

Smart Contracts for Supply Chain Management

Miulescu Cristina-Maria, SCPD

Abstract

Supply chains are becoming more globalized, making management and control more difficult. As a distributed digital ledger platform that assures transparency, traceability, and security, blockchain technology is showing promise in addressing some global supply chain management issues. The potential application of blockchain technology and smart contracts to supply chain management is critically addressed in this article. Government, community, and consumer pressures on local and global governments, communities, and consumers to meet sustainability targets have prompted us to look at how blockchain can address and improve supply chain sustainability. Part of this critical analysis is to see how blockchains, a potentially disruptive technology yet in its early stages of development, can overcome a variety of obstacles.

1 Introduction

Blockchain, the digital ledger technology that underpins Bitcoin and other cryptocurrency networks, has the potential to revolutionize the financial industry. However, supply chain management is another area where it has a lot of potential. By enabling faster and more cost-effective product delivery, improving product traceability, boosting partner coordination, and facilitating access to funding, blockchain can significantly improve supply chains. A distributed, or decentralized, ledger is a computer system for documenting transactions between several parties in a verifiable, tamperproof manner. The ledger can also be designed to automatically initiate transactions. The basic role of blockchain in cryptocurrency networks aimed to replace fiat currencies is to allow an endless number of anonymous parties to deal discreetly and securely with one another without the need for a central middleman. It is to allow a small number of well-known parties to safeguard their company operations against hostile actors while also promoting greater performance in supply chains.

2 Project Architecture

In supply chain, smart contracts are useful in three areas:

- **Making payments** - a smart contract can decide whether a payment should take place based on pre-defined criteria.
- **Recording ledger entries** - it records changes to the shipping container (delivery and violation status).
- **Alert for manual intervention** - when violations happen to the shipment these will be recorded

3 Scenario

Supplier A manufactures the product for supplier B. Once created, it is shipped and its details are registered to the smart contract. If any violation happens (example: temperature increases), the state from the contract will be changed.

When supplier B receives the package, an evaluation should take place before the final payment. Also, if the criteria is not met, there may be negotiations between the suppliers.

