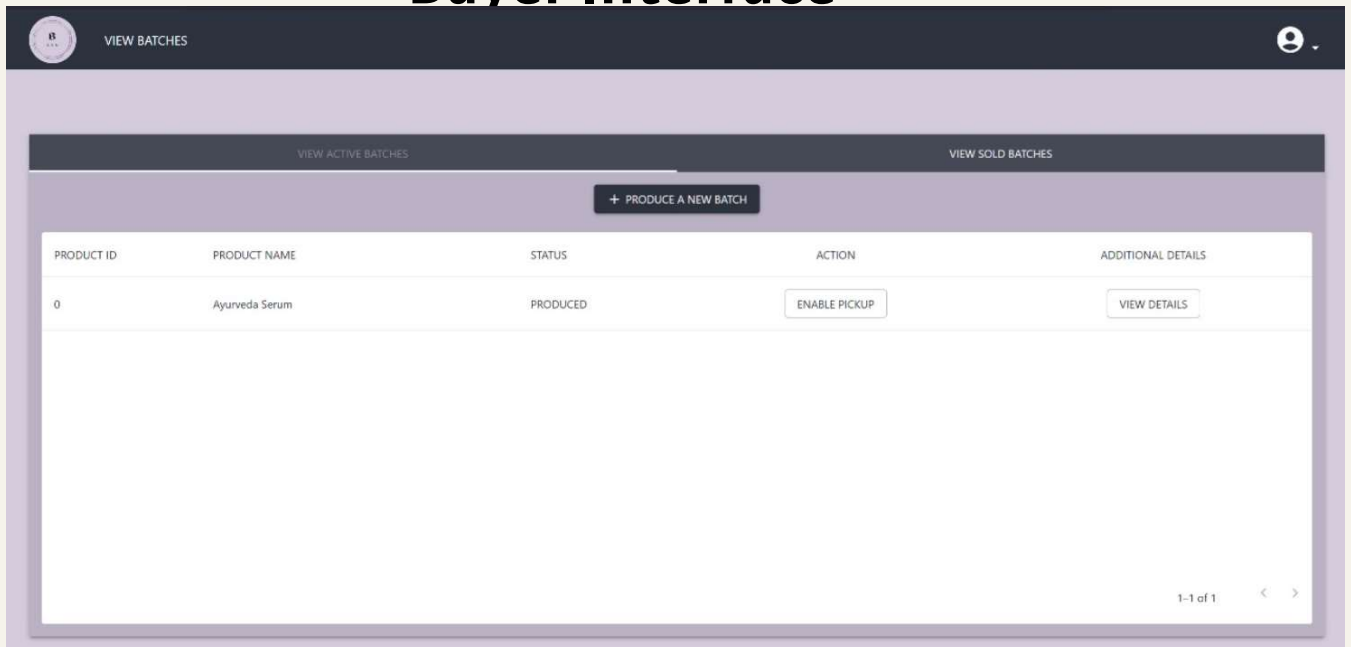
  

Powered by [material-ui](#) and [React](#) © 2025

# Producer Interface

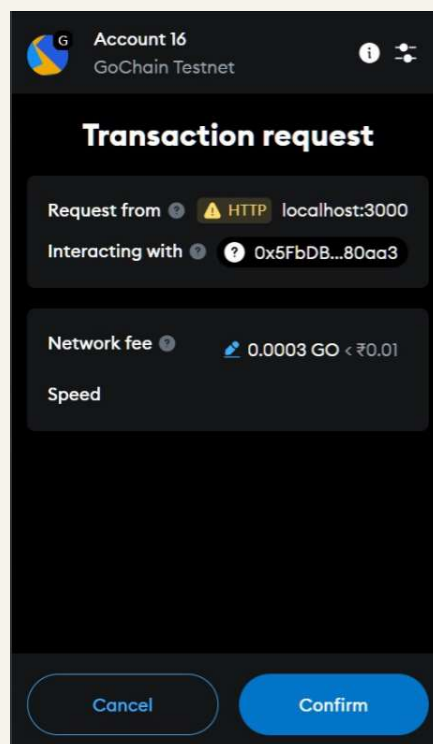
The screenshot displays a web application interface for a producer. At the top, a dark header bar contains a logo on the left, the text "VIEW BATCHES" in the center, and a user profile icon on the right. Below the header, the main content area is divided into two sections: "VIEW ACTIVE BATCHES" on the left and "VIEW SOLD BATCHES" on the right. A central button labeled "+ PRODUCE A NEW BATCH" is positioned above a modal window titled "Enter Batch Details". The modal contains the following fields: "Product Name \*" with the value "Ayurveda Serum", "Product Description \*" with the value "Face", "Product Price \*" with the value "1000" and a currency selector set to "₹", and "Product Quantity \*" with the value "50" and a unit selector set to "Kg". At the bottom of the modal are "CLOSE" and "PRODUCE" buttons. The background interface includes a table with columns "PRODUCT ID" and "PRODUCT NAME" under the "VIEW ACTIVE BATCHES" section, and a section titled "ADDITIONAL DETAILS" under the "VIEW SOLD BATCHES" section. A pagination indicator at the bottom right shows "0-0 of 0" with navigation arrows.

