Project Writeup

Background

"Fair Trade Coffee" is a company ensures that farmer get their rightful share. All the information related to supply chain is available on blockchain, so that consumer buying our product can contribute to fair trade and stop distributors from oppressing farmers due to information asymmetry.

In this project, we are going to make use of smart contract to keep track of the flow of product. Consumer can use our website to know the origin of their product, and how much they paid to the farmer for the product they buy.

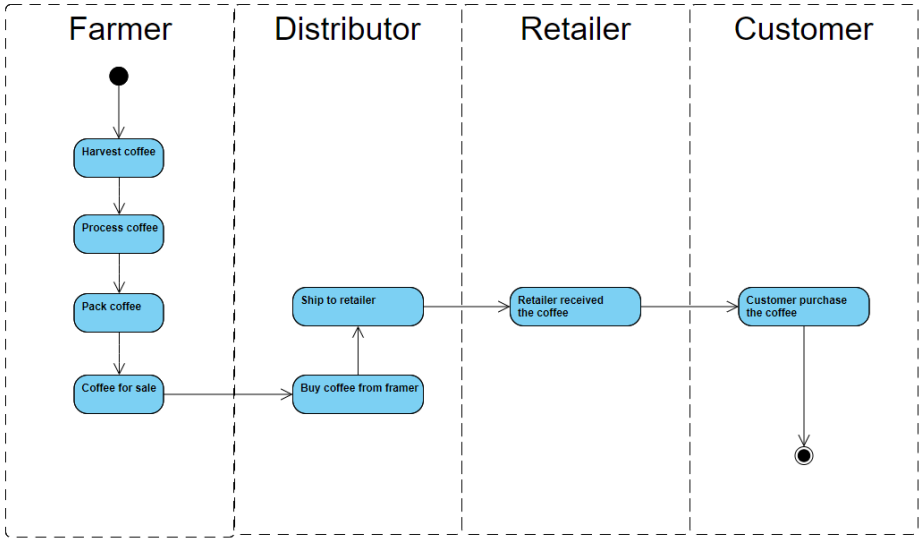
Definition

Fair Trade Coffee	A company promoting fair trade. Use smart contract to achieve the goal. As a smart contract owner to keep track of a supply chain visible to all.
Farmer	Farmer use our platform to sell their product to Distributor.
Distributor	Distributor is the one who is trading directly with our farmer. They buy stocks from our inventory and pay farmer using Ethereum.
Retailer	Retailers buy coffee from distributor. They have no direct business with our company. They contribute data to our supply chain because they want their customer know that they are buying coffee fairly.
Consumer	Customers buy coffee from retailer. They can use our website to track the history of coffee they brought.
Ethereum	Ethereum is a global, decentralized platform for money and new kinds of applications
Smart contract	Smart contracts are applications that run on the Ethereum

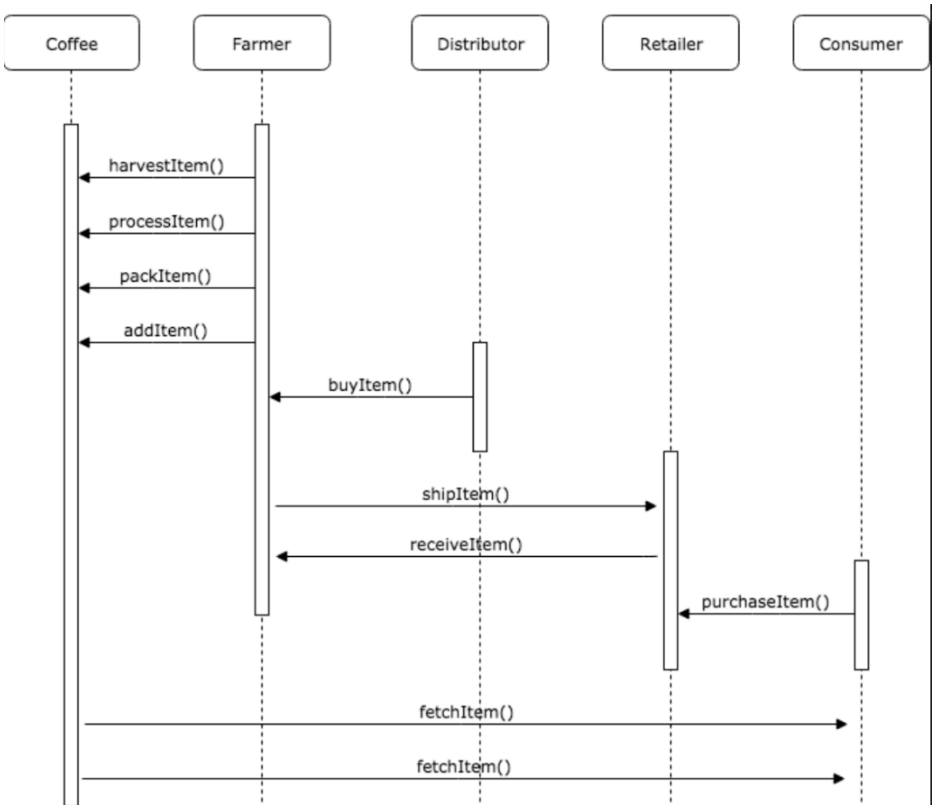
	Virtual Machine. This is a decentralized “world computer” where the computing power is provided by all those Ethereum nodes.
Supply chain	A supply chain is a network between a company and its suppliers to produce and distribute a specific product to the final buyer.

