جامعة الملك فهد للبترول والمعادن
King Fahd University of Petroleum & Minerals

Information & Computer Science Department, King Fahd University of Petroleum and Minerals

ICS 440: Cryptography and Blockchain Applications

Section 1

Project Final Report

Project Title: Blockchain-Based Supply Chain Tracking System (BSTS)

Instructor: Dr. Sultan Almuhammadi

December 10, 2025

SID	ID	NAME
[redacted]		WALEED ALRAJHI
		ALI MASHNI
		MOHAMMED ALAMRI

- Leader is highlighted.

Introduction

Transparency is a major key in supply chains because it refers to the extent to which all stakeholders, suppliers, distributors and even customers can see and verify the flow of goods, information and processes across the entire supply chain. This gives the user a chance to visible the product movement, accuracy and integrity of the data. This major key prevents fraud, counterfeiting and builds trust in the network. Fraud and counterfeiting are affecting the process because it might lead to manipulation of records, or substitution of personal goods. Transparency and clearance in the supply chain prevents these and eliminates the trust issues that might arise.

Among the features that blockchain should account for is the immutability and the traceability of a record. Blockchain provides immutability by maintaining the recorded data on the blockchain unchanged and cannot be deleted. This ensures a tamper-proof log of all supply chain events, from raw material sourcing to final product delivery. Additionally, blockchain provides traceability by allowing each record or item in our case to be tracked at every stage using a unique digital identity. All transactions and movements are recorded and securely linked to previous records which create eventually a trusted history for participants.

Transparency is achieved using a blockchain through decentralized data storage by giving stakeholders a shared view of the supply chain. No need to rely on a single authority, all parties can verify the integrity and authenticity of data independently. As mentioned in the first paragraph, this visibility reduces disputes, increases overall confidence and improves coordination in the system.

Key benefits of the BSTS include improving transparency, increasing the trust between participants and the system, reducing the fraud and counterfeiting, enhancing the traceability and immutability of the records. The purpose of it is to address challenges of trust by establishing a transparent, secure and trustworthy supply chain ecosystem.

System Design and Architecture

This project implements a simple supply-chain tracking dApp where producers publish products on Ethereum, suppliers/retailers purchase them, and consumers can verify the product history by scanning a QR code.

1. Workflow: producer–supplier–consumer–smart contract

The end-to-end workflow of the proposed BSTS is illustrated in Fig. 1.1.

At a high level:

- **Product registration:** The producer opens the Add Product page, enters product details, and confirms a registerProduct transaction via MetaMask. The smart contract stores the product on-chain, and the app later generates a QRcode linked to its identifier.
- **Purchase and ownership transfer:** The supplier views the list of approved products and clicks Buy. The dApp sends a payable transferProduct transaction through MetaMask. On success, the contract updates the product owner and credits the seller's internal balance.
- **QR-based verification:** The consumer scans the QR code, which opens a product page containing the product identifier. The dApp calls getProductHistory(productId) and displays the origin and transfer history returned by the smart contract.
- **Balance withdrawal:** A producer or supplier can withdraw their accumulated balance by clicking Withdraw Balance. The dApp issues a withdrawBalance transaction via MetaMask, and the contract transfers the corresponding Ether to the user's wallet.