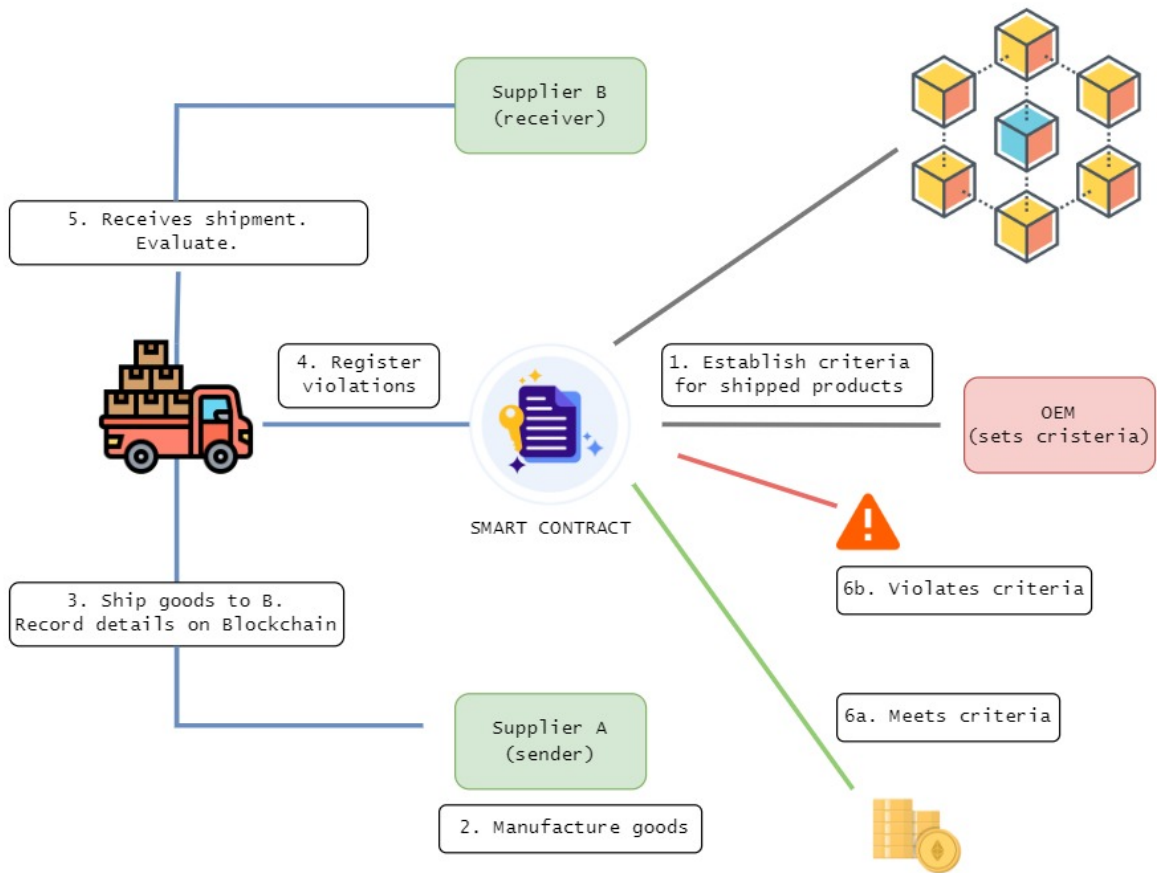


Figure 1:

4 Smart contract creation

The smart contract is written in Solidity and deployed to a test network such as Goerli. The compilation and deployment is possible with the help of Hardhat (ethereum development environment) and a node provider such as **Alchemy/Infura**.

There are two contracts:

- **SupplyChainManagement Contract**
- **Ownable Contract**

4.1 SupplyChainManagement contract

This contract has two roles: to keep track of the products and also to trigger the events and change a container's delivery or violation status. As we can see from the figure below a product can change its state during the delivery process.

The contract's owner will establish a new container with a name, price, quantity and customer address and then the delivery will be triggered. The customer will initiate the product's arrival once the delivery is triggered. The product should then be evaluated, and based on the violation status the container will be considered compliant or non-compliant.

In order to be fair to both parties, after the evaluation stage, if the container is non-compliant the negotiation will be triggered and further discussion should take place.

4.2 Ownable contract

This last contract decides the allowed actions for each one of the participants. For example, the owner of the contract is not allowed to pay for its product. In the same way, the customer cannot create a new product and set its price.

