Mid-Term Report: Supply Chain Tracking on the Blockchain

17.04.2024

Team Members

Lazaro Nicolas Hofmann, 19-731-306

Lennart Paul Eikens, 21-933-106

Jingyi Chen, 22-739-478

Xirui Yang, 23-756-265

Saksham Joshi, 24-753-394

Problem Statement

The cocoa bean supply chain is a central component of the global agricultural and food production system. It links cocoa producers, who are predominantly located in the tropics, with export markets where the cocoa is transformed into a wide variety of products, the majority of which are chocolate. The supply chain supports a complex network of actors, including farmers, cooperatives, traders, processors, manufacturers, and retailers, all involved in the movement and processing of cocoa from production to final consumption.

Cocoa is a commodity that has significant economic significance on the global platform. It is a critical component of the economies of several producing countries and provides livelihoods to millions. With growing world demand for chocolate and cocoa products, the effectiveness, coordination, and traceability of the cocoa supply chain increase in significance. The supply chain not only facilitates international trade but also enables quality control, product consistency, and innovation in food production.

Traditional supply chain systems often face challenges such as lack of transparency, inefficient record-keeping, and the risk of fraud or mislabeling. This project aims to address these issues by developing a blockchain-based decentralized application (DApp) for a cocoa supply chain.
The following problems can be addressed using blockchain technology:

Lack of Transparency

In the cocoa industry, stakeholders often have limited visibility into the origin and handling of cocoa beans. Blockchain can enhance transparency by recording each transaction—from farm to factory—in an immutable ledger accessible to all authorized participants, including consumers.

Inefficient Record-Keeping

Much of the cocoa trade still relies on paper-based documentation or isolated digital systems, which are prone to errors, tampering, and data loss. Blockchain introduces a decentralized, tamper-proof system for storing supply chain data, ensuring the reliability and integrity of records.

Risk of Fraud and Mislabeling

Unscrupulous actors may mix low-quality or unsustainably sourced beans with certified ones, or falsify labels about origin or ethical sourcing. Blockchain enables product verification and traceability from the farmer to the end consumer, reducing the risk of fraud and helping maintain trust in ethically sourced cocoa.

Suitability of Using Blockchain in Cocoa Beans Supply Chain

Blockchain technology is particularly well-suited to address the key challenges identified in the cocoa beans supply chain due to its core features: transparency, immutability, decentralization, and traceability. These characteristics make it an ideal foundation for building trust, improving data reliability, and streamlining multi-party coordination within the supply chain.

First, blockchain significantly improves **transparency** by recording every transaction—such as harvesting, processing, shipping, and receiving—on a shared, immutable ledger. This allows all stakeholders, from cocoa farmers to chocolate manufacturers and consumers, to access accurate and up-to-date information regarding the product's journey.

Second, blockchain provides a **tamper-resistant system for record-keeping**, eliminating many of the inefficiencies and vulnerabilities of paper-based or siloed digital systems. Each entry on the blockchain is cryptographically secured and time-stamped, preventing unauthorized alterations and reducing the chances of human error or data loss.

Third, blockchain enhances **product verification and authenticity**, helping to combat **fraud and mislabeling**. By allowing each batch of cocoa beans to be digitally tagged and tracked, it becomes far more difficult for unethical actors to introduce uncertified or low-quality products into the supply chain. Consumers and regulators can verify origin, certification status, and handling processes with greater confidence.

Overall, the integration of blockchain into the cocoa supply chain offers a scalable and secure solution for addressing long-standing issues around visibility, trust, and integrity. It enables a more ethical, efficient, and transparent global cocoa trade ecosystem.

