**Music Supply Chain Project Plan**

**Intro**

This project is about a traditional Music Supply Chain, such an artist (singer, compositor, band, etc.) work alongside a record company to create a song. In this kind of supply chain we have an artist, the first actor composing the music, what makes him or them the owner of the item.

Some record company provides services like creation of the mix (with mixing engineers and producer, along other technicians), also the same with creation of a master (the digital product). The record company is responsible for paying the artists in full and or some other royalties sort of contracts in the industry – the record company here owns the item (music, album, etc.).

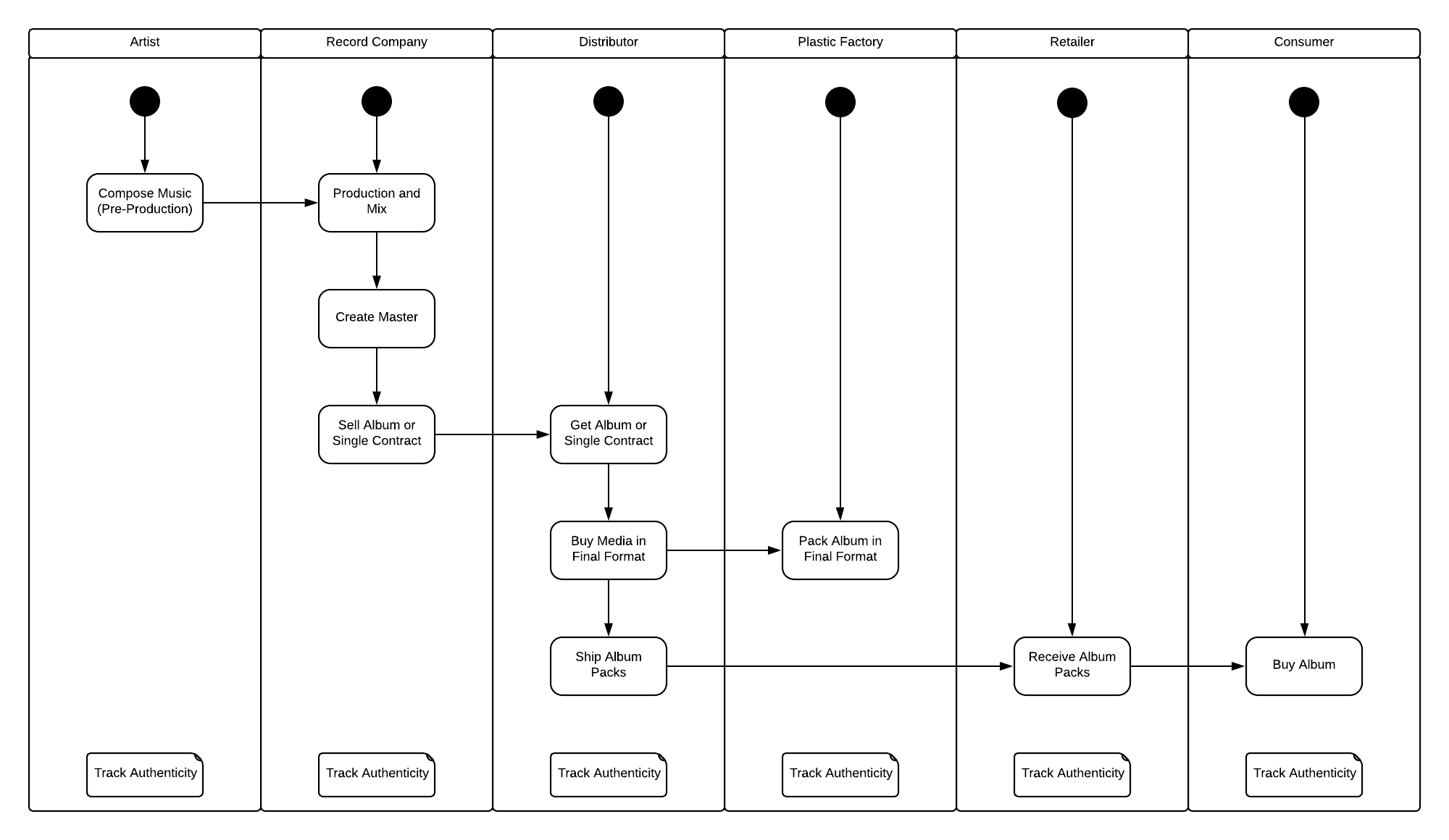
Having the master complete, the record company signs a contract to a distributor (like Universal, or Sony Music), selling to them the rights on the item (depending on the contract they can get some participation on the royalties as well). The owner is the distributor.