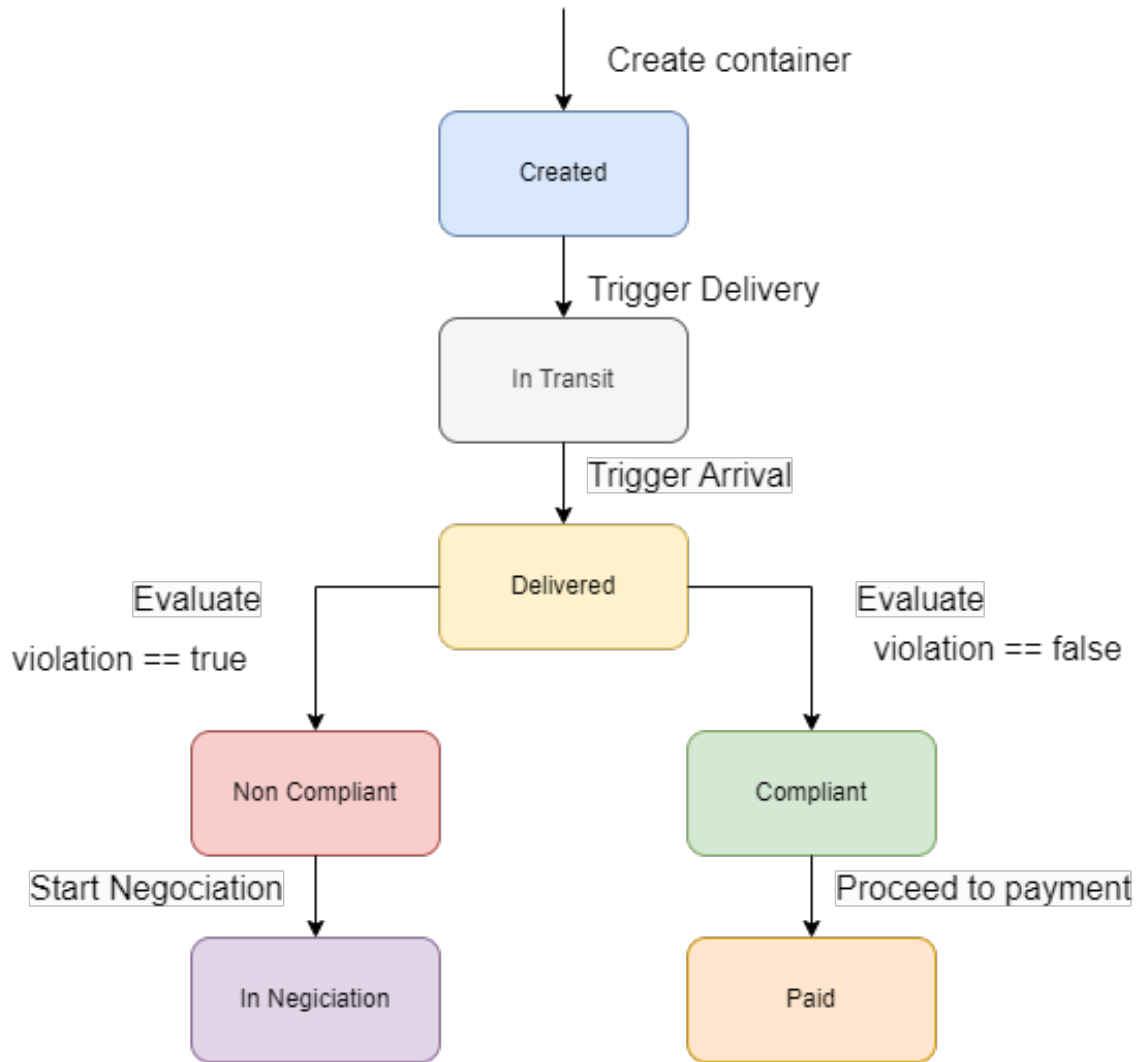


Figure 2: Simulator view

5 Smart Contract Deployment

For the smart contract deployment to the Goerli test network, I used Alchemy. Alchemy is a blockchain scaling platform that allows developers to securely create, test, and monitor their decentralized apps (DApps). The platform provides dependable network connectivity and node management endpoints.

6 Supply Chain Simulator

In order to show the stages a container would take since the creation and until it arrives to the final customer, I build a simulator. This would be the container itself. Based on the sensor data, it can establish if any violations may take place. As showed in the figure below, the container will trigger the delivery and mark any violation while they are in transit.