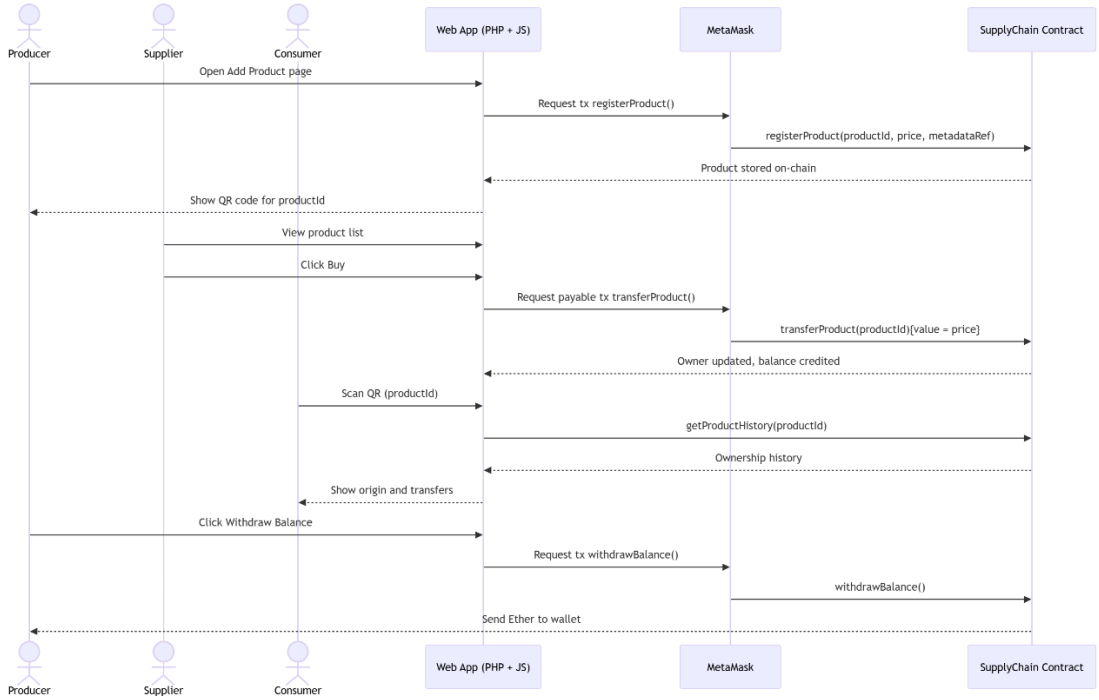


Fig. 1.1. Sequence diagram of the BSTS workflow showing interactions between producer, supplier, consumer, web application, MetaMask, and the supply-chain smart contract.

2. Functional Modules

The Blockchain-Based Supply Chain Tracking System (BSTS) is decomposed into a small set of modules; the main components and their interactions are summarized in Fig. 1.2.

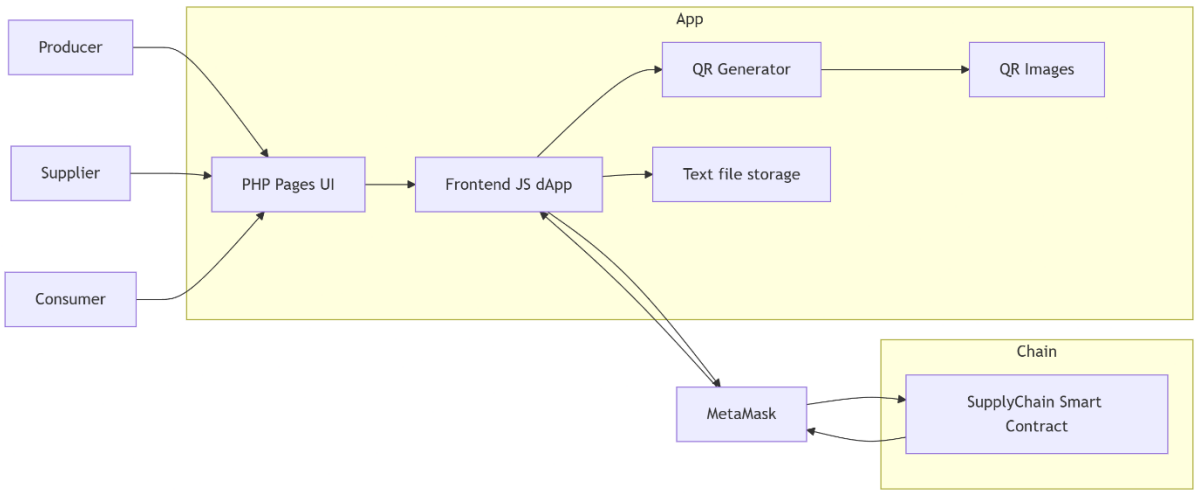


Fig. 1.2. System architecture and main functional modules.

- **User layer:** Producers, suppliers/retailers, and consumers access the system through a standard web browser.
- **Client (browser) layer**
  - **PHP Pages UI:** screens for login, product registration, product details…
  - **Frontend JS dApp:** connects the UI to MetaMask, sends registerProduct, withdrawBalance, and performs read-only calls such as getProductHistory.
- **Blockchain access layer (MetaMask):** Manages user accounts, signs transactions and forwards them to the Ethereum testnet.
- **QR and file layer:** The QR generator creates QR codes for registered products, and QR storage keeps the images so they can be shown or printed.

