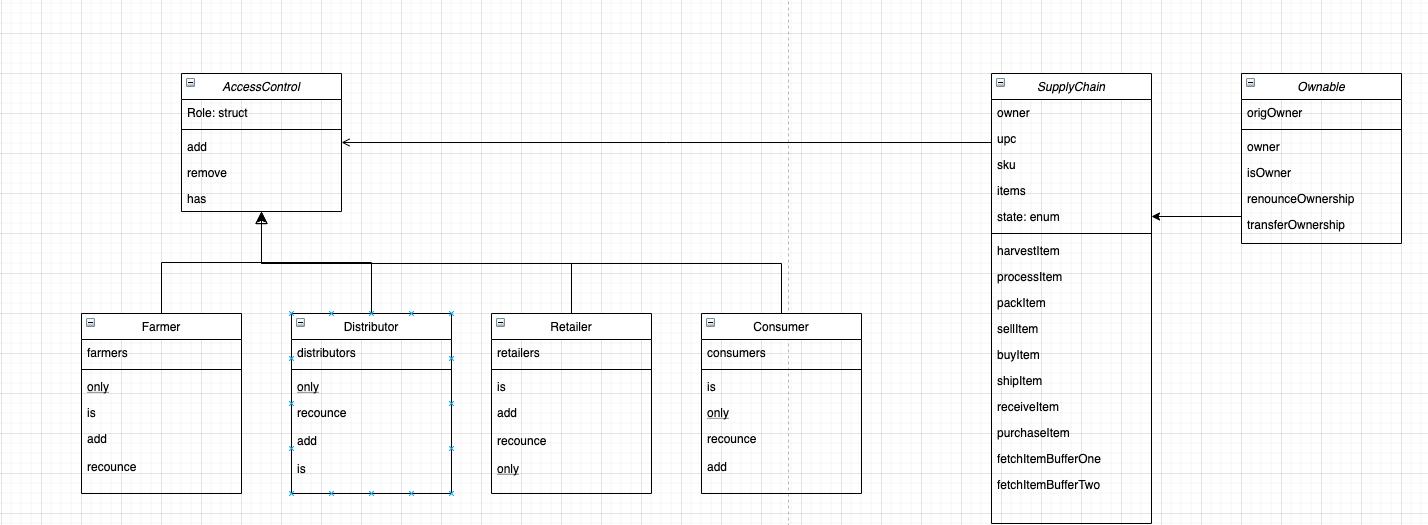
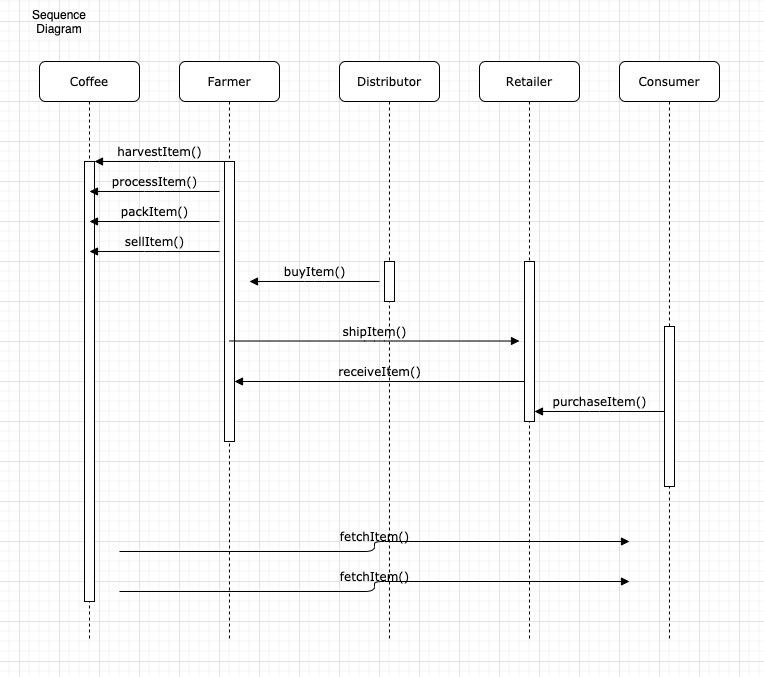
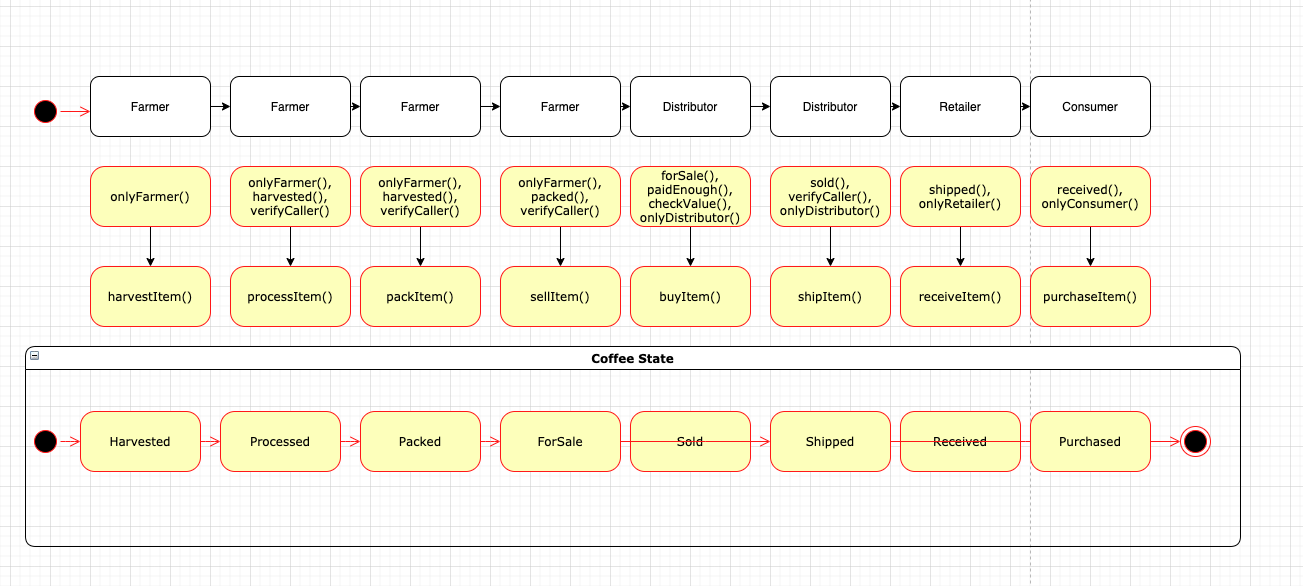