System design in UML

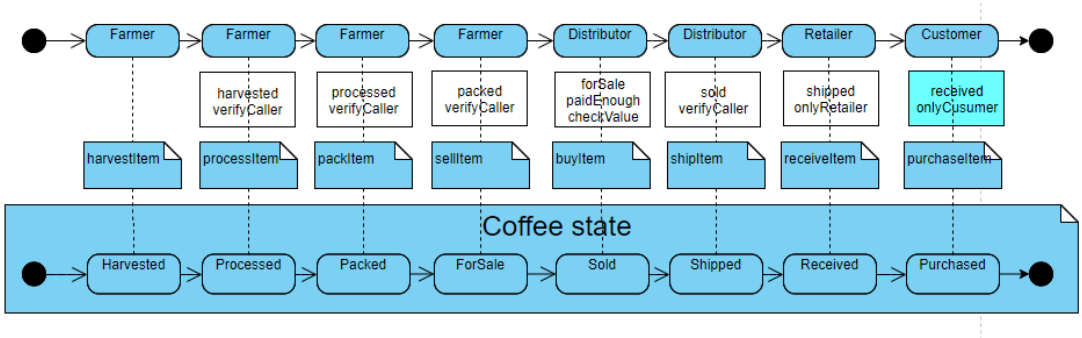
Activity



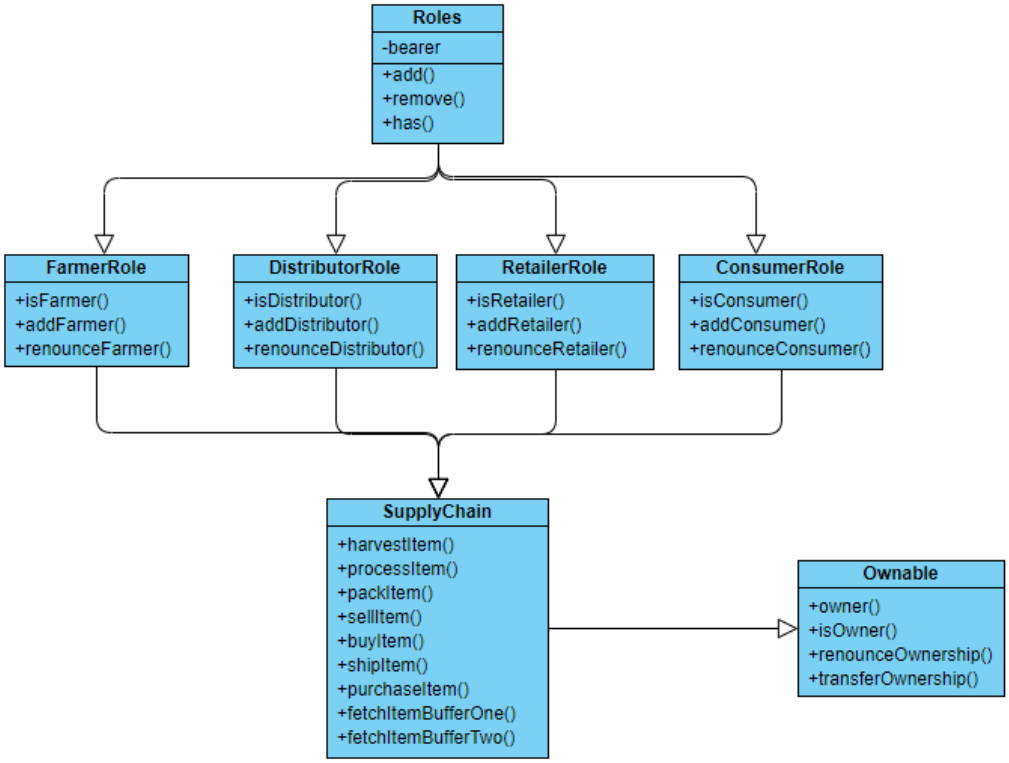
Sequence



State


Farmer
Farmer
Farmer
Farmer
Distributor
Distributor
Retailer
Customer
harvested
verifyCaller
processed
verifyCaller
packed
verifyCaller
forSale
paidEnough
checkValue
sold
verifyCaller
shipped
onlyRetailer
received
onlyCusumer
harvestItem
processItem
packItem
sellItem
buyItem
shipItem
receiveItem
purchaseItem
Coffee state
Harvested
Processed
Packed
ForSale
Sold
Shipped
Received
Purchased

Class


Roles
-bearer
+add()
+remove()
+has()
FarmerRole
+isFarmer()
+addFarmer()
+renounceFarmer()
DistributorRole
+isDistributor()
+addDistributor()
+renounceDistributor()
RetailerRole
+isRetailer()
+addRetailer()
+renounceRetailer()
ConsumerRole
+isConsumer()
+addConsumer()
+renounceConsumer()
SupplyChain
+harvestItem()
+processItem()
+packItem()
+sellItem()
+buyItem()
+shipItem()
+purchaseItem()
+fetchItemBufferOne()
+fetchItemBufferTwo()
Ownable
+owner()
+isOwner()
+renounceOwnership()
+transferOwnership()

Technical specification

Contract address:

The contract is deployed on Rinkeby test network.

Contract address: 0x93A7e33FE9cBc53e80C06018e9b3e138005d94dD

Development environment:

Truffle v5.1.21 (core: 5.1.21)
Solidity v0.5.16 (solc-js)
Node v9.4.0
Web3.js v1.2.1

To build the project

Change directory to `project-6` folder and install all requisite npm packages (as listed in `package.json`):

```
cd project-6
npm install
```

Launch Ganache:

```
ganache-cli -m "spirit supply whale amount human item harsh scare congress
discover talent hamster" -l 10000000
```

Your terminal should look something like this:

```
ubuntu@ubuntu-vbox:~/Desktop/Blockchain Nanodegree/Week4/nd1309-Project-6b-Example-Template-master$ ganache-cli -m "spirit supply whale amount huma