The distributor is responsible for creating the disk plastic cases (usually buying that from some plastic industry or other company to serve as third party here). After that, it we have some complete product ready to be sold out in the retailing market. The distributor is the owner until sells the item to the retailer (retailer owns the material – not the legal rights on the song – off course).

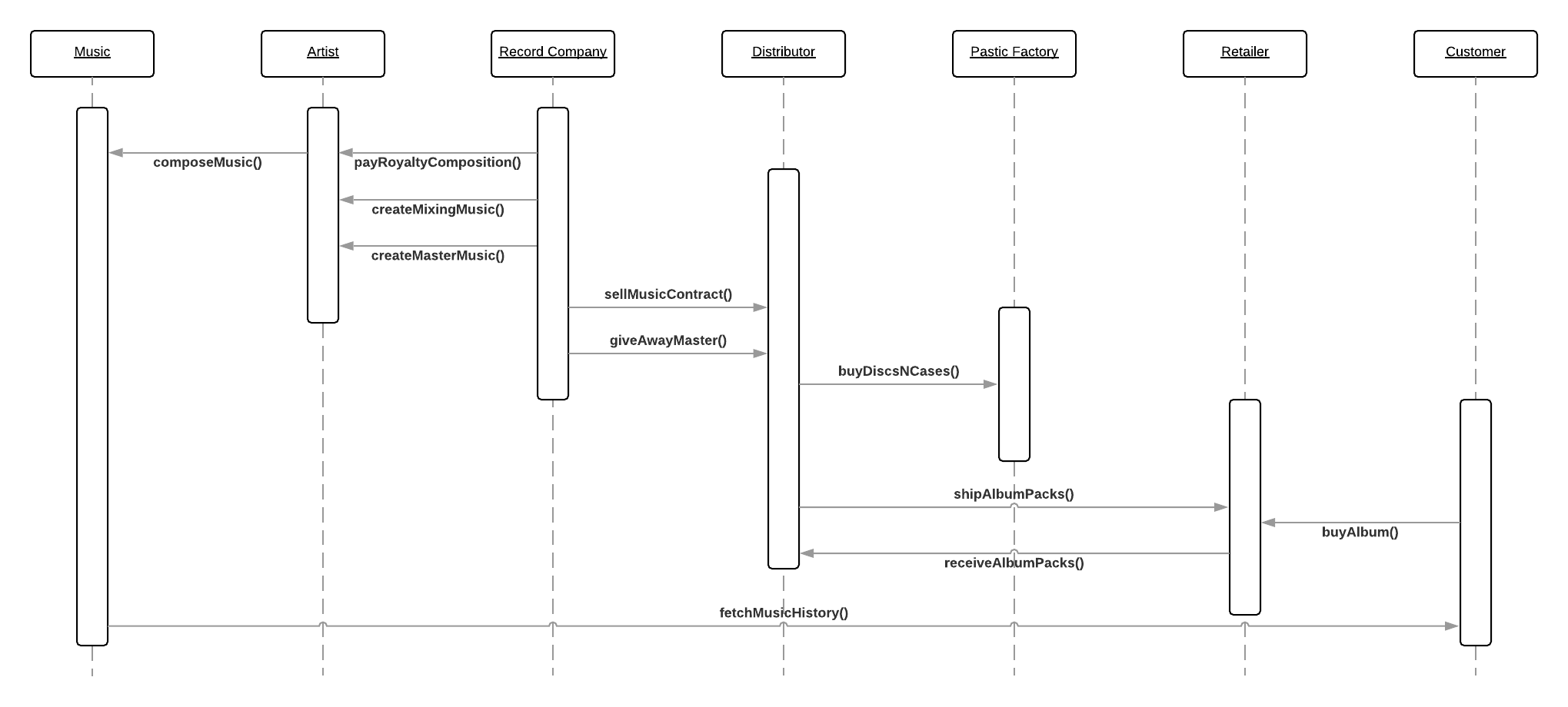
The retailer them can sell it to a consumer – the final packed product, that now is the owner of the material item.

