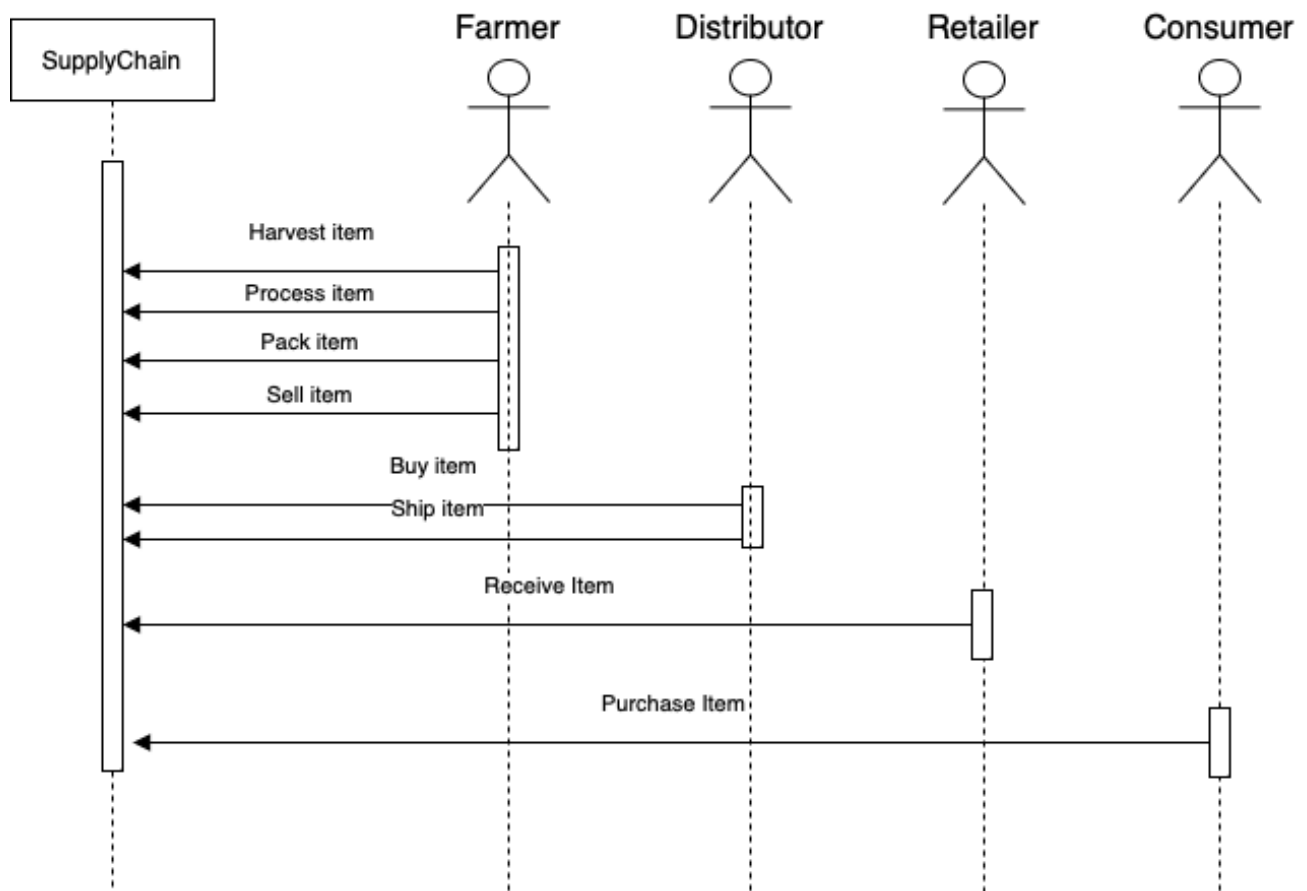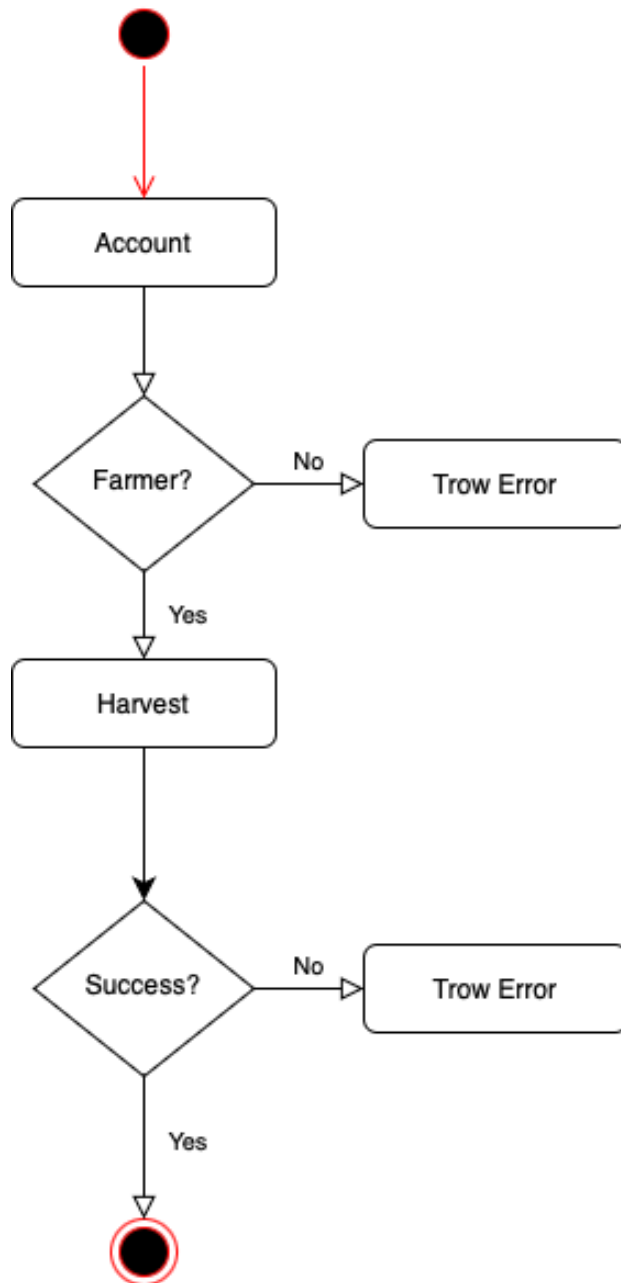# Supply Chain DAPP