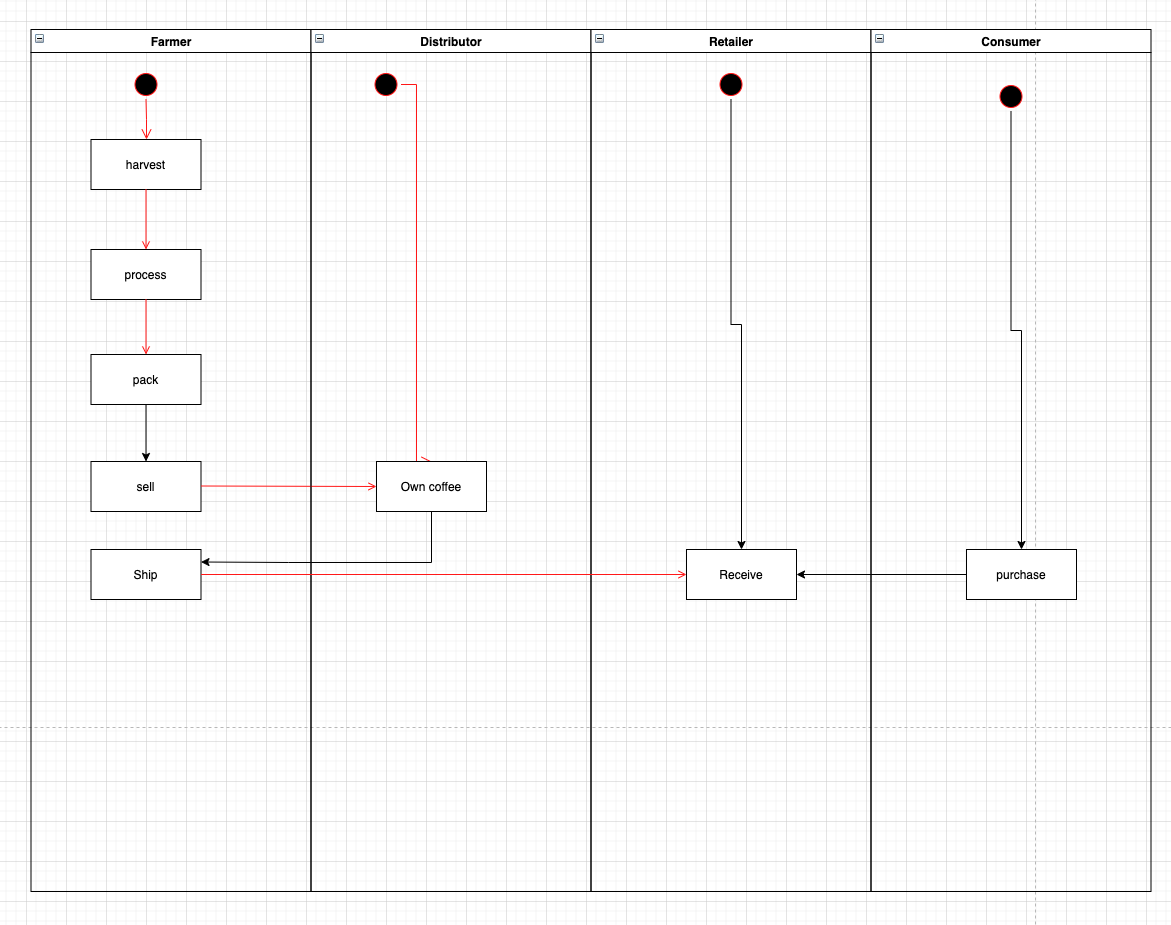
**Diagram:** XML source can be found in the root folder