n item harsh scare congress discover talent hamster" -l 10000000
Ganache CLI v6.9.1 (ganache-core: 2.10.2)
eth_blockNumber

Available Accounts
==================
(0) 0x27D8D15CbC94527cAdf5eC14B69519aE23288B95 (100 ETH)
(1) 0x018C2daBef4904ECbd7118350A0c54DbeaE3549A (100 ETH)
(2) 0xCe5144391B4aB80668965F2Cc4f2CC102380Ef0A (100 ETH)
(3) 0x460c31107DD048e34971E57DA2F99f659Add4f02 (100 ETH)
(4) 0xD37b7B8C62BE2fdDe8dAa9816483AeBDBd356088 (100 ETH)
(5) 0x27f184bdc0E7A931b507ddD689D76Dba10514BCb (100 ETH)
(6) 0xFe0df793060c49Edca5AC9C104dD8e3375349978 (100 ETH)
(7) 0xBd58a85C96cc6727859d853086fE8560BC137632 (100 ETH)
(8) 0xe07b5Ee5f738B2F87f88B99Aac9c64ff1e0c7917 (100 ETH)
(9) 0xBd3Ff2E3adEd055244d66544c9c059Fa0851Da44 (100 ETH)

Private Keys
==================
(0) 0x9137dc4de37d28802ff9e5ee3fe982f1ca2e5faa52f54a00a6023f546b23e779
(1) 0x18911376efeff48444d1323178bc9f5319686b754845e53eb1b777e08949ee9b
(2) 0xf948c5bb8b54d25b2060b5b19967f50f07dc388d6a5dada56e5904561e19f08b
(3) 0xfad19151620a352ab90e5f9c9f4282e89e1fe32e070f2c618e7bc9f6d0d236fb
(4) 0x19d1242b0a3f09e1787d7868a4ec7613ac4e85746e95e447797ce36962c7f68b
(5) 0x3bb675f8c07099816e23a3e283090cfb0f793ab625b73ca51a2d027a3c1f2d0e
(6) 0x0faf45306c7daf14d86c266690ce54490e8c0104154cafa87d9e93724efc239d
(7) 0xf2a921dee0ebd7bfaba1a271bcd48c99baa6341a1cdf84ba843521a5555e0273
(8) 0x627345948 40dade92a24448c8f676cc3c59fd68909837303417295f2c0f27963
(9) 0xc29afb730456eb83415046550faf8065c8531765396156db8d97fd1fd64c6a6e

HD Wallet
==================
Mnemonic:      spirit supply whale amount human item harsh scare congress discover talent hamster
Base HD Path:  m/44'/60'/0'/0/{account_index}

Gas Price
==================
20000000000

Gas Limit
==================
10000000
```