This is a Dapp implementation of a simple Supply Chain.
Actors can interact with the smart contract and according to their status, can interact with an item: Modify its properties, check its status and follow its ownership.
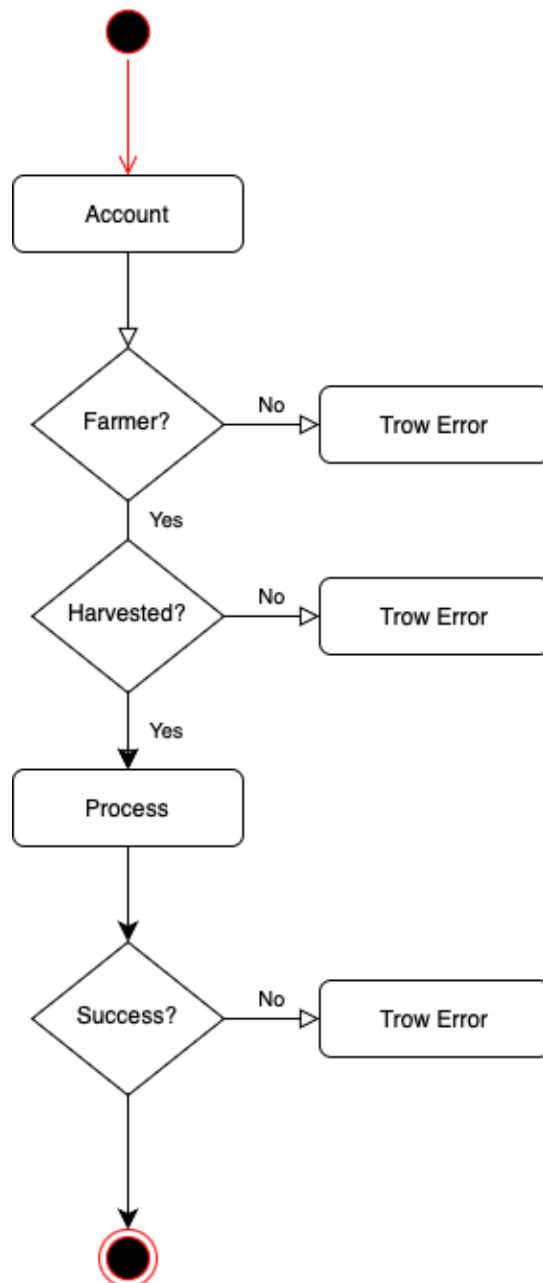Below are the sequence, activity, state and class diagrams.