3. Technology stack

- **Blockchain layer**
  - Ethereum-compatible test network (Sepolia)
  - Smart contracts written in **Solidity** and deployed via **Remix IDE**
- **Wallet and dApp interaction**
  - **MetaMask** browser extension for transaction signing
  - **Sepolia Etherscan** to verify and visualize transactions.
- **Web and backend**
  - **XAMPP** stack running locally: Apache + PHP on localhost
  - PHP for routing, simple authentication, QR generation
  - Text files will be used for data storage

Implementation and Results

1 Smart Contract Implementation

The core logic of the Blockchain-Based Supply Chain Tracking System (BSTS) is implemented in a custom Solidity smart contract named **ProductsChain**, deployed on the Ethereum Sepolia test network. The contract defines the full lifecycle of a product, from creation by a producer, to transfer to a supplier, and finally to purchase by a consumer, while maintaining a tamper-proof immutable record on the blockchain.

1.1 Role-Based Access Control

```
enum Role { None, Producer, Supplier, Consumer }
```

Roles are assigned once by the admin using setRole() inside Remix:

- **Producer**: can create products
- **Supplier**: can purchase products from producers
- **Consumer**: can purchase products from suppliers

This ensures that each operation respects business logic, and unauthorized actions are rejected with meaningful error messages.

1.2 Product Registration

```
function addProduct(    infinite gas
    string calldata metaHash,
    uint256 price,
    uint256 qty
) external {
```

Key behaviors:

- Only **Role.Producer** account can call it.
- A new product is created with owner = producer
- Timestamps stored (createdAt, updatedAt)

1.3 Supplier Purchase

```
function paySupplier(uint256 productId, uint256 qty)
```

Key behaviors:

- Ensures caller has **Role.Supplier.**
- Validates that the requested quantity does not exceed available stock.
- Updates timestamps

1.4 Consumer Purchase

```
function payConsumer(uint256 id, uint256 qty)
```

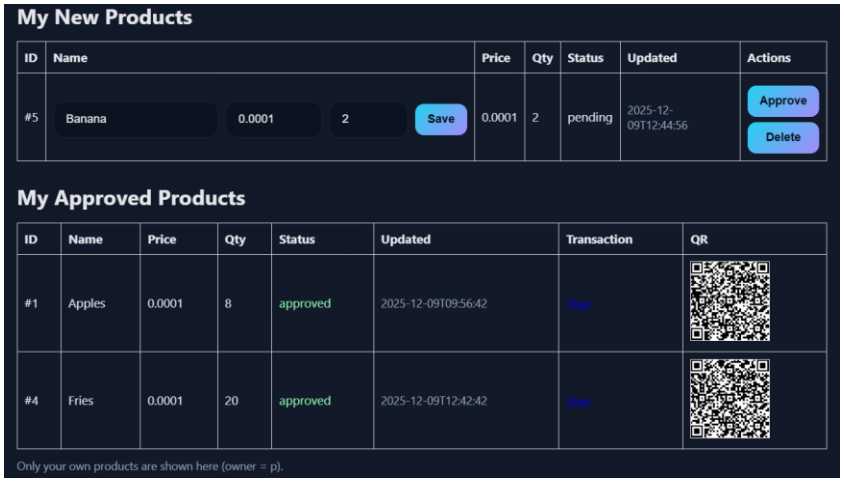
Key behaviors:

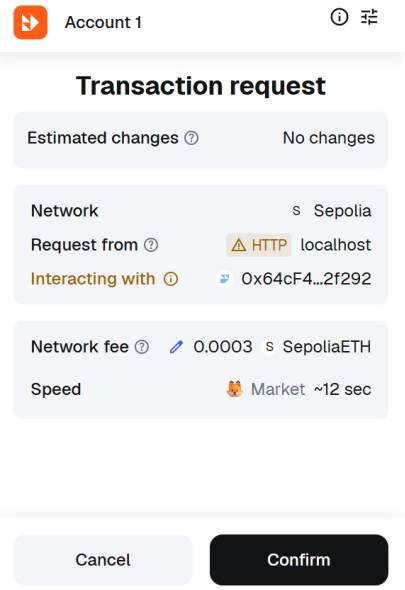
- Ensures caller has **Role.Consumer**
- Modifying quantity and validating payments.
- Credits the supplier's on-chain balance