In a separate terminal window, Compile smart contracts:

```
truffle compile
```

Your terminal should look something like this:

```
truffle(develop)> truffle compile

Compiling your contracts...
===========================
> Compiling ./contracts/Migrations.sol
> Compiling ./contracts/coffeeaccesscontrol/ConsumerRole.sol
> Compiling ./contracts/coffeeaccesscontrol/DistributorRole.sol
> Compiling ./contracts/coffeeaccesscontrol/FarmerRole.sol
> Compiling ./contracts/coffeeaccesscontrol/RetailerRole.sol
> Compiling ./contracts/coffeeaccesscontrol/Roles.sol
> Compiling ./contracts/coffeebase/SupplyChain.sol
> Compiling ./contracts/coffeecore/Ownable.sol
> Artifacts written to /home/ubuntu/Desktop/Blockchain Nanodegree/Week4/nd1309-Project-6b-Example-Template-master/project-6/build/contracts
> Compiled successfully using:
   - solc: 0.5.16+commit.9c3226ce.Emscripten.clang
```

This will create the smart contract artifacts in folder `build\contracts`.

Migrate smart contracts to the locally running blockchain, ganache-cli:

```
truffle migrate development
```

Your terminal should look something like this:

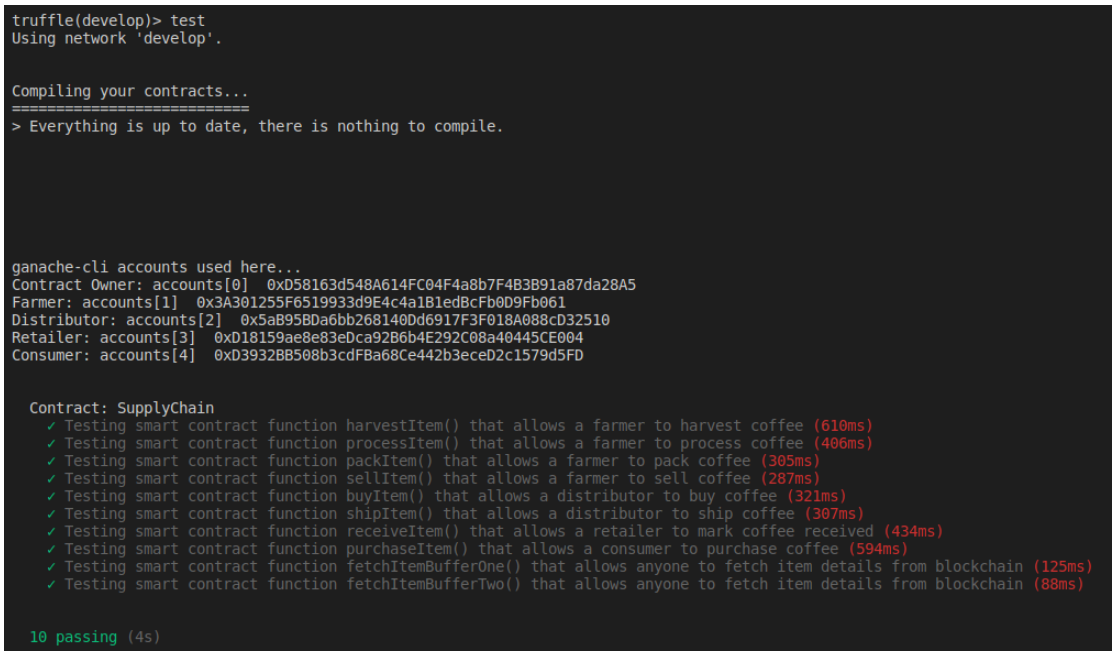
```
truffle(develop)> migrate --reset --network development

Compiling your contracts...
===========================
> Everything is up to date, there is nothing to compile.



Starting migrations...
======================
> Network name:    'development'
> Network id:      1586925017208
> Block gas limit: 10000000 (0x989680)


1_initial_migration.js
======================

   Deploying 'Migrations'
   ----------------------
   > transaction hash:    0x1ad487209e9f34c316454f97722afd8aad5e0f264ac247c38b0398781fb84126
   > Blocks: 0            Seconds: 0
   > contract address:    0xFEeCfF2CB7d6f3BfcBE5fa41c49c8fB642f2dDbF
   > block number:        1
   > block timestamp:     1586925503
   > account:             0x27D8D15CbC94527cAdf5eC14B69519aE23288B95
   > balance:             99.99549526
   > gas used:            225237 (0x36fd5)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.00450474 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00450474 ETH
```

```
2_deploy_contracts.js
=====================

   Deploying 'SupplyChain'
   -----------------------
   > transaction hash:    0xdc6caa5f3c29e83d69903109347cd6d7ce02436f5199d571682d105b52c28140
   > Blocks: 0            Seconds: 0
   > contract address:    0xf2ee0b0Cdcae5013930B92c0Ba54F7F7f1933613
   > block number:        3
   > block timestamp:     1586925504
   > account:             0x27D8D15CbC94527cAdf5eC14B69519aE23288B95
   > balance:             99.93182414
   > gas used:            3141193 (0x2fee49)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.06282386 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.06282386 ETH


Summary
=======
> Total deployments:   2
> Final cost:          0.0673286 ETH
```

