

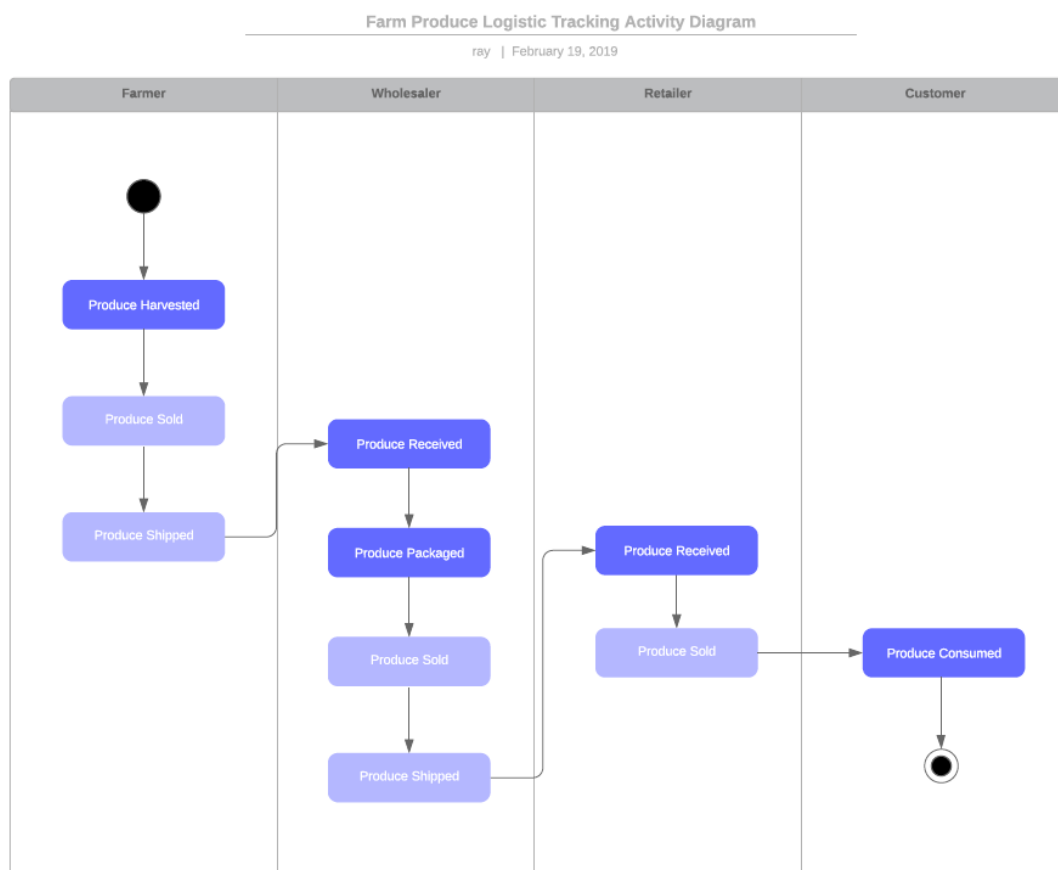
## Project Write-Up with UML Diagrams

### Overview

This Project aims to develop an Ethereum-based D-app that help track the origin of a product and its logistical path. The design of this project is rather simple but could be expanded into something more complex to allow additional functionalities.

