Gestor de identidades sobre una blockchain

VERIFICADOR DE ID

Alejandro Nardo González c412
Carlos Aguilera Medina c411|
Sistemas distribuidos | 22.12.2017

Introducción

En nuestro proyecto decidimos hacer un gestor de identidades sobre una blockchain, o sea un sitio donde cualquier persona pueda registrarse y el sistema esté completamente distribuido en la red.

¿Por qué en una blockchain? La blockchain es una estructura de datos inmutable, o sea no puede ser cambiada, si alguna persona o software intenta cambiar dicha blockchain, por sus características esta quedaría invalidada por completo por si sola y no podría funcionar más dentro de una red. Con esta característica, será imposible modificar el estado de un sistema a no ser que sea para agregar algo nuevo. Las identidades quedarían inmutables y no habría manera de modificarlas.

Esta poderosa herramienta se combina con criptografía y se pueden hacer cosas como firmas digitales, con facilidad de identificar si son verdaderas ya que no pueden ser cambiadas.

ARQUITECTURA

En nuestro sistema la topología de red no está definida. Estamos en presencia una red peer-to-peer y dichas redes son muy inestables, o sea nodos van y vienen todo el tiempo. Cada nodo podría estar conectado con todos, pero en la práctica no es necesario tener un puñado de nodos vecinos por cada nodo. El sistema esta implementado en Python.

Cada nodo puede tener varias funcionalidades. Todos los nodos presentan la funcionalidad de la red, o sea todos los nodos participantes en la red están todo el tiempo escuchando nuevos mensajes, esto puede ser bloques, identidades o nodos introductorios. Cada nodo puede tener la funcionalidad de wallet o card en nuestro caso, es decir una función que gestiona tu identidad, la crea y la esparce por la red. Al igual que bitcoin cada nodo puede tener o no la full blockchain, aunque nuestra blockchain pesaría menos que la de bitcoin. A diferencia de bitcoin, en nuestro sistema no existe el concepto de minado, usamos otro algoritmo de consenso, proof-of-Authorization. Como en nuestro sistema no hay monedas ni ganancias de por medio, es menos sensato usar un algoritmo de consenso que use tantos recursos como proof-of-work. Entonces usamos otro dicho algoritmo, que se basa en proof-of-stake, lo que los nodos confiables que pueden crear nuevos bloques ya están predefinidos.

Estas son las funcionalidades que puede tener cada nodo en la red.

CARACTERISTICAS DEL SISTEMA

Este proyecto es algo peculiar, no usa los típicos algoritmos de sistemas distribuidos dados en clase. En nuestro caso los datos que son compartidos son la blockchain y está repartida en toda la red. O sea, cada nodo debería tener el mismo estado en la red. Esto es uno de los requisitos que tiene que cumplir cada nodo para poder participar en la red y no crear inconsistencias. Es decir, la replicación en este sistema es bastante sencilla de entender porque todos los nodos deben tener lo mismo.

Pero explicaremos que ocurre desde que un nodo entra en la red hasta que se cae y vuelve a entrar, no entrando en detalles porque no quedaría nada para la exposición.

Antes de comenzar, para que un nodo pueda entrar en una red peer2peer, usualmente tiene que conocer al menos la dirección de un nodo en la red, entonces al igual que bitcoin, nuestra red debe tener nodos corriendo todo el tiempo, dichos nodos se le llaman nodos semillas que tienen la característica de darle introducción a un nodo nuevo en la red. Aunque nuestro sistema cualquier nodo puede hacer esto, debe de haber al menos uno semilla que introduzca nodos nuevos en la red. Una vez que un nodo conoce al menos otro nodo en la red, se establece una conexión TCP, todas las conexiones en nuestra red son de este tipo, se envían mensajes de versión de cada nodo, y el nodo semilla introduce al nuevo nodo enviándole la dirección de todos los nodos conocidos por este. Después, una vez que el nodo nuevo conoce un puñado de direcciones, hace una introducción con todos estos nodos haciéndoles saber que está participando en la red.