Test smart contracts:

```
truffle test
```

All 10 tests pass.



In a separate terminal window, launch the DApp:

```
npm run dev
```

Migrate smart contracts to the Rinkeby test network:

```
truffle(develop)> migrate --reset --network rinkeby
```

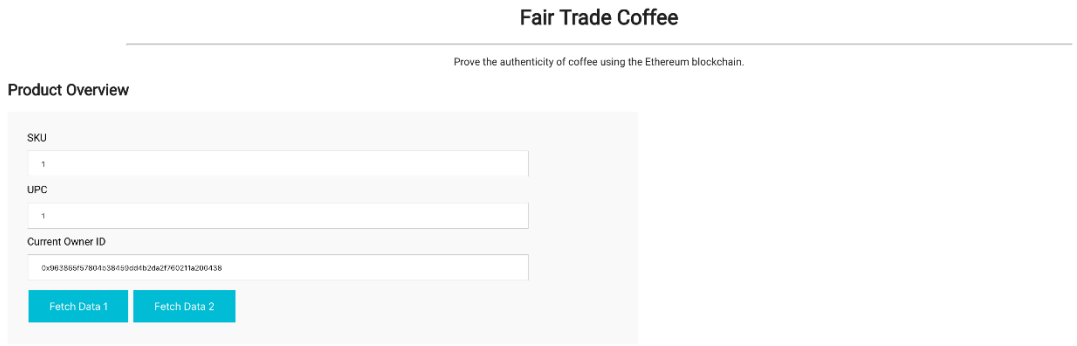
```
Starting migrations...
======================
> Network name:    'rinkeby'
> Network id:      4
> Block gas limit: 10000000 (0x989680)


1_initial_migration.js
======================

   Replacing 'Migrations'
   ----------------------
```