Software Development Process

We have opted for a streamlined waterfall model for developing our dApp. The process is divided into four clearly defined phases:

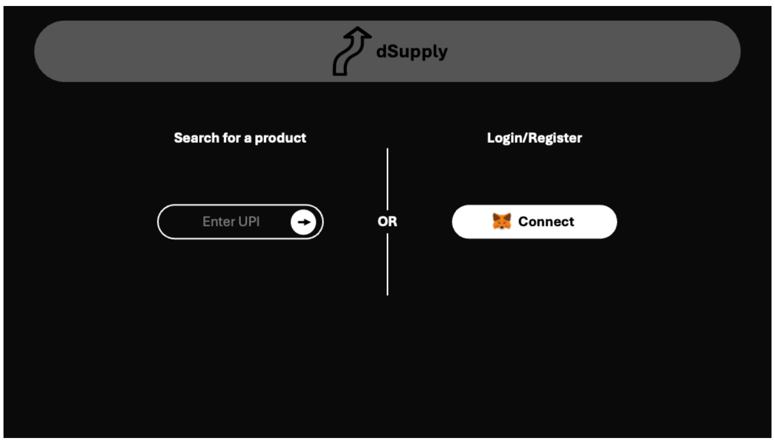
1. **Requirements:** Gather and document all system needs and stakeholder expectations.
2. **Design:** Develop detailed system architecture and design specifications.
3. **Implementation:** Build and integrate the components of the application.
4. **Testing:** Validate the functionality, performance, and security of the system.

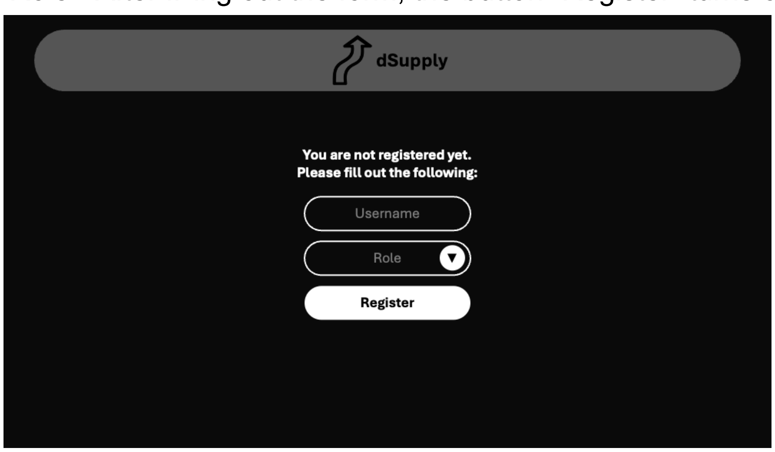
Although these phases progress in a sequential manner, the model includes the flexibility to revisit and refine any previous phase to address unforeseen issues and incorporate necessary adjustments.

Requirements

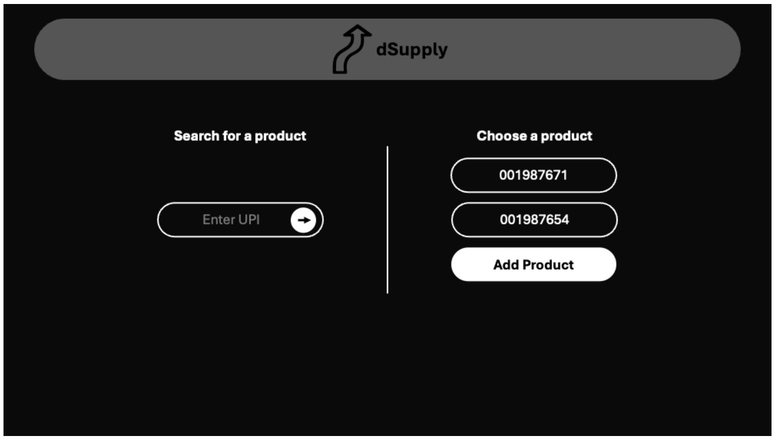
Functional Requirements

The functional requirements are written as user stories which describe what a software feature should do. Additionally, each requirement is accompanied by a UI concept design.