2 Results and UI Demonstration

2.1 Producer Dashboard

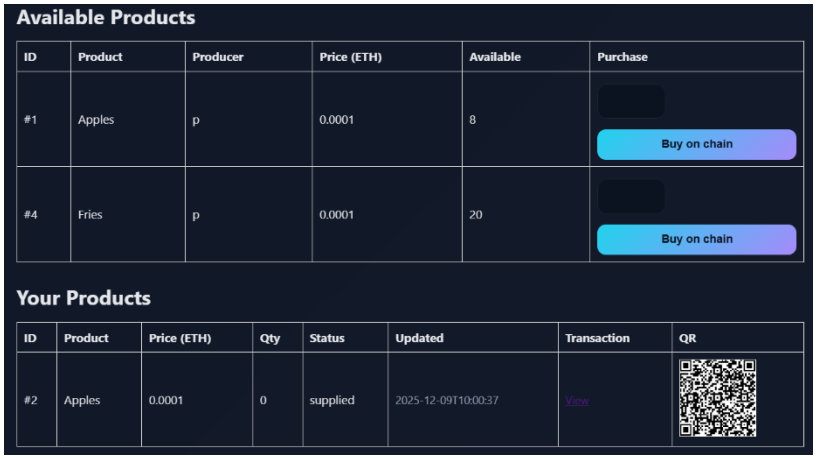
The dashboard contains two main tables, the first table for pre-approved products. Approving products require a Role.Producer and Metamask transaction confirmation to be saved in the blockchain and moved to the second table which contains details about the product, Sepolia transaction and a QR code for product information as shown in the following screenshots.

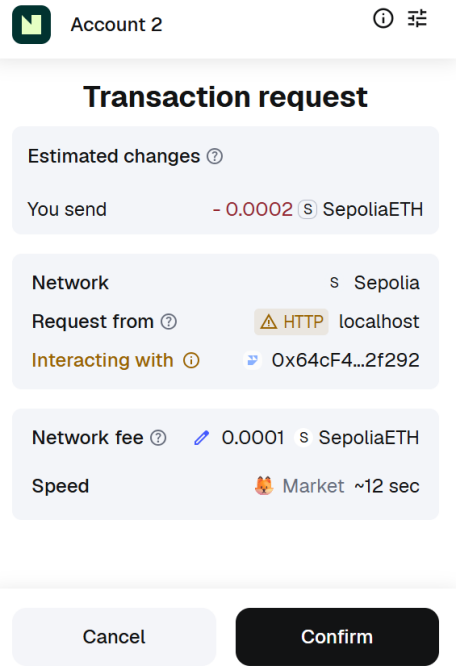




2.2 Supplier Dashboard

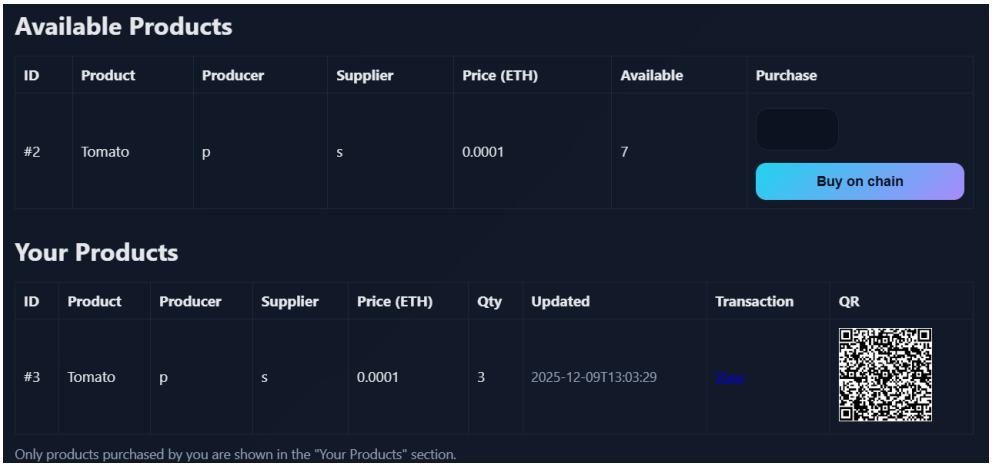
In order to complete the cycle of the blockchain, supplier dashboard mostly contains the same elements as for the producer. However, the major difference is in the display criteria. The first table contains products that have been approved by the producer, and the supplier can buy them if needed. The second table contains products that the supplier has bought and confirm the purchase through Metamask account. Similarly, products in supplier's second table show details of product, Sepolia transaction and QR code for product details and status as shown in the following screenshots.