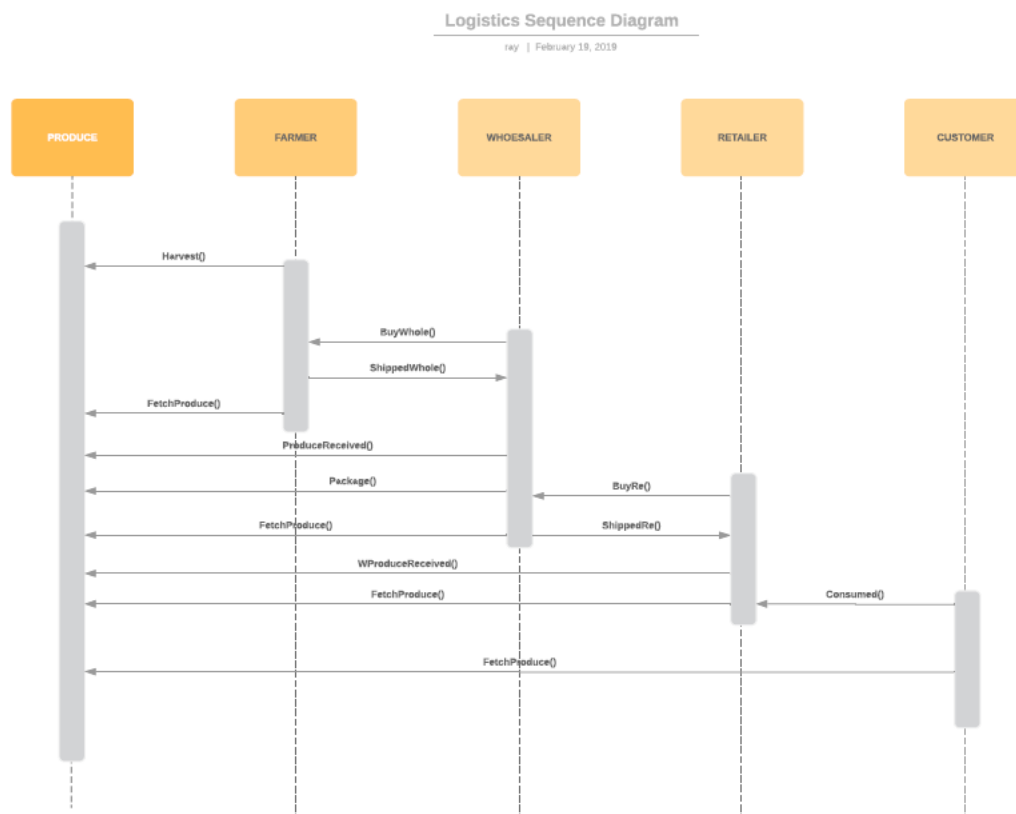
In its basic setup, the owner of the smart contract can assign 4 different roles, Farmer, Wholesaler, Retailer and Customer. Each of the 4 roles can perform different actions and the state of the farm product will change as it has been passed along different actors in the system.

### Activity Diagram



The Activity Diagram describes how the state of the product changes as it moves along the supply chain.

## Sequence Diagram

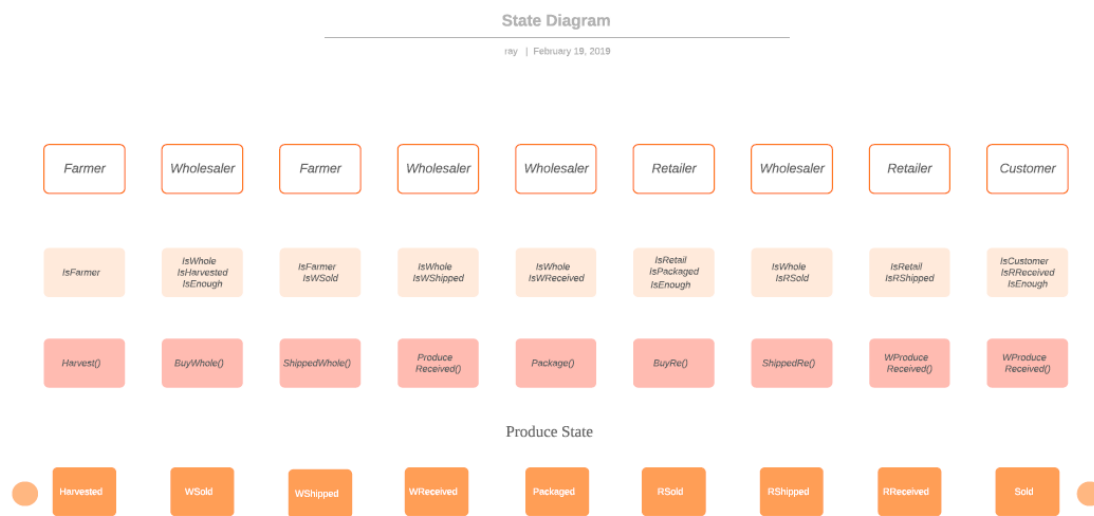


The Sequence Diagram describes the functions/actions that could be performed by the different actors.