Después de la introducción, viene la sincronización del estado de la red para el nuevo nodo. El nodo nuevo cuando intercambia el mensaje de versión con otro nodo, se intercambia el tamaño de la blockchain de cada uno, y el nuevo nodo sabrá cuantos bloques le faltan y pedirá el hash de cada bloque al otro nodo. Este le envía de 500 en 500. Después el nodo nuevo le pide a cada vecino una cantidad equitativa de bloques con dichos hashes. Esto se hace para no sobrecargar la red. Una vez que llegan los primeros 500 bloques, se guardan en la blockchain y se repite el proceso hasta que el nodo este sincronizado. Después que el nodo esta sincronizado, ya puede participar como nodo activo en la red con las mismas funcionalidades, salvo la de crear bloques

En cuanto a tolerancia a fallas, esto teóricamente no es muy problemático ya que es una red p2p donde cada nodo tiene el mismo estado y si un nodo falla habrá otros. El problema es que para que se creen nuevos bloques en la red, tiene que haber al menos un nodo confiable participando en la red, aunque mientras más nodos confiables haya que puedan crear bloques, mejor y más rápido funcionara nuestro sistema. Notar que, aunque no haya nodos confiables activos, seguirá funcionando el sistema, lo que con lo que ya está creado y no con nuevos bloques.

En cuanto a seguridad, esto es un tema que no hay mucha necesidad de pensar cuando se usa blockchain, aquí lo complicado es la cantidad de métodos criptográficos que se usan, sobre todo las firmas digitales para poder asegurar que una persona es quien dice ser.

Una vez que el sistema está funcionando. Cada n minutos se crea un bloque, este se propaga por la red y cada nodo lo verifica, una vez verificado cada nodo lo reenvía por la red y así se propaga por la red. En cuanto a las identidades estas llegan a un nodo y se validan y se meten en un conjunto de identidades en caso de ser validadas y se reenvían por la red. De esta forma se propagan las identidades. Para crear bloques, se usa proof-of-authorization, con este algoritmo con que un solo nodo confiable valide un bloque o identidad ya basta, pero en nuestro sistema para dejamos que todos lo hagan para que sea más homogéneo. Entonces la estrategia que usamos para crear nuevo bloque es simple. Cada nodo debe tener una cola de nodos confiables, y en el caso de que un nodo sea confiable, cuando recibe un bloque, pregunta quien fue el creador y revisa su posición en la cola y verifica si es el siguiente en crear un bloque.

El sistema lo probamos localmente, creando varios proyectos iguales, y corriéndolos. Entonces intentamos probar todos los casos posibles que pudieran ocurrir manualmente, claro que pueda que algún caso no hayamos probado por la falta de tiempo. Pero pienso que lo esencial de casos que pudieran ocurrir que puedan romper el sistema los probamos. Esto mismo lo hicimos con tres laptops, conectadas por wifi y todo funciono igualmente. Intentamos probarlo con un teléfono, ya que nuestro sistema está hecho en Python 3.5, no funciono porque la versión más nueva de Python que encontramos para Android fue 3.2 y los problemas que hubieron fueron de implementación de Python por la versión, pero lograba conectarse igualmente y sincronizarse.

POSIBLES MEJORAS

Este sistema podría mejorarse, o ampliar su funcionalidad con autorizaciones de una persona a otra, o sea que una persona pueda autorizar a otra de hacer o poseer algo y luego verificarse en la blockchian si es real. También se podría usar como una plataforma para logearse, sería algo sencillo el hecho de entrar a un sitio por medio de nuestro sistema, lo interesante es que se podría lograr un mecanismo que se pueda identificar si la persona que quiere acceder al sitio es en realidad quien dice ser, por medio de la llave privada de cada usuario y firmas digitales. O sea, con este mecanismo los nombre usuario y contraseña quedarían obsoletos.