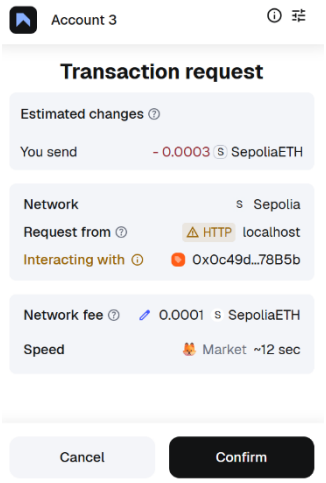




2.3 Consumer Dashboard

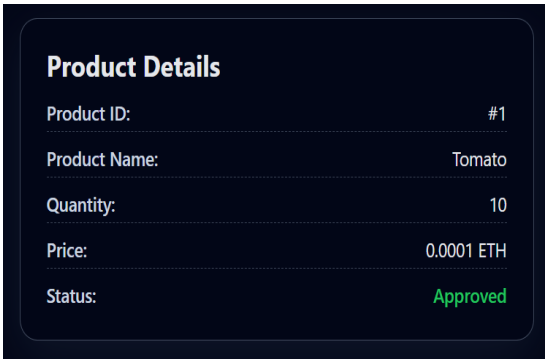
Products that have been approved to be supplied by the suppliers will appear in the consumer dashboard to be purchased. Once it is purchased using a Metamask account confirmation for payment, it will be considered as a consumed product and move to the second table where details such as product information, Sepolia transaction and QR code for product information are displayed as shown in the following screenshots.





2.4 QR Code Implementation

QR code is implemented as a deployed single page application for displaying product details. QR code generated during the purchasing, supplying, consuming activites and being attached to the product. The following screenshot is a demonstration of the UI of the QR code result generally.



- Additionally the transaction for each product that has been created, supplied or consumed can be checked through Sepolia transaction website which ensures product traceability and an example for one of the transactions can be checked through this link: https://sepolia.etherscan.io/tx/0xad5e1260b415325d773a86ae5e3c592a62b2d17480bedc7d4b08e81623497c97
- For full project implementation can be found through our GitHub repository link: https://github.com/Ali-Mashni/Blockchain-Based-Supply-Chain-Tracking-System

Discussion and Conclusion

The Blockchain-Based Supply Chain Tracking System (BSTS) relies fundamentally on several cryptographic mechanisms that are embedded within the Ethereum blockchain and its supporting tools. At the core of BSTS is public-key cryptography, which underpins the identity and authentication of every participant in the system. Each producer, supplier, and consumer interacts with the system using an Ethereum wallet where actions, such as adding a product, purchasing it, or withdrawing balances, are authorized through cryptographic digital signatures. These signatures ensure that only legitimate users can modify on-chain data and that every transaction is non-repudiable. It was specifically mentioned in the results section that each role has his own specific actions which ensures authenticity.

Additionally, BSTS benefits from the built-in hashing functions used by blockchain networks. Once a product is added to the blockchain, its associated metadata (price, quantity, metaHash) becomes part of an irreversible transaction record. This hashing provides strong integrity guarantees, meaning that any attempt to modify stored product data would break the cryptographic link between blocks, making tampering immediately detectable.

Although BSTS successfully demonstrates decentralized traceability and transparency, several enhancements could improve scalability, usability, and robustness. One promising enhancement involves introducing role-based automation with smart-contract events, enabling external systems to listen for blockchain triggers. For example, when a supplier purchases a product, an automated notification service. This makes BSTS more practical for real-world supply chains, where synchronization with existing software systems is essential.