Sequence Diagram

Harvest Item- Activity diagram

Account

Farmer? — No → Trow Error

Yes

Harvested? — No → Trow Error

Yes

Process

Success? — No → Trow Error

Process Item - Activity diagram

Account

Farmer? — No → Trow Error

Yes

Processed? — No → Trow Error

Yes

Pack Item

Success? — No → Trow Error
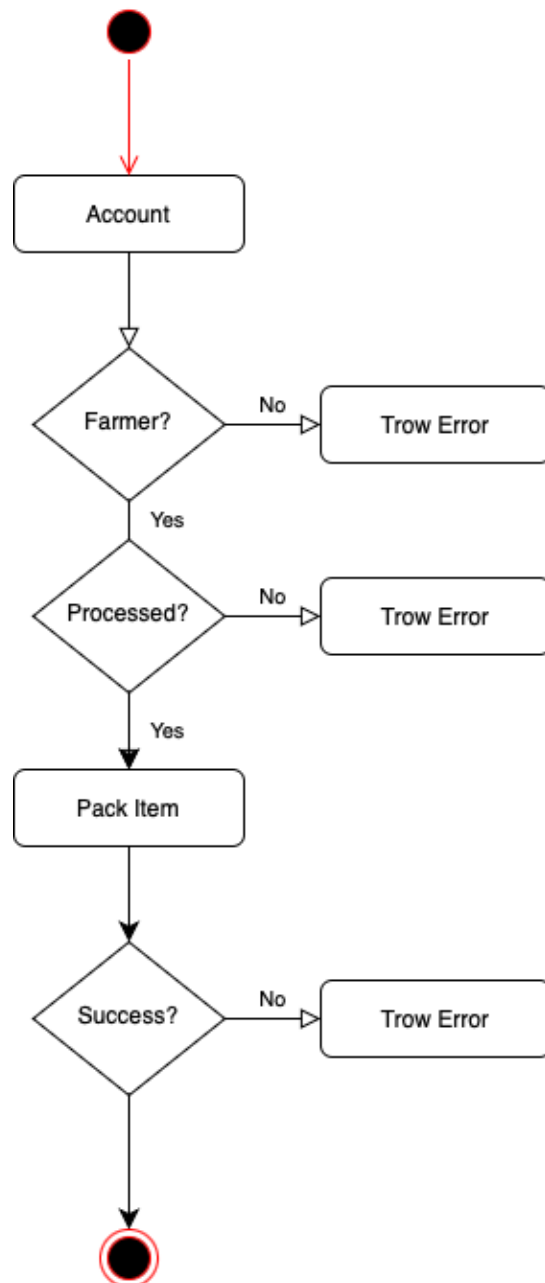
Pack Item - Activity diagram