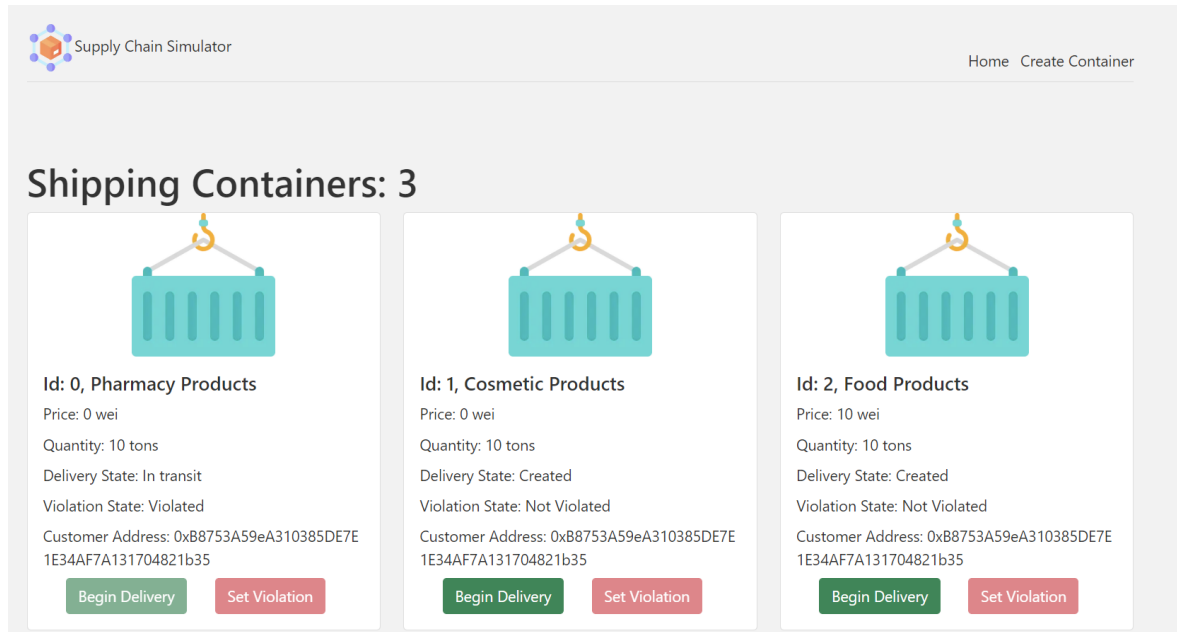


Figure 3:

7 Supply Chain Client

The client platform, will require the customers to use their Metamask accounts for login. Once they are connected, they will be able to see all their containers. The action they can take are the following:

- **Confirm Arrival**
- **Begin Evaluation** - this will establish if the container respects the terms
- **Begin Negotiation** - negotiations will take place if the container is non-compliant
- **Proceed to payment** - if the container respected the terms of delivery then the customer may proceed to payment

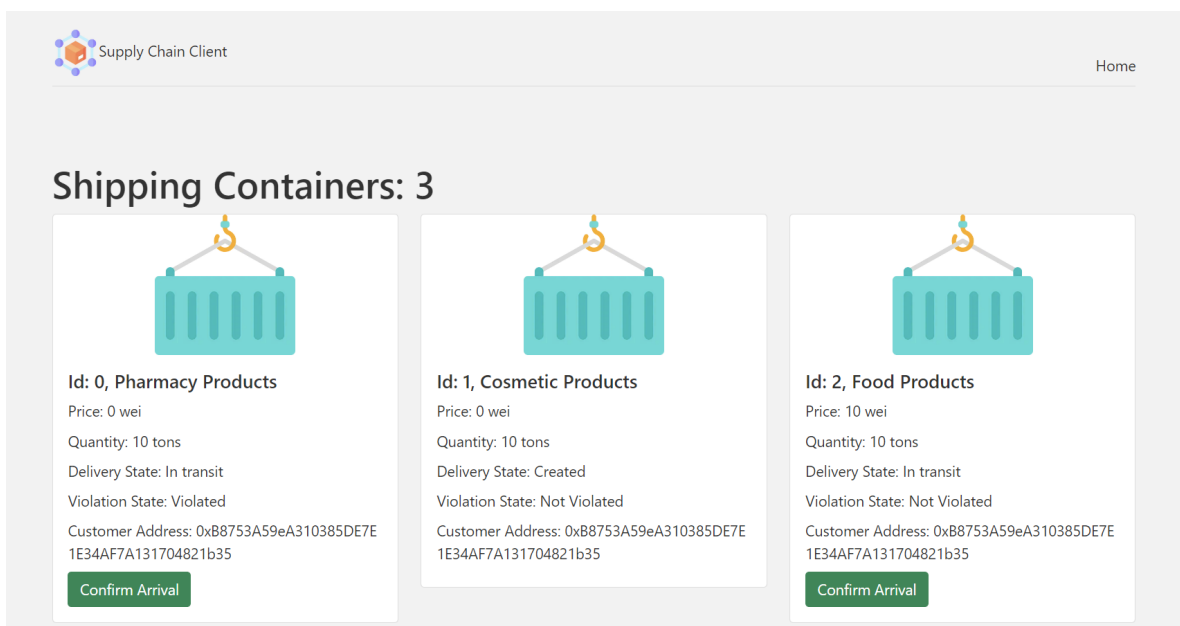


Figure 4: Client platform