# Buyer Interface



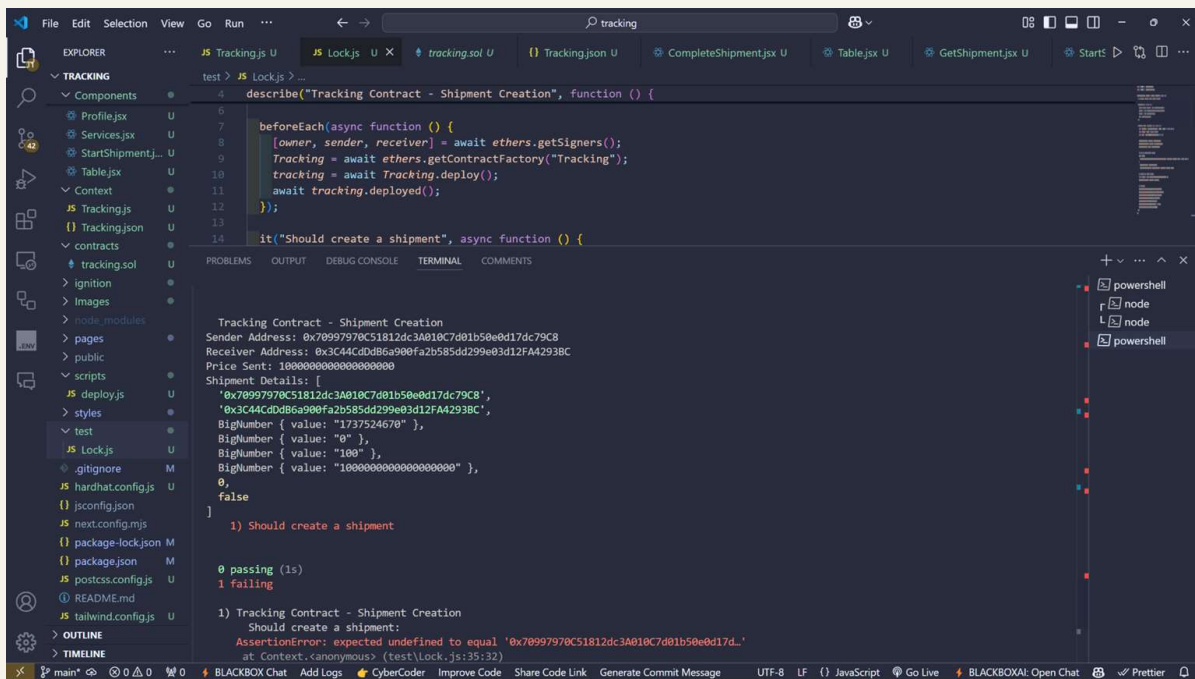
# METAMAASK

17



## Back-End Current Phase

Shipment created and sent to the receiver, but assertion errors are persisting due to the connection error in metamask



```
test > JS Lock.js > ...
4 describe("Tracking Contract - Shipment Creation", function () {
5
6
7   beforeEach(async function () {
8     [owner, sender, receiver] = await ethers.getSigners();
9     Tracking = await ethers.getContractFactory("Tracking");
10    tracking = await Tracking.deploy();
11    await tracking.deployed();
12  });
13
14  it("Should create a shipment", async function () {
15
16    Tracking Contract - Shipment Creation
17    Sender Address: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
18    Receiver Address: 0x3C44CdD86a900fa2b585dd299e03d12FA4293BC