Sell Item - Activity diagram
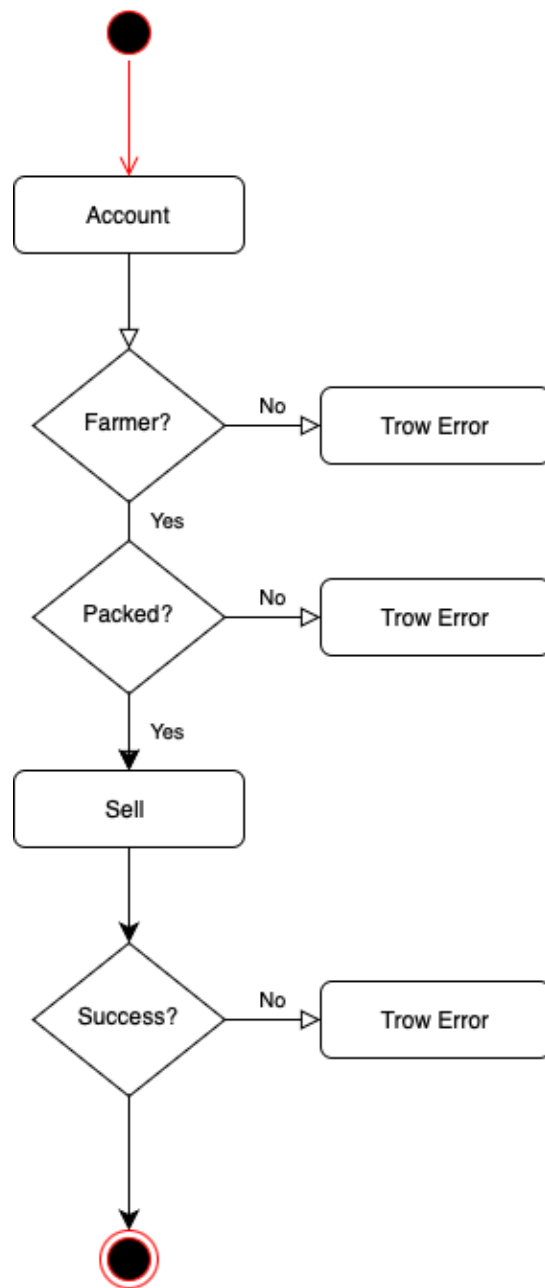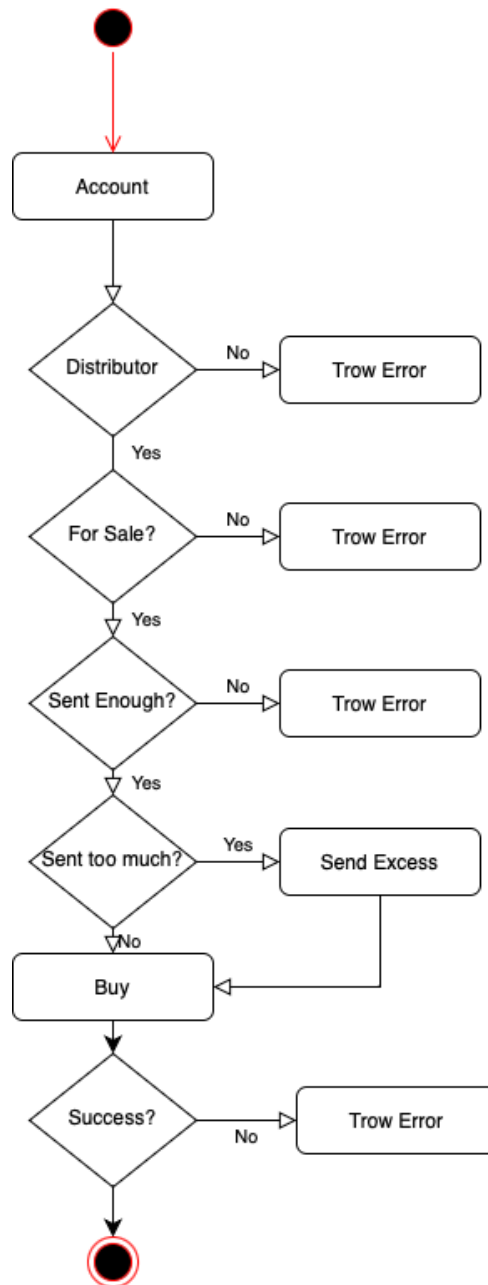