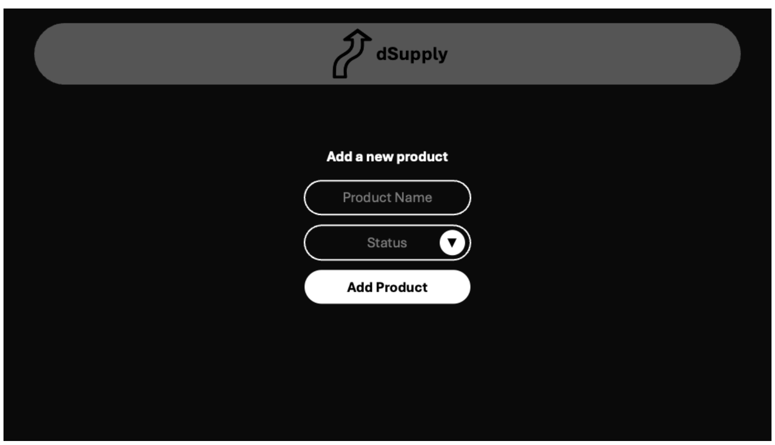
1. As a user, I want to call the web app and land on the starting page.
   Definition of Done: Either call a website or localhost.
   

2. As a user, I want to register in order to start using the app. I want to specify my role (producer, shipper, retailer, customer).
   Definition of Done: After pressing “Connect”, these input fields appear: “Username” and “Role”. After filling out the form, the button “Register” turns clickable.
   

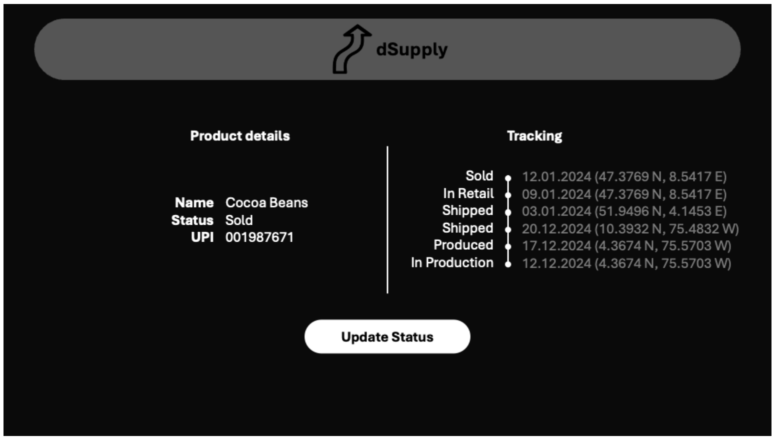
3. As a registered user, I want to login to the dApp in order to use the dApp and manage my products.
   Definition of Done: After pressing “Connect”, the dApp automatically logs the user in (by checking the user’s wallet address), and forwards the user to the product overview section.



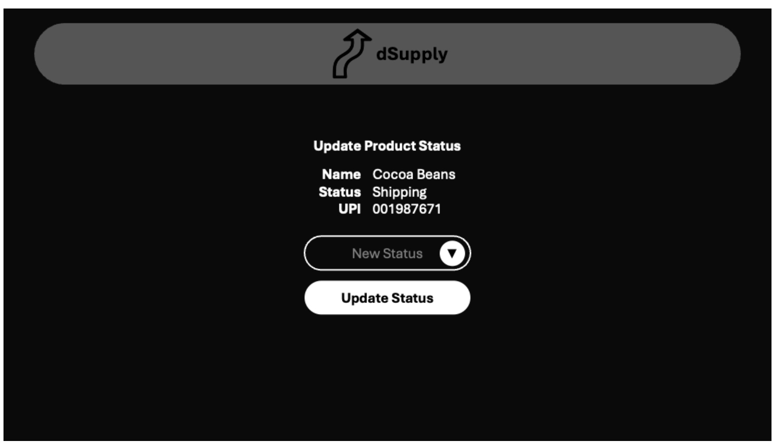
4. As a user, I want to create a new product in order to manage it.
   Definition of Done: When clicking on “Add product”, a new window loads, where the new product can be created. The UIP is created automatically.



5. As a customer, I want to enter a product's UPI in order to look up a product's supply chain route and status.
   Definition of Done: On the main page, a UPI can be entered. Afterwards, a new window pops up with the product's information (location, route, status, name, etc.).



6. As a user, I want to update a product's status and location.
   Definition of Done: Once logged in, a product's UPI (Unique Product Identifier) can be entered and then edited (location and status).



Non-Functional Requirements

1. **Decentralization**
   The dApp runs solely on the blockchain, ensuring access and possibility for modification of the data remains in case of a frontend breakdown.

