Farmer	Distributor	Retailer	Consumer

Harvest

Process

Pack

Sell → Buy

Ship → Receive → Purchase

Coffee_UML

Farmer	Farmer	Farmer	Farmer	Distributor	Distributor	Retailer	Consumer

harvested	processed	packed	forSale	sold	shipped	received !purchased

harvestItem	processItem	packItem	sellItem	buyItem	shipItem	receiveItem	purchaseItem

Coffee State

Harvested	Processed	Packed	ForSale	Sold	Shipped	Received	Purchased

AccessControl　　　　　　　　　　　　　　　　　　　　　　　　　　**Base**　　　　　　　　　　　　　**Core**

<<Library>>
Roles

+ Role: struct

---
add(Role, address):
remove(Role, address):
has(Role, address):

SupplyChain

+ owner: address
+ upc: uint
+ sku: uint
+ items: mapping(uint=> item)
+ itemsHistory: mapping(uint=> string[])
+ State: enum
+ Item: struct

+ kill()
+ harvestItem(uint, address, string, string, string, string, string)
+ processItem(uint)
+ packItem(uint)
+ sellItem(uint, uint)
+ buyItem(uint)
+ shipItem(uint)
+ receiveItem(uint)
+ purchaseItem(uint)
+ fetchItem(uint): uint, uint, address, address, string, string, string, string
+ fetchItemBufferTwo(uint): uint, uint, uint, string, uint, uint, address, address, address

---

Ownable

- origOwner: address

+ owner(): address
+ isOwner(): bool
+ renounceOwnership()
+ transferOwnership(address)
# _transferOwnership(address)

FarmerRole

- farmers: Role.Role

+ isFarmer(address): bool
+ addFarmer(address)
+ renounceFarmer()
# _addFarmer(address)
# _removeFarmer(address)

DistributorRole

+ isDistributor(address): bool
+ addDistributor(address)
+ renounceDistributor()
# _addDistributor(address)
# _removeDistributor(address)

RetailerRole

+ isRetailer(address): bool
+ addRetailer(address)
+ renouceRetailer()
# _addRetailer(address)
# _removeRetailer(address)

ConsumerRole

+ isConsumer(address): bool
+ addConsumer(address)
+ renounceConsumer()
# _addConsumer(address)
# _removeConsumer(address)