Blockchain come certificatore

Giorgio Mecca

14 settembre 2021

Sommario

...

Indice

Capitolo 1

Introduzione

1.1 Descrizione del Progetto

1.2 Descrizione dell'azienda

Capitolo 2

Blockchain

2.1 Problema dei generali bizantini

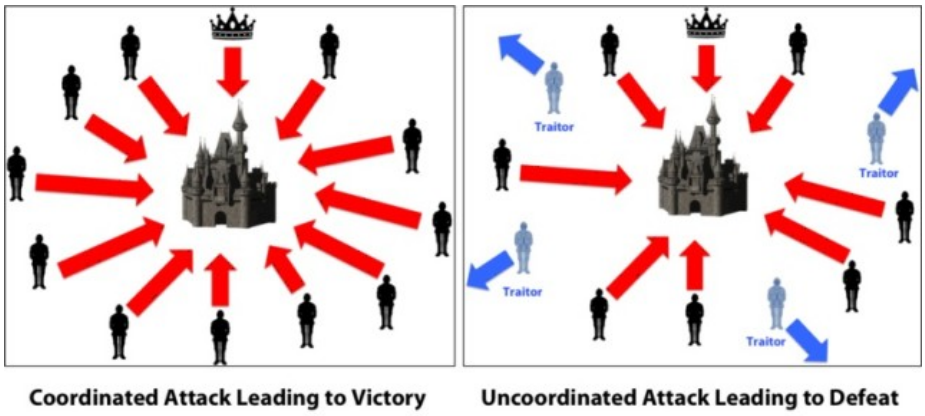

Figura 2.1: Problema dei generali bizantini

Il problema dei generali bizantini è un problema informatico su come raggiungere consenso in situazioni in cui è possibile la presenza di errori. Il problema consiste nel trovare un accordo, comunicando solo tramite messaggi, tra componenti diversi nel caso in cui siano presenti informazioni discordanti. Il problema è stato teorizzato dai matematici Leslie Lamport, Marshall Pease e Robert Shostak nel 1982, i quali crearono la metafora dei generali, caso di studio molto utilizzato nei sistemi basati o che comunque utilizzano una network. La metafora si basa su diversi generali che durante un assedio sono sul punto di attaccare una città nemica. Essi sono dislocati in diverse aree strategiche e possono comunicare solo mediante messaggeri al fine di coordinare l'attacco decisivo (Figura 2.1). I generali possono attaccare o ritirarsi, l'importante e che ci sia una decisione unanime, l'utilizzo di sola metà forza bellica porterebbe ad una sconfitta o una perdita. Il problema risiede quindi nell'alta probabilità che tra questi vi sia un generale traditore che mandi messaggi che vanno contro la strategia dell'esercito. La possibile soluzione punta al trovare un meccanismo secondo il quale un generale non traditore che riceva più messaggi sappia rico-

noscere quello veritiero. Secondo l'articolo di Lamport, Shostak e Pease non esiste una soluzione perfetta se il numero di processi non corretti è maggiore o uguale a un terzo del numero totale di processi. Una soluzione proposta è quella di Nakamoto che in una sua relazione sulla blockchain descrive un meccanisco per arrivare al consenso chiamato PoW Proof of Work.

2.2 Struttura di una blockchain

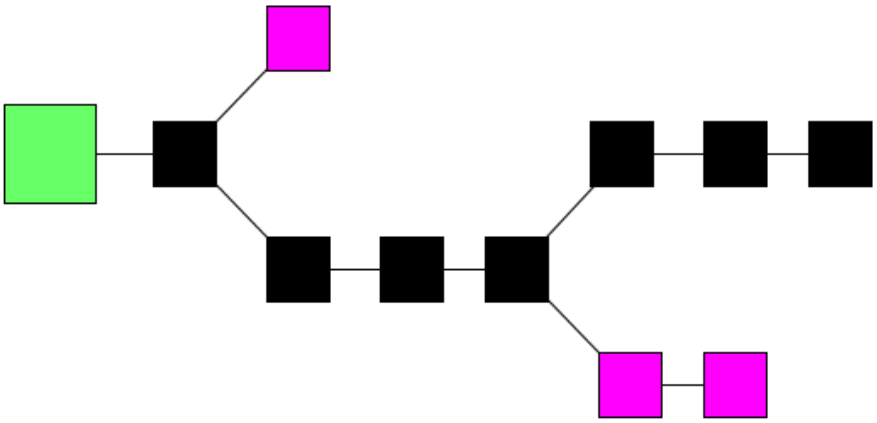

Figura 2.2: Rappresentazione struttura di una blockchain

Una Blockchain come suggerisce l'etimologia della parola è una catena di blocchi. È una struttura dati formata da un insieme di blocchi(struttura prioritaria) collegati univocamente 1 ad 1 così da creare una metaforica catena. Una blockchain è considerata una struttura condivisa e immutabile in quanto il suo contenuto una volta scritto non è più né modificabile né eliminabile, a meno di non invalidare l'intera struttura. Questa tecnologia fà parte dei Distributed Legder cioè dei "libri mastro" o registri condivisi infatti tutti i partecipanti della blockchain, detti anche nodi, posseggono lo stesso registro cioè le stesse informazioni andando a costruire il contrapposto di una struttura centralizzata come un Database, quindi una struttura Decentralizzata in cui ogni nodo ha la possibilità di leggere autonomamente le informazioni contenute. Nella figura 2.2 viene visualizzata una semplice blockchain in cui sono presenti tre tipologie di blocchi quali, il blocco verde visto come il blocco di genesi, i blocchi neri che vanno a costituire la catena principale e i blocchi viola considerati blocchi orfani. L'aggiunta di un nuovo blocco è globalmente regolata da un protocollo condiviso e se autorizzata ogni nodo aggiorna la propria copia privata del registro così da evitare manipolazioni future

2.3 Hashing

Un codice hash è una qualunque sequenza di caratteri alfanumerici generati da una particolare funzione di hash. Questa funzione prende in input un qualun-

que tipo di informazione e restituisce una stringa di lunghezza prefissata, questo rende la funzione one-way o non invertibile in quanto conoscendo il digest(codice hash restituito) non è possibile risalire all'informazione che lo ha generato. In una blockchain l'Hash viene utilizzato per la costruzione della catena, viene calcolato l'hash di un blocco e il blocco che lo succederà avrà come parametro questo hash. In questo modo ogni blocco è legato univocamente al blocco precedente e siccome il codice hash di un blocco viene calcolato utilizzando anche il codice hash precedente modificando un singolo blocco verrà invalidata tutta la struttura blockchain immediatamente successiva.

2.4 Transazioni

2.5 Blocchi

2.6 Mining e Meccanismi del consenso

2.6.1 Consenso Trustless

2.6.2 Proof of Work

2.6.3 Proof of Stake

2.6.4 Proof of