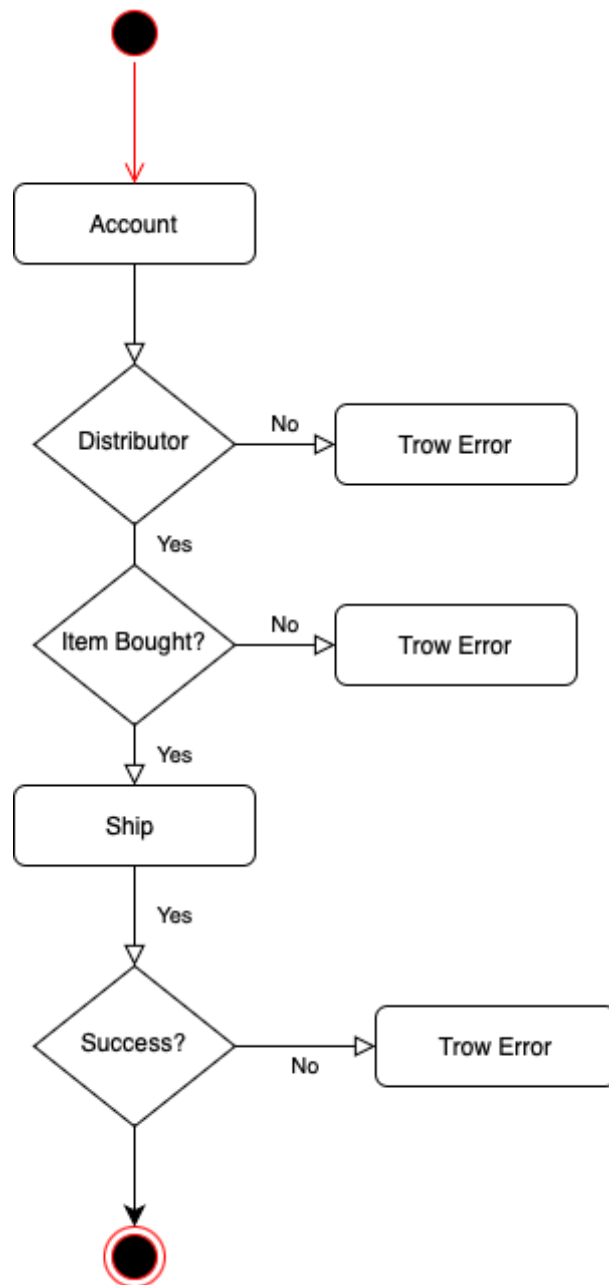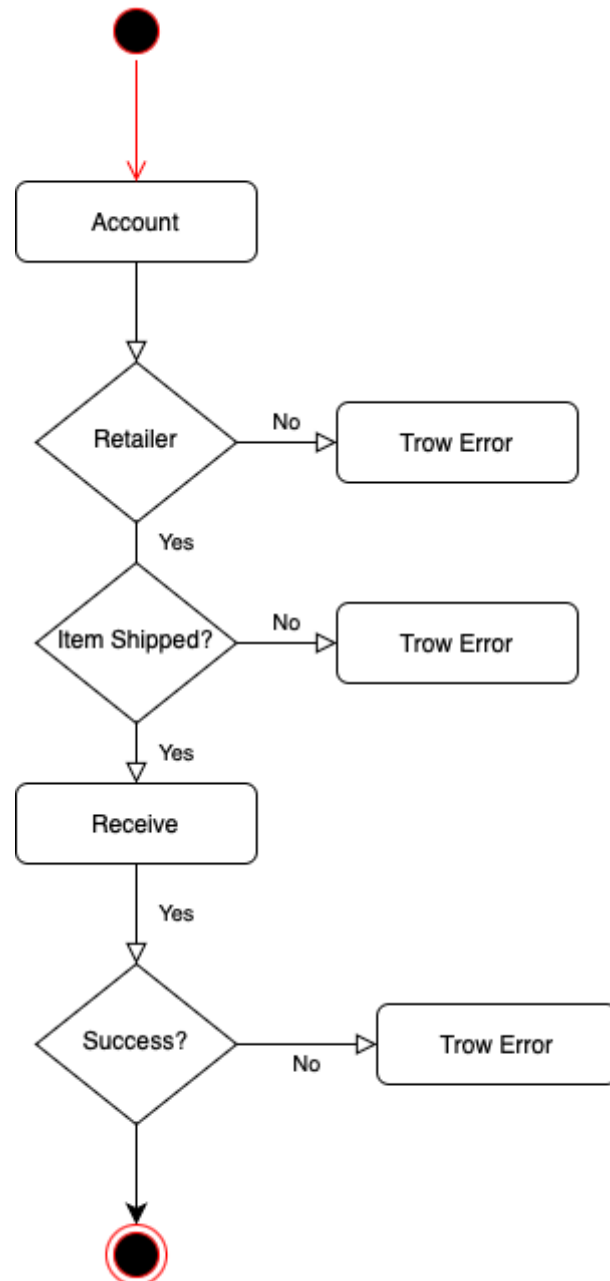
Buy Item - Activity diagram