**1 – UML (From Project 6 Part A)**

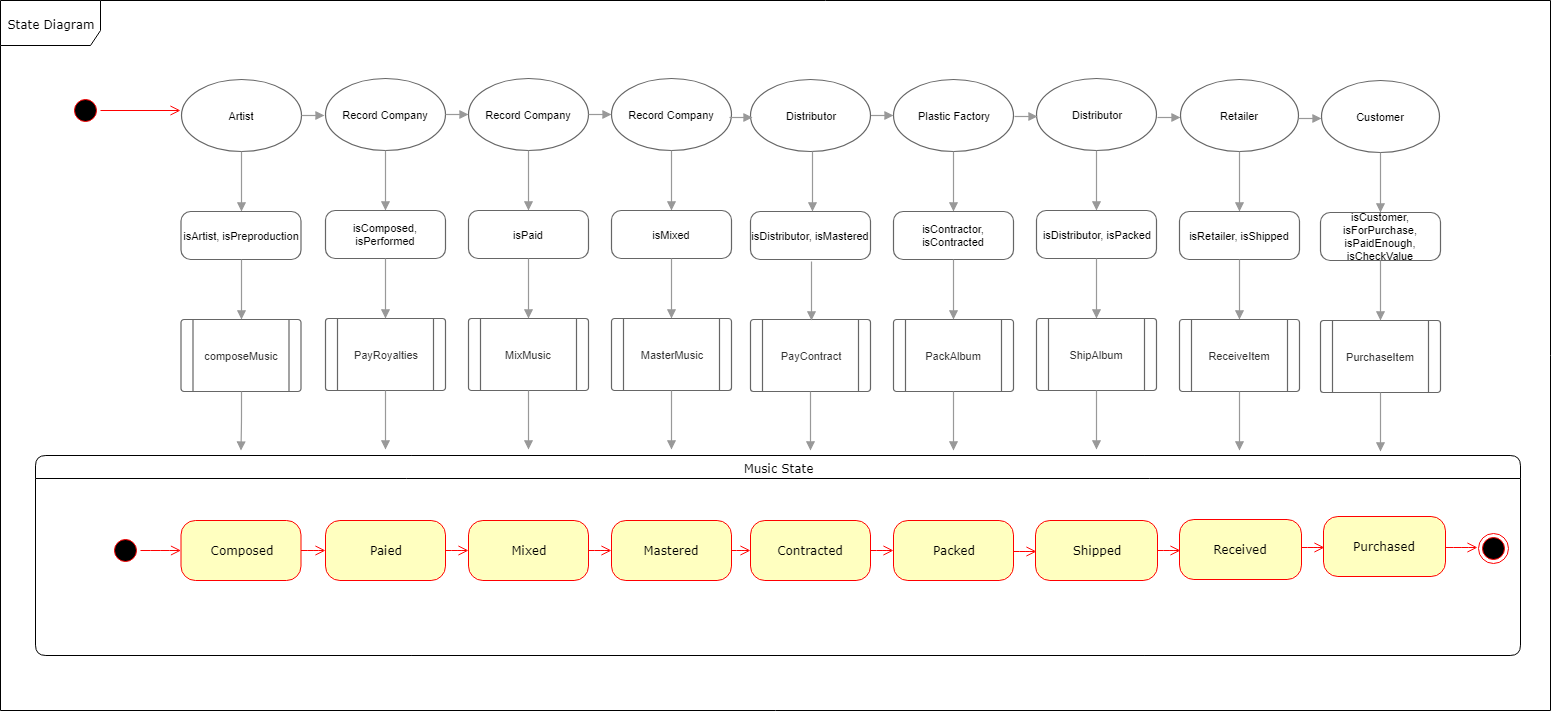
1. **Activity Diagram**



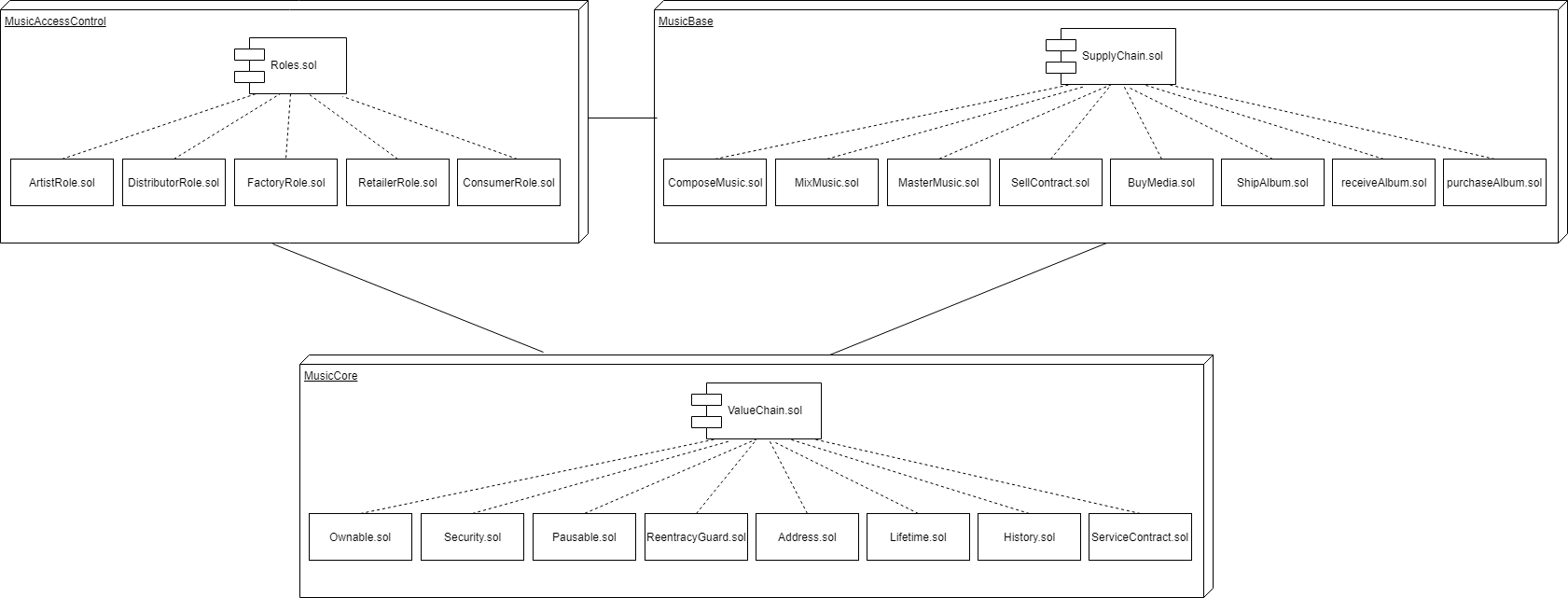
**2 – Sequence Diagram**



**3 – State Diagram**



**4 – Class Diagram**



**2 – Libraries**

**Node** Version: 8.11.3

**Truffle** Version: 4.1.14

**Web3**: Version: 1.0.0-beta.48

**NodeJS truffle-hdwallet-provider: Version: 1.0.5**

Used to handle smart contracts deployed to Etherium networks

OBS: Depending on the version of Web3, the function call in TestSupplychain.js line 18 may result in error:

web3.utils.toWei(1, "ether");

Needing a replacement for:

web3.toWei(1, "ether");

**3 – Submitted Contract Address (SupplyChain)**

[https://rinkeby.etherscan.io/address/0xef2ace9c9bcc0bc7d71e7e7773bb8bc118dba3b0#](https://rinkeby.etherscan.io/address/0xef2ace9c9bcc0bc7d71e7e7773bb8bc118dba3b0)

**Contract ID:**

0xef2ace9c9bcc0bc7d71e7e7773bb8bc118dba3b0#

**Transaction ID:**

0x1c718de4f4dc7739cbd8a3afa499e16510539580f21f034459c7405e1182b635