19    Price Sent: 1000000000000000000
20    Shipment Details: [
21      '0x70997970C51812dc3A010C7d01b50e0d17dc79C8',
22      '0x3C44CdD86a900fa2b585dd299e03d12FA4293BC',
23      BigNumber { value: "1737524670" },
24      BigNumber { value: "0" },
25      BigNumber { value: "100" },
26      BigNumber { value: "1000000000000000000" },
27      0,
28      false
29    ]
30
31    1) Should create a shipment
32
33    0 passing (1s)
34    1 failing
35
36    1) Tracking Contract - Shipment Creation
37       Should create a shipment:
38       AssertionError: expected undefined to equal '0x70997970C51812dc3A010C7d01b50e0d17d...'
39       at Context.<anonymous> (test/Lock.js:35:32)
```

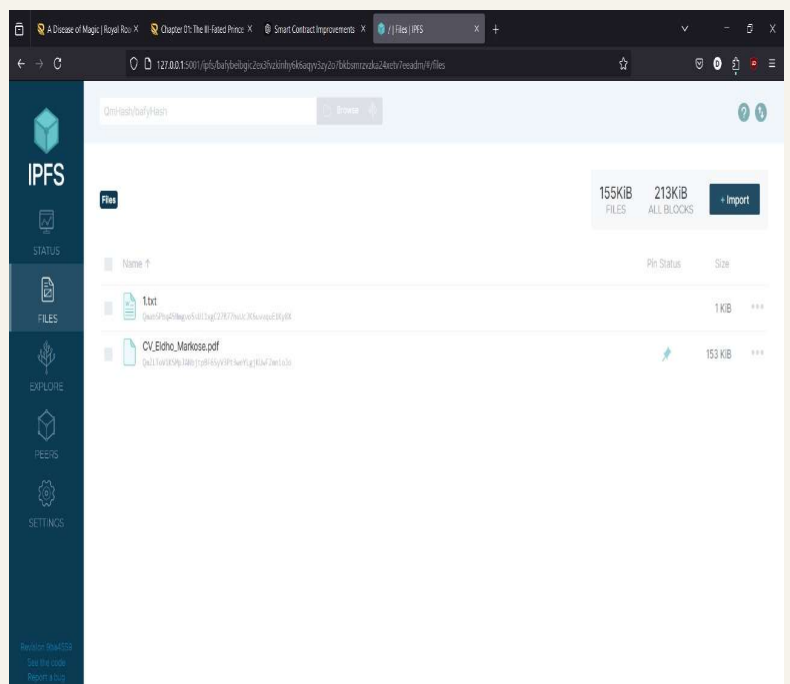
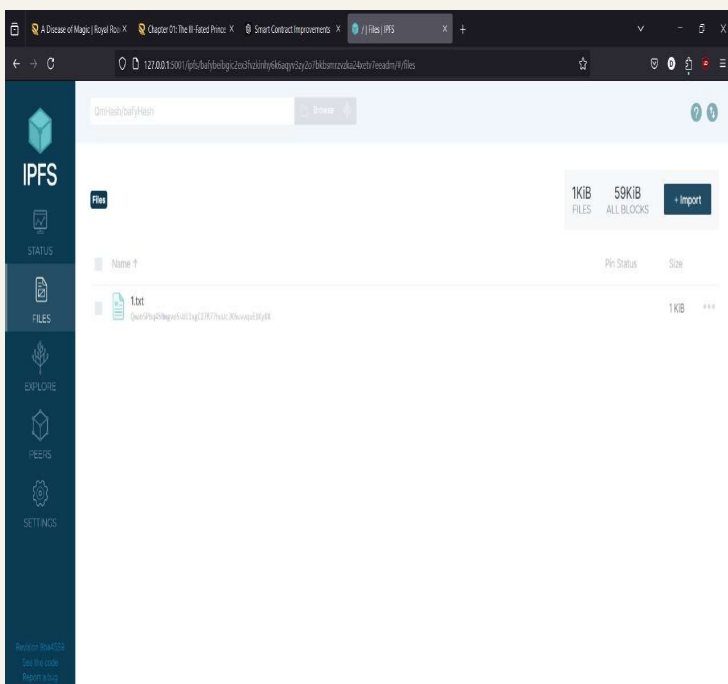
# IPFS Local Node Creation