Ship Item - Activity diagram

```
                    ●

                    │
                    ▼
            ┌───────────────┐
            │    Account    │
            └───────────────┘
                    │
                    ▼
                  ◇           No      ┌───────────────┐
              Retailer    ────────▷   │  Trow Error   │
                  ◇                    └───────────────┘
                    │
                   Yes
                    │
                  ◇           No      ┌───────────────┐
            Item Shipped?  ────────▷   │  Trow Error   │
                  ◇                    └───────────────┘
                    │
                   Yes
                    ▼
            ┌───────────────┐
            │    Receive    │
            └───────────────┘
                    │
                   Yes
                    │
                  ◇                    ┌───────────────┐
              Success?    ────────▷   │  Trow Error   │
                  ◇           No       └───────────────┘
                    │
                    ▼
                   ◉
```

Receive Item - Activity diagram
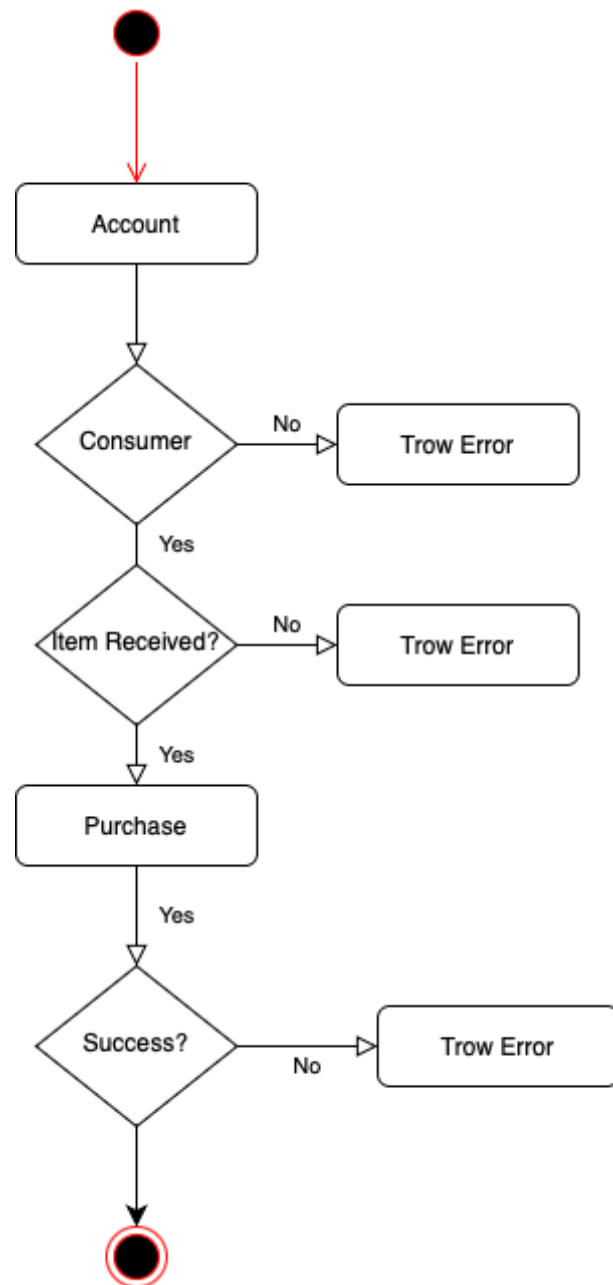
Purchase Item - Activity diagram

