**Farm Supply Chain**

**Traditional farm supply chain** **drawback:**

Traditional farm supply chains often lack transparency, making it difficult for consumers to trace the origin of food items and verify their authenticity. Without clear visibility into the entire supply chain, consumers may be left in the dark about where their food comes from and how it was produced.

Data management within traditional farm supply chains can be fragmented, with information scattered across multiple stakeholders and systems. This fragmentation makes it challenging to consolidate and access comprehensive data on the journey of food items from farm to table, hindering traceability efforts.

Due to the lack of standardized tracking mechanisms, traditional farm supply chains may struggle to provide comprehensive traceability for food items. Without robust tracking systems in place, it becomes difficult to identify and address issues related to food safety and hygiene effectively.

Traditional chains may be vulnerable to fraud and misrepresentation, where food items are falsely labeled or adulterated along the supply chain. Without robust authentication mechanisms, consumers may unknowingly purchase counterfeit or contaminated products, putting their health at risk.

**Decentralized Approach of** **Farm Supply Chain**

In the decentralized farm supply chain facilitated by the Hyperledger Fabric network, transparency and efficiency are paramount. The network comprises multiple stakeholders, including farmers, producers, distributors, and retailers, each participating in the supply chain process. Through smart contracts and distributed ledger technology, the entire lifecycle of agricultural products is recorded and managed securely.

Food quality and safety are important topics today as everyone is concerned about the quality of the food that is being consumed.

Food items like fruits and vegetables generally do not have any expiry date mentioned so it becomes really important to understand the origin of these food items and know when was it sent to the distributor from the farmer and so on. The following cycle is generally a usecase followed in the Hyperledger Fabric food supply chain:

1. Producer: The producer can harvest fruits and vegetables, sell them to a distributor, and track authenticity.
2. Distributor: The distributor can buy and distribute the fruits and track authenticity.
3. Retailer: The retailer can buy and put the fruits for sale and track authenticity.
4. Consumer: The consumer can buy the fruits and track authenticity.

**Features of the application:**

1. Farm Product has the following parameters:
2. productId: Asset Id
3. productDescription: Description of the asset
4. producerName: Name of the producer or farmer
5. producerAddress: Address of the producer or farmer
6. harvestDate: Date of the harvest
7. distributerName: Name of the distributer
8. distributerAddress: Address of the distributer
9. prodToDistDate: Date of transfer from producer to distributer
10. retailerName: Name of the retailer
11. retailerAddress: Address of the retailer
12. distToRetaDate: Date of transfer from distributer to retailer

1. Add a new farm product to the ledger 🡪 ***addNewAgroproduct***

This function is used to add a new product to the ledger. This function is called by the producer or farmer by using the below parameters.

*Input Parameters:*

*\* @param ctx the transaction context*

*\* @param id the product id of the farm product*

*\* @param description the description of the farm product*

*\* @param producerName producer or farmer name*

*\* @param producerAddress producer or farmer address*

*\* @param harvestdate harvestdate of the farm product*

*\* @return the Product details*

*\*/*

This function also checks if:

* The same asset with the same product ID does not exist already

1. Transfer the asset to the distributer from the producer 🡪changeAgroproductOwnership

This function helps to transfer the farm product from a producer (farmer) to a distributor.

Input Parameters:

\* @param ctx the transaction context

\* @param id product Id of the Farm Product

\* @param distributerName distributer name

\* @param distributerAddress distributer address

\* @param transferDate transaction date between distributer and producer

\* @return the product id

\*/

This function also does the following check:

* An asset should be present in the ledger

1. Transfer the asset to the retailer from the distributor 🡪 changeAgroproductOwnership

This function helps to transfer the farm product Ownership to the retailer from the distributor.

Input Parameters:

\* @param ctx the transaction context

\* @param id product Id of the Farm Product

\* @param retailerName retailer name

\* @param retailerAddress retailer address

\* @param transferDate transaction date between distributer and retailer

\* @return the product id

\*/

This function also does the following check:

.

* An asset should be present in the ledger.

1. View asset details from the ledger 🡪 ***queryAgroproductById***

This function helps to retrieve asset product details from the ledger.

Input Parameters

\* @param ctx the transaction context

\* @param id product Id of the farm Product

\* @return Farm Product supply chain details

\*/

**Recommended technologies:**

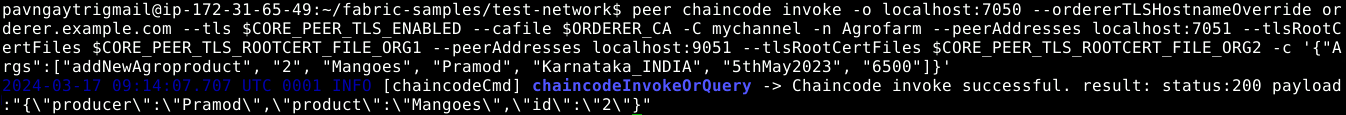
1. IDE Tool: Eclipse
2. Chaincode Language: Java
3. Build Automation tool: Gradle
4. Blockchain: Hyperledger Fabric
5. Server: Test network

**Project development guidelines:**

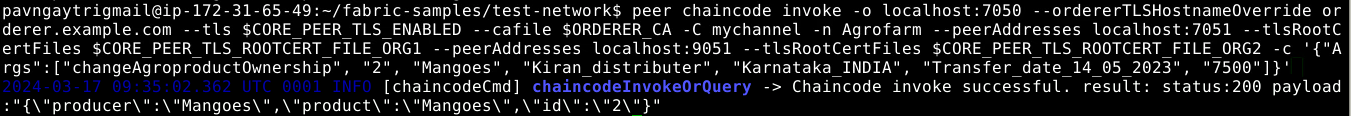
* The project will be delivered within four sprints with every sprint delivering a minimal viable product.
* It is mandatory to do proper sprint planning with user stories to develop all the components of the project.
* The learner can use any technology from the above-mentioned technologies for different layers of the project.
* The learner has to maintain the version of the application over GitHub and every new change should be sent to the repository.
* The learner should also deploy and host the application on any blockchain instance.

**Project Outputs:**

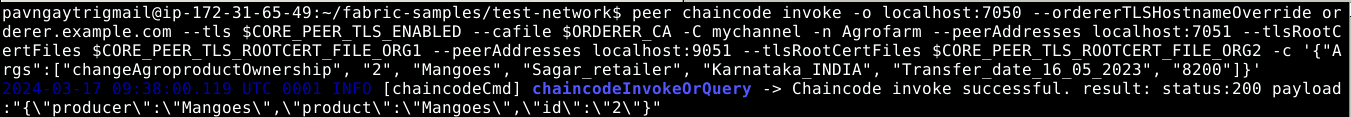
Add a new farm product asset of the producer to the ledger:



Transfer the farm product asset from producer to the distributer:



Transfer the farm product asset from distributer to the retailer:



Query the farm product by ID:

