

ETHEREUM TRACKING CHAIN

Overview

Cada vez se hace más necesaria la trazabilidad para saber la historia de un determinado producto que consumimos.

Normalmente en el proceso desde la creación de un producto y la venta al cliente final intervienen varias organizaciones que no necesariamente están conectadas.

Una fruta que se cosecha en una zona determinada cuando se recoge se etiquetan las cajas. Estas cajas se transportan, se almacenan, se distribuyen a minoristas y al final llegan a un consumidor final.

¿Qué le gustaría al consumidor? Cuando consumo una fruta me gustaría saber cuándo se recolectó, cuando entró en almacén, cuando se distribuyó a minorista. En general nos gustaría saber la historia del producto.

Esto es muy complejo, ya que el almacén puede recibir varias entregas de producto de diferentes fechas y luego enviar al minorista una remesa que contiene diferentes fechas de entrada.

Pero, no intentando resolver todo el problema, vamos a realizar un proyecto donde solucionemos los problemas, haciendo algunas suposiciones.

Especificaciones

Tenemos los siguientes actores en el sistema

1. Productor
2. Almacén
3. Mayorista
4. Minorista

El consumidor ha de saber las fechas de paso del producto por cada organización. Se podría complementar con una ficha de producto.

Vamos a suponer que las cajas van etiquetadas con un código único. El productor debe disponer de una aplicación que permita meter la relación de códigos.

Estos códigos irán al Almacén y este registrará en el sistema la entrada de las cajas,

El almacén enviará al mayorista cajas cada una con su código.

El mayorista hará lo mismo registrará las cajas

BLOCKCHAIN TECHNOLOGY ETHEREUM PROJECTS

El mayorista se comunicará con el minorista de la misma manera. Al final el consumidor podrá ver por el código la trazabilidad del producto.

Consideraciones

Hemos considerado que la cadena de distribución es Productor, Almacén, Mayorista, Minorista. A veces no es así

Vamos a suponer que lo que se transporta tiene un código único y que permanece así hasta el minorista.

Es posible considerar otras cadenas de suministro. En principio lo que hace el sistema es anotar un evento por cada cambio de situación del producto.

Implementación

Deberíamos de tener un registro de organizaciones. Estas organizaciones tendrían un address y un nombre. No sé si sería necesario.

La etiqueta que pone el productor servirá para obtener los eventos.

Esta etiqueta puede ser un aleatorio obtenido por una librería como `uuid()`.

En el contrato inteligente tendríamos un mapa indexado por etiqueta que contenga un array de eventos.

Cada evento tendrá una fecha, un address de la organización y la firma de la organización. Los datos a firmar son la fecha.

Si pensamos en eventos ligados a la geolocalización podríamos pensar en guardar coordenadas geográficas (lon, lat). Esto se hace con el tracking de productos en logística.

Cada evento nuevo se añade al anterior.

Para conocer los pasos que ha seguido el producto se accederá por etiqueta y se obtendrá el array de eventos. Como tenemos las firmas podemos validarlas.

BLOCKCHAIN TECHNOLOGY ETHEREUM PROJECTS

Funcionalidades

1. Programación Solidity
 - a. Insertar un evento a una etiqueta. Esta puede existir o no.
 - b. Insertar un evento a un array de etiquetas. Proceso batch para cuando una organización recibe muchos productos etiquetados.
 - c. Devolver todos los eventos de una etiqueta.

2. Dapp
 - a. Insertar una etiqueta. Cada evento debe de ir firmado.
 - b. Upload de un fichero con etiquetas.
 - c. Ver los eventos de una etiqueta. Debe de sacar todos los eventos disponibles.

Tecnología

1. Para construir la Dapp usaremos React con Vite o NextJS
2. Para realizar el Smart Contract usaremos Remix o Truffle.

