Name:Sujal Maheshbhai Patel
Enroll No. :202203103510166
Sem .& Div: 6(A)

Supply Chain Management Using Smart Contracts

Introduction

Supply chain management (SCM) is a critical process that ensures the smooth flow of goods and services from suppliers to consumers. Traditional supply chains rely on centralized systems that often suffer from inefficiencies, lack of transparency, and fraud. Blockchain technology, with its decentralized and immutable nature, provides a revolutionary way to enhance trust, security, and automation in supply chains. Smart contracts, which are self-executing contracts with predefined rules, play a key role in ensuring seamless operations by automating transactions and enforcing agreements without intermediaries.

Problem Statement

The traditional supply chain faces several challenges, including:

- **Lack of Transparency**: Stakeholders often struggle to track the movement and origin of goods.
- **Manual and Paper-Based Processes**: Many processes involve extensive paperwork, leading to delays and errors.
- **Fraud and Counterfeiting**: The inability to verify product authenticity results in counterfeit goods entering the market.
- **Inefficient Dispute Resolution**: Disputes between suppliers, manufacturers, and retailers take time to resolve due to a lack of reliable records.
- **Limited Trust Among Participants**: Centralized control leads to trust issues, as data can be manipulated.

Proposed Solution

The proposed solution integrates **Blockchain-based Smart Contracts** into the supply chain to:

- **Ensure Transparency**: Every transaction is recorded on the blockchain, accessible to all authorized parties.
- **Automate Processes**: Smart contracts handle payments, shipments, and approvals based on predefined conditions.
- **Enhance Security**: Data stored on the blockchain is tamper-proof, reducing fraud and unauthorized modifications.
- **Improve Efficiency**: Automated workflows reduce reliance on intermediaries and paperwork, leading to faster processing.

Name:Sujal Maheshbhai Patel
Enroll No. :202203103510166
Sem .& Div: 6(A)

System Architecture

Components:

1. **Blockchain Network**: The decentralized ledger where transactions are recorded.
2. **Smart Contracts**: Self-executing contracts handling supply chain rules and automation.
3. **Participants**:
    - **Suppliers**: Provide raw materials.
    - **Manufacturers**: Convert raw materials into finished products.
    - **Distributors**: Transport products.
    - **Retailers**: Sell products to consumers.
    - **Consumers**: End users of the products.
4. **IoT Sensors** (Optional): Track real-time location and conditions of goods.
5. **User Interface**: A dashboard for stakeholders to monitor supply chain activities.

Interaction Flow:

1. **Supplier dispatches raw materials** → Transaction is recorded on the blockchain.
2. **Manufacturer receives materials** → Smart contract verifies authenticity.
3. **Products are shipped by distributor** → Smart contract updates shipment status.
4. **Retailer receives products** → Payment is automatically released via smart contract.
5. **Consumer purchases product** → Product authenticity is verified via blockchain.

Workflow

1. **Supplier uploads product details (batch number, quantity, quality check) to blockchain.**
2. **Smart contract validates supplier credentials and stores the data.**
3. **When the manufacturer receives the goods, the smart contract updates the status.**
4. **The manufacturer processes the product and updates the blockchain with the final product details.**
5. **Distributors receive goods, and their tracking details are recorded using IoT sensors (optional).**
6. **Retailers receive products, confirm receipt, and the smart contract releases payments automatically.**
7. **Consumers can verify product authenticity using a blockchain explorer.**

Name:Sujal Maheshbhai Patel
Enroll No. :202203103510166
Sem .& Div: 6(A)

Implementation :

We use **Ethereum** and **Solidity** for smart contracts along with **Python** for interaction.

Solidity:

```
pragma solidity ^0.8.0;

contract SupplyChain {
   struct Product {
      uint id;
      string name;
      address owner;
      bool delivered;
   }
   mapping(uint => Product) public products;
   uint public productCount;

   function addProduct(string memory name) public {
      productCount++;
      products[productCount] = Product(productCount, name, msg.sender, false);
   }

   function transferOwnership(uint id, address newOwner) public {
      require(products[id].owner == msg.sender, "Not owner");
      products[id].owner = newOwner;
   }

   function markDelivered(uint id) public {
      require(products[id].owner == msg.sender, "Not owner");
      products[id].delivered = true;
   }
}
```

Python:

```
from web3 import Web3

# Connect to blockchain

w3 = Web3(Web3.HTTPProvider('http://127.0.0.1:7545'))

contract_address = '0xYourContractAddressHere'

abi = "Your_Contract_ABI_Here"
```

Name:Sujal Maheshbhai Patel
Enroll No. :202203103510166
Sem .& Div: 6(A)

```
contract = w3.eth.contract(address=contract_address, abi=abi)

# Adding a product

def add_product(name):

    tx = contract.functions.addProduct(name).transact({'from': w3.eth.accounts[0]})

    w3.eth.wait_for_transaction_receipt(tx)

    print(f"Product '{name}' added successfully")
```

Explanation of Code

1. **Smart Contract:**
    - Defines a product structure containing ID, name, owner, and delivery status.
    - Provides functions to add a product, transfer ownership, and mark delivery.
2. **Python Script:**
    - Connects to the Ethereum blockchain.
    - Calls the addProduct function to register a product.
    - Executes transactions and waits for confirmation.

Challenges and Considerations

- **Scalability**: Blockchain networks can experience latency issues during high transaction volumes.
- **Cost**: Smart contract execution involves gas fees, which can be costly on Ethereum.
- **Interoperability**: Integration with existing supply chain systems requires standardization.
- **Legal and Compliance Issues**: Regulations surrounding blockchain and smart contracts vary across jurisdictions.
- **Data Privacy**: Sensitive business information needs to be protected while maintaining transparency.

Name:Sujal Maheshbhai Patel
Enroll No. :202203103510166
Sem .& Div: 6(A)

Future Scope

- AI and IoT Integration: Use AI for demand forecasting and IoT for real-time tracking.
- Cross-Chain Solutions: Enable interoperability between different blockchains.
- NFT-based Supply Chain: Use Non-Fungible Tokens (NFTs) for product authenticity verification.
- Smart Payments: Implement automated payments using stablecoins or Central Bank Digital Currencies (CBDCs).
- Regulatory Compliance: Enhance compliance by integrating legal frameworks into smart contracts.

Summary

Blockchain-powered **smart contracts** revolutionize **supply chain management** by automating transactions, enhancing transparency, and reducing fraud. This project demonstrates how smart contracts streamline operations using a decentralized approach. While challenges exist, integrating blockchain with IoT, AI, and other emerging technologies can further enhance supply chain efficiency. The proposed Python-based implementation provides a fundamental framework that can be expanded into a full-scale blockchain-based supply chain solution.