****



**Libraries:**

Ethereum: as Ethereum is publicly accessible ledger, this will help us build a transparent Dapp allowing each actor of the supply chain to track the coffee beans.

Truffle: To ease the development process and have and environment set up fast, we will use Truffle.

For testing our contracts, we will use Mocha assertion libraries. This will help us build confidence around our contract’s logic

**IPFS:**

We are building a Dapp, as a consequence, no element of the architecture should be centralized. This is why we have chosen to use IPFS for data storage instead of a centralized one (like s3)

**Versions:**

Truffle v5.1.46 (core: 5.1.46)

Solidity v0.5.16 (solc-js)

Node v12.18.3

Web3.js v1.2.1

**Contracts:**

\* FarmerRole: 0x1aB06732FA46e2a139f206A1629F9878c263531c  
\* DistributorRole: 0xC8D314013437081f4c375db7614F6e5D4E34014C  
\* RetailerRole: 0x5055E53A90c06d5E92d14ADc3951a3479f456Fae  
\* ConsumerRole: 0x8C15861ab152828Cbdb4696a42EAB8d8D3f22A29  
\* SupplyChain: 0xB2A725f13Db9CB16257586E4e4bd67455116bD20

**Transaction ids:**

harvestItem: 0x66ce18a1e712c70171a44ce907578e2fd65b4443091d8f3e1611982c35a3e303

processItem: 0x86072c14523fb691cc72ff1e11a120a162e01268c15a5027764d100540496cf6

packItem: 0xdb8a5cbaa06874a0661f8954c310f07cc4d663d1415d63e5b8bf90b83eea6a33

sellItem: 0xbc77784665fea49272568f556cc79bd212886a4d677f887d8d5499464535cbfa

buyItem: 0x2fdb822aeded0ac7d15fb5fbb38be9d4ddc73200d1d54e2cb7f7d842fa6e0b11

shipItem: 0xe814f5f6a076961a857466327cde13f90762467c0b22bc8d16561ad9bcd01064

receiveItem: 0xc112faeece8e3c8554c0af56b0d62430dc1d8410f029ace84295fe37d71f14c3

purchaseItem: 0x92afa17fffdc37df59d82bc75e63651ac5837caf66397655093212dfad456128