The flow starts with the Farmer, where an item is harvested with basic farm information stored with the item on the blockchain. From there Wholesaler can buy products from farmer and packaged it for re-selling to retailer. Retailers can then purchase and on-sell to customer. The information of the different actors and the state of the product are updated as the product is being passed from one actor to another. Each actor can ship and mark receipt of the item.

Price of the product could be set to reflect the latest willingness to sell. Product information could be fetched by all actors. Two functions will be implemented – one for basic farm product information, the other for the logistical information.

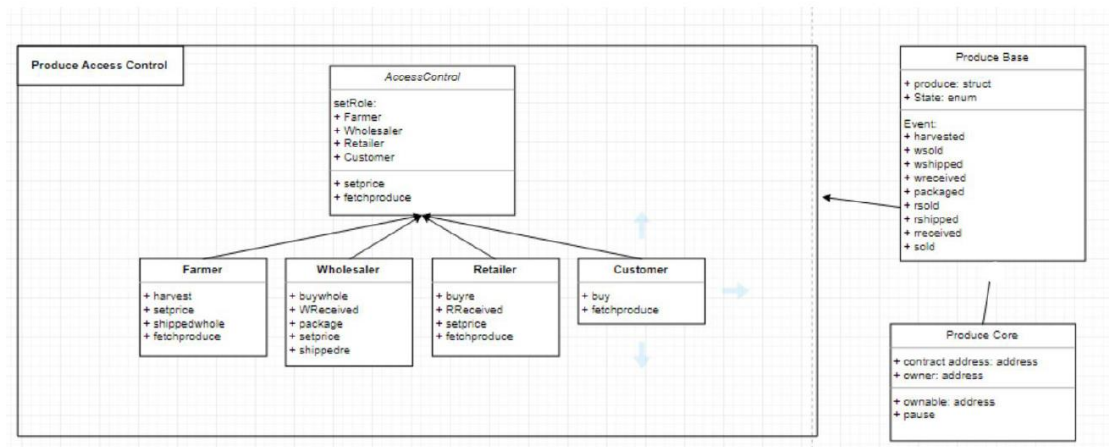
## State Diagram



The state diagram describes how the state of the product changes over the process. In the design of the D-app there are 9 different states:

- |            |                                  |
|------------|----------------------------------|
| Harvested, | - Product harvested              |
| WSold,     | - Product sold to wholesaler     |
| WShipped,  | - Product shipped to wholesaler  |
| WReceived, | - Product received by wholesaler |
| Packaged,  | - Product packaged               |
| RSold,     | - Product sold to retailer       |
| RShipped,  | - Product shipped to retailer    |
| RReceived, | - Product received by retailer   |
| Sold       | - Product sold to customer       |

## Data Diagram



The data diagram describes the structure of the smart contracts and their inter-relationship.

In our D-app, we have a Produce Access Control folder that keeps all the roles related contract. The 4 roles will each has a different contract. They all use the same library contract – Roles.sol which has the basic functionalities in common of all these contracts – add/remove/check role. Each of the Farmer, Wholesaler, Retailer & Customer role will inherit from these functions.

There is also a separate Ownable contract in the Produce Core folder which sets all the ownership related functions of a contract. This contract will allow our main contract owner to transfer the ownership right to others.

The main contract will be a SupplyChain.sol contract in the Produce Base folder. This contract will inherit from both the Ownable.sol contract and the other roles contract, allowing the owner to add/remove actors. The functions of all the actors are also implemented in this contract.