19

```
C:\Users\emoff\Downloads\k... x + v
Initializing daemon...
Kubo version: 0.32.1
Repo version: 16
System version: amd64/windows
Golang version: go1.23.3
PeerID: 12D3KoolWMTt42ZY7aB2W8PCCBKyCeMYcPC9hVc46jqHTJwuS1tgM
Swarm listening on 127.0.0.1:4001 (TCP+UDP)
Swarm listening on 169.254.161.78:4001 (TCP+UDP)
Swarm listening on 169.254.195.139:4001 (TCP+UDP)
Swarm listening on 169.254.43.100:4001 (TCP+UDP)
Swarm listening on 169.254.74.119:4001 (TCP+UDP)
Swarm listening on 172.25.73.167:4001 (TCP+UDP)
Swarm listening on [2409:40f3:13:beaa:3492:d6ce:f4fe:77d6]:4001 (TCP+UDP)
Swarm listening on [2409:40f3:13:beaa:7dbf:8690:81cd:6815]:4001 (TCP+UDP)
Swarm listening on [::1]:4001 (TCP+UDP)
Run 'ipfs id' to inspect announced and discovered multiaddrs of this node.
RPC API server listening on /ip4/127.0.0.1/tcp/5001
WebUI: http://127.0.0.1:5001/webui
Gateway server listening on /ip4/127.0.0.1/tcp/8080
Daemon is ready
```

## IPFS Local Node Creation

20



# IPFS Local Node Creation


21

← → ↺

127.0.0.1:8080/ipfs/QmZLT6VikSMpJANbjtpBF6SyV3Pl3weYLgJKUwFZmnioJo

1 of 1

60%



**ELDHO MARKOSE**  
6/308, RAJAGIRI HOUSE,  
POURAMKUL # 3,  
CHULUKULU PIN 680009  
**Contact Information**  
Mobile: +91 9048657733  
Email: wenscott41@gmail.com  
**Education**  
Btech in Computer Science and  
Technology - Rajagiri School of  
Engineering and  
Technology, Kakkanad  
CGPA - 7.96  
90 (Computer Science) - Chavara Public  
School, Pata - 95%  
X - Jawahar Narendradev Vidyapeeth,  
Kottayam - 92.6%

**Background**

- Computer Science graduate from Rajagiri School of Engineering and Technology with a CGPA of 7.96.
- Designed and developed a full-stack e-commerce platform to handle basic operations like purchasing, returning, and stock updates as group project.
- Took charge of database management and schema modeling, designing efficient schemas in MongoDB to support unstructured datasets.
- Implemented a recommendation system using lazy learning algorithms to provide personalised book suggestions.
- Took responsibility for data preprocessing, cleaning and preparing raw datasets from Kaggle for machine learning model training.
- Created and fine-tuned machine learning models in Google Colab, optimizing their performance for accurate recommendations.