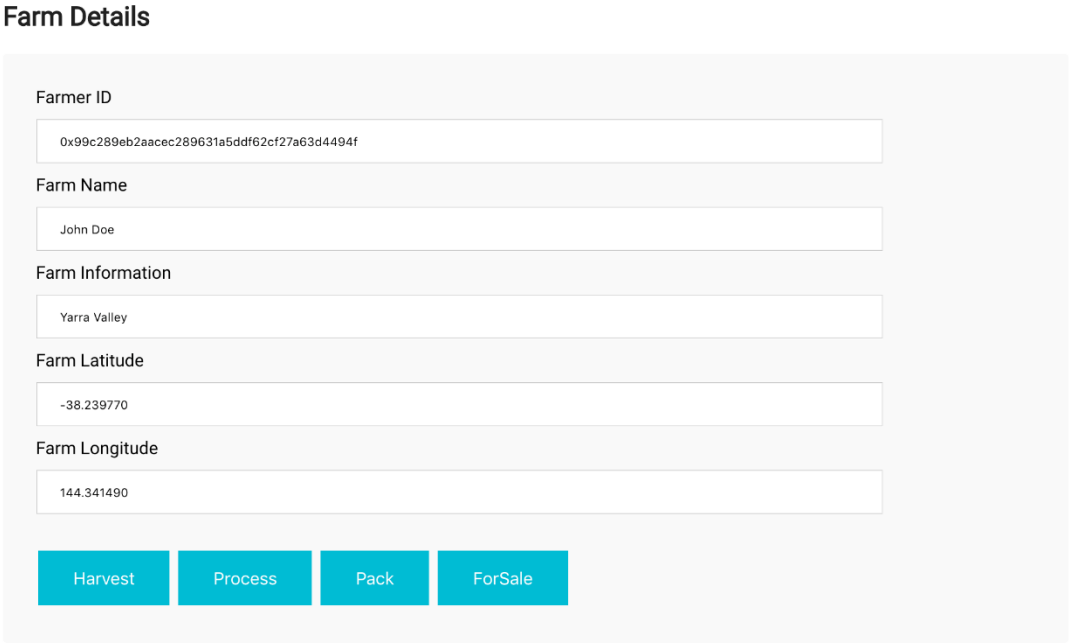
```
   > transaction hash:
0xa387bf793a9166dd3e82d9e8f522662ed48e38895804af1104ecd5967174d661
   > Blocks: 4            Seconds: 71
   > contract address:    0x84C40700C267F96bfCB0e4bea71915Ec6F31A4e7
   > block number:        6389378
   > block timestamp:     1587996098
   > account:             0x7734bF52F5F4C2278d3bA2B6f0C2Fa76d2356273
   > balance:             7.63110803
   > gas used:            225237 (0x36fd5)
   > gas price:           10 gwei
   > value sent:          0 ETH
   > total cost:          0.00225237 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.00225237 ETH


2_deploy_contracts.js
=====================

   Replacing 'SupplyChain'
   -----------------------
   > transaction hash:
0xbf34892645696a29d38fa7c61ebcbff6ef7d7637612e875b71046e34dd9fadcf
   > Blocks: 2            Seconds: 17
   > contract address:    0x93A7e33FE9cBc53e80C06018e9b3e138005d94dD
   > block number:        6389382
   > block timestamp:     1587996158
   > account:             0x7734bF52F5F4C2278d3bA2B6f0C2Fa76d2356273
   > balance:             7.59859551
   > gas used:            3208889 (0x30f6b9)
   > gas price:           10 gwei
   > value sent:          0 ETH
   > total cost:          0.03208889 ETH
```

```
   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.03208889 ETH


Summary
=======
> Total deployments:   2
> Final cost:          0.03434126 ETH
```

Front end

View:





Product Details

Product Notes

Best beans for Espresso

Product Price

1 ETH

Distributor ID

0xf1b1bcd24dad92303dd9fe78be639f7bcf9c238d

Retailer ID

0x0fb2bfefd526966c87efa19f6693d50971763fc2

Distributor ID

0x5d777e9127b28fb119e81a6304278a0a21bef1c4

Buy | Ship | Receive | Purchase

Transaction History

- Harvested - 0x1120fc1764e39832d6ad4a905bb341201e860e64716e2c212544098b6f39e044
- Processed - 0x0bf9f14716e7215ca31c4a4337e618627ffac43a760d98f705c2df254093ae12
- Packed - 0x1758b0661098f9a7829ffe323366d80bfa89659c6506e5fa1175190ec744f67a
- ForSale - 0x5b37b7d07c1ac4526f5f8dd28e301c379bdf21ebd59257286e3df742b60dd716
- Sold - 0x6eb2aa9e38f346f579bb2aa249a0525f25841d6bd248d9a95a44f02380d6da03
- Shipped - 0x22ed68b2b2bd19bed39460458a3a42d04457cb4f6a78cad181af1f36a6dea71d
- Received - 0xbae9ca8ac14b1ffc65f3c4e27b6bc67e875e81143a5b3cbaa1497b3f5288e34f
- Purchased - 0xc0d1cd924b6e072ea6f4e537f20847e7ed9727e3bed7431c625b3b5d16e7fbb8