State Diagram

role

+ bearer: mapping (address => bool)

Role

+ role: Struct

add()
remove()
has()

Farmer

+ farmers: Role

+ isFarmer()
+ addFarmer()
+ renounceFarmer()
# _addFarmer()
# _removeFarmer()

Distributor

+ distributors: Role

+ isDistributor()
+ addDistributor()
+ renounceDistributor()
# _addDistributor()
# _removeDistributor()

Retailer

+ retailers: Role

+ isRetailer()
+ addRetailer()
+ renounceRetailerr()
# _addRetailer()
# _removeRetailer()

Consumer

+ consumers: Role

+ isConsumer()
+ addConsumer()
+ renounceConsumer()
# _addConsumer()
# _removeConsumer()

Ownable

+ originOwner: address

+ owner()
+ isOwner()
+ transferOwnership()
# _transferOwnership()

SupplyChain

+ sku: uint
+ items: mapping (uint => Item)
+ Item: struct
+ State: enum

+ harvestItem()
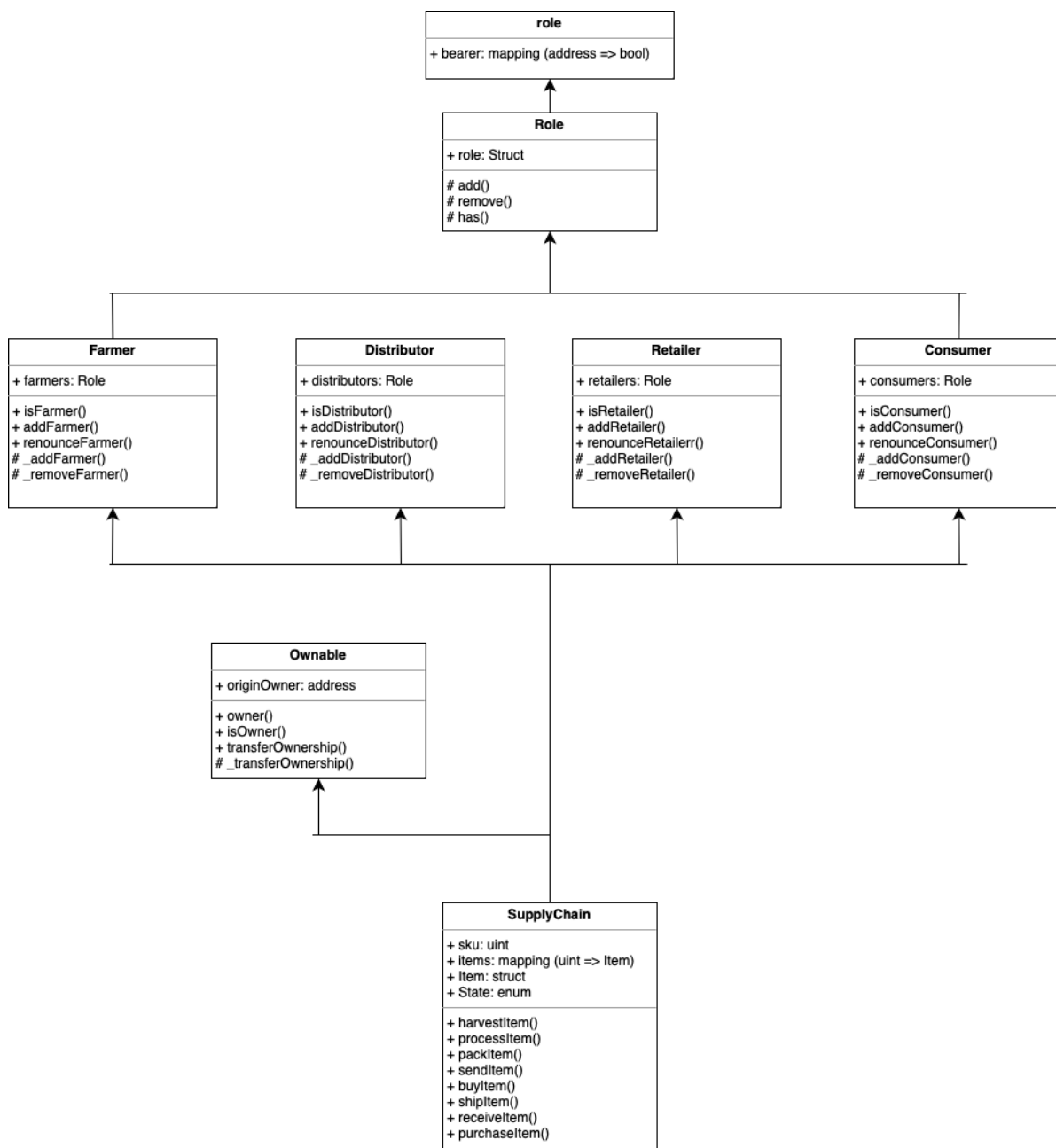+ processItem()
+ packItem()
+ sendItem()
+ buyItem()
+ shipItem()
+ receiveItem()
+ purchaseItem()

Class Diagram

Instructions

To use the Dapp's front-end, you need to take into account some considerations and follow a set of instructions in the terminal. These can be followed in the readme.md file.

The accounts that are generated are:

Contract Owner: 0xba1d5313f63916588fbe5b912801c07d3a7ec0b6
Farmer: 0x0a896038580ef8c2ec81ccedb531f9ad484be6ca
Distributor: 0xb705dd03a3981590b942fafffe318470f05597f9
Retailer: 0x3dfe9d12025f4c4b9271b872a8820de44d82e9c5
Consumer: 0x18d0ed0cb978ddd066c9e681de8aafcfab497648

When you open the website, the owner account will be asked to interact with the smart contract, to give all of these accounts their respective roles.
These will be the ones that will be hard-coded into the html file.
Whenever you switch accounts in Metamask, the select account will change after a few clicks in the website. Make sure you do this every time you switch accounts, otherwise you will be interacting with an account you didn't mean to. All of the functions have specific permissions, and if you don't use them with a permission account, your transaction will fail.

This would be a logical use of the front end:

Harvest item (Farmer Only)

Fill in the information about the product. The most important one is the UPC, as it will be the index on which a mapping to the item structure will be made.

Pack item (Farmer Only)

Change the status of the item from harvested to processed.

Pack item (Farmer Only)

Change the status of the item from processed to packed

Sell item (Farmer Only)

Define a price in the correct box and change the status of the item from packed to for sale.

Buy item (Distributor Only)

Buy the item, send ETH to the farmer's account, change the ownership of the item and change the status of the item from for sale to bought.

Ship item (Distributor Only)

Change the status of the item from bought to shipped.

Receive item (Retailer Only)

Change the status of the item from shipped to received.

Purchase item (Costumer Only)

Change the status of the item from Received to Purchased.

Fetch Data 1/ Fetch Data 2 (ALL)

Have a look at what information the UPC number has about the current supply chain.