**Professional Experience**  
Nil

**Programming Skills**

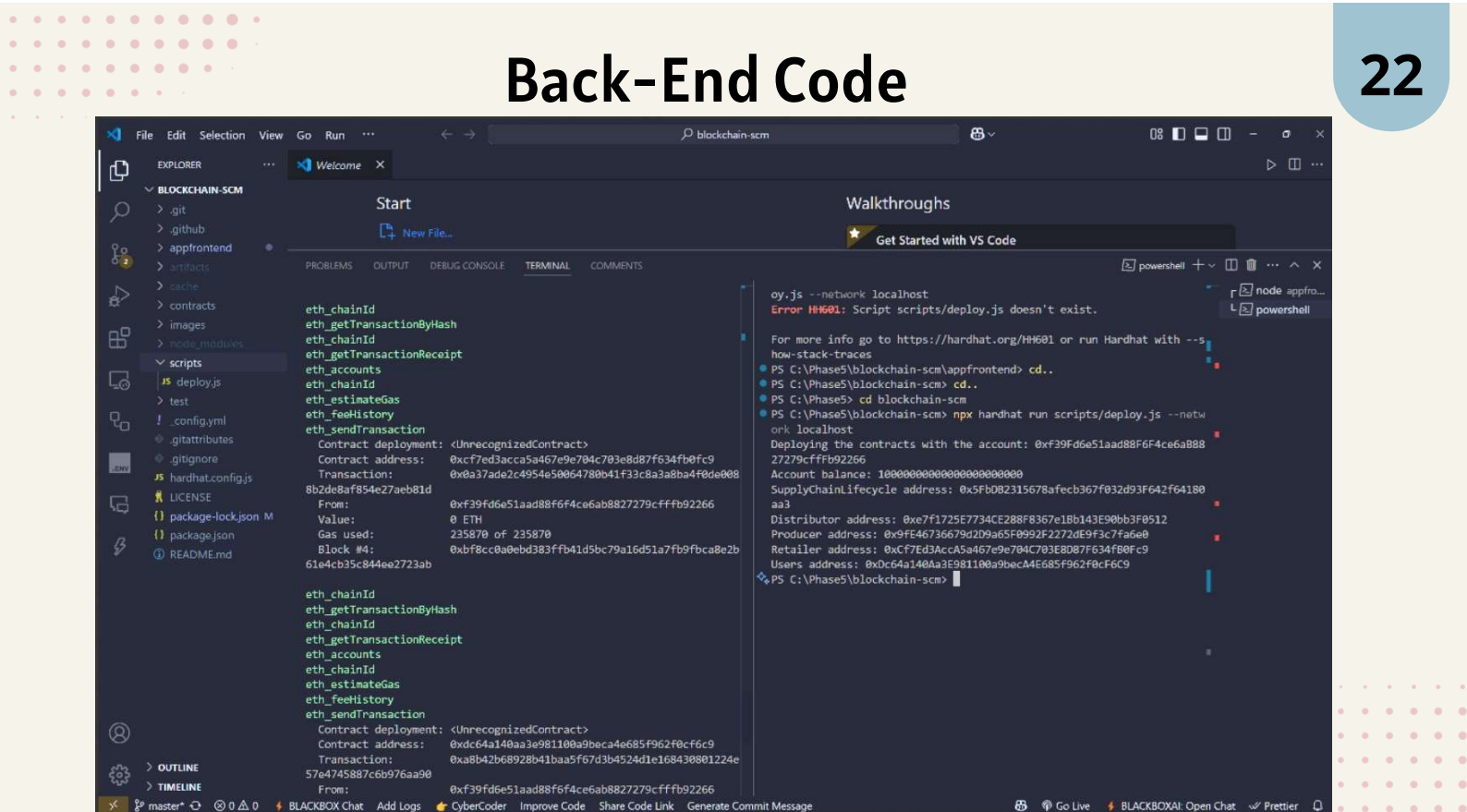
- C/C++: Intermediate proficiency in problem solving and algorithms.
- SQL: Intermediate proficiency in relational database querying and management.
- Python: Java: Beginner proficiency with foundational programming concepts.
- MongoDB: Beginner proficiency in schema modelling and unstructured data management.
- R: Basic knowledge in statistical computing and data analysis.