2. **Integrity**
   The blockchain's immutability guarantees that once data is recorded, it cannot be tampered with.

3. **Usability**
   The application should have an intuitive and simple interface that caters to different user roles (buyers, sellers, shippers) with clear navigation and minimal training required.

Threat Model

1. **Sybil Attacks**: An attacker creates multiple fake accounts to distort reputation or flood the network.

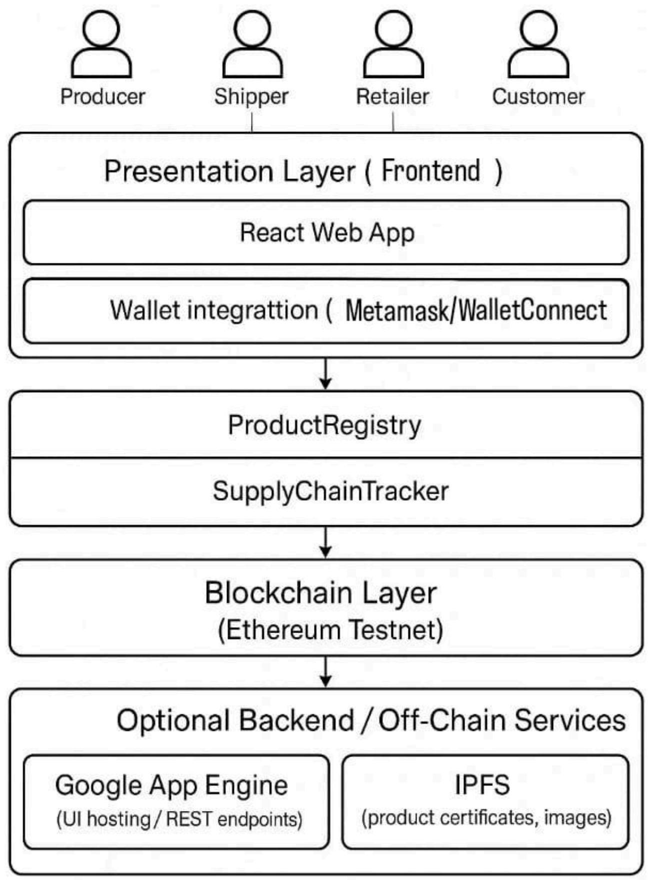
2. **Data Tampering via Off-Chain Services**: An attacker intercepts or alters information in off-chain APIs (e.g., location updates).

3. **Role Escalation**: A user attempts to bypass role restrictions to gain unauthorized access.

4. **Frontend Spoofing/Phishing**: Users may be tricked into interacting with fake interfaces.

Software Architecture

The software architecture of our decentralized supply chain tracking system is designed to ensure transparency, security, and traceability throughout the cocoa supply chain. It is divided into four layers that separate core functionalities; The **Presentation Layer,** The **Smart Contract Layer,** The **Blockchain Layer** and an **Optional Off - chain Backend Layer.**



Software Architecture

Frontend

Presentation Layer provides an intuitive and role based user interface for all stakeholders - producers, shippers, retailers, and customers. Users interact with the system through a browser and authenticate via Metamask or WalletConnect, ensuring secure identity verification using Ethereum wallet addresses.

Smart Contract

The smart contracts are deployed on a public Ethereum testnet, providing a tamper-proof, distributed ledger for storing all relevant supply chain data. Each transaction is time-stamped and permanently recorded, ensuring that the data cannot be altered or deleted. This layer guarantees immutability, data integrity and auditable traceability from the origin of the product to its final consumption.

Optional Layers

To reduce on-chain storage costs and improve scalability, we include optional off-chain components:

- IPFS (InterPlanetary File System) is used to store large files such as product certificates and images. The content hashes are stored on-chain to ensure data authenticity.
- Google App Engine hosts the frontend and can optionally serve REST APIs for enhanced functionality, dashboards, or future analytics.

The application follows a waterfall development model and is version-controlled using GitHub. A CI/CD pipeline is configured to automatically build and deploy the frontend to Google App Engine upon changes to the main branch. Testing is conducted using Hardhat for smart contracts and Jest for the frontend, ensuring robustness and correctness of the system.