**Soft Skills**

- Team Player: Collaborates effectively to achieve project goals.
- Analytical Thinking: Breaks down problems into manageable solutions.
- Quick Learner: Adapts rapidly to new technologies and tools.
- Critical Thinking: Evaluates information to make sound decisions.



# Back-End Code






## Conclusion

23


The project aims to develop a **blockchain-based supply chain management system** that leverages **smart contracts**, **IPFS**, and **secure authentication**. By integrating these technologies, the system will enhance transparency, data integrity, and efficiency in managing supply chain processes. The decentralized approach ensures that all transactions and data exchanges are immutable, tamper-proof, and verifiable in real-time. The use of **smart contracts** automates workflows, reducing errors and delays, while **IPFS** and **NDN** improve storage scalability and data access speed. This system promises better stakeholder collaboration and trust in supply chain operations.





## References

24

- Yigit, E. and Dag, T., 2024. Improving Supply Chain Management Processes Using Smart Contracts in the Ethereum Network Written in Solidity. *Applied Sciences*, 14(11), p.4738.
  - Ahmad, A.Y.B., Verma, N., Sarhan, N., Awwad, E.M., Arora, A. and Nyangaresi, V.O., 2024. An IoT and Blockchain-Based Secure and Transparent Supply Chain Management Framework in Smart Cities Using Optimal Queue Model. *IEEE Access*.
  - Lamrani Alaoui, H., El Ghazi, A., Zbakh, M., Touhafi, A. and Braeken, A., 2021. A Highly Efficient ECC-Based Authentication Protocol for RFID. *Journal of Sensors*, 2021(1), p.8876766.
  - Kemmoe, V.Y., Stone, W., Kim, J., Kim, D. and Son, J., 2020. Recent advances in smart contracts: A technical overview and state of the art. *IEEE Access*, 8, pp.117782-117801.
  - Kang, P., Yang, W. and Zheng, J., 2022. Blockchain private file storage-sharing method based on IPFS. *Sensors*, 22(14), p.5100.
- 

The background features a light beige gradient. On the left, there are three vertical stripes: a wide red one, a medium blue one, and a narrow tan one. On the right side, there are two rectangular areas filled with a pattern of small, light red dots.

**THANK YOU**

## **Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes**

## **Vision, Mission, Programme Outcomes and Course Outcomes**

**Vision:** To become a Centre of Excellence in Computer Science & Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

**Mission:** To inspire and nurture students, with up-to-date knowledge in Computer Science & Engineering, Ethics, Team Spirit, Leadership Abilities, Innovation and Creativity to come out with solutions meeting the societal needs.

### **Programme Outcomes (PO)**

**PO1:** Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO2:** Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO3:** Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4:** Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5:** Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO6:** The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO7:** Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO8:** Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO9:** Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

**PO10:** Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

**PO11:** Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

**PO12:** Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

### **Programme Specific Outcomes (PSO)**

**PSO1:** Computer Science Specific Skills: The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

**PSO2:** Programming and Software Development Skills: The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

**PSO3:** Professional Skills: The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

### **Course Outcomes (CO)**

**CO1:** Model and solve real world problems by applying knowledge across domains.

**CO2:** Develop products, processes or technologies for sustainable and socially relevant applications.

**CO3:** Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks.

**CO4:** Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms.

**CO5:** Identify technology/research gaps and propose innovative/creative solutions.

**CO6:** Organize and communicate technical and scientific findings effectively in written and oral forms.



## Appendix C: CO-PO-PSO Mapping

## CO-PO and CO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	2	2	2	1	2	2	2	1	1	1	1	2	3		
CO2	2	2	2		1	3	3	1	1		1	1		2	
CO3									3	2	2	1			3
CO4					2			3	2	2	3	2			3
CO5	2	3	3	1	2							1	3		
CO6					2			2	2	3	1	1			3

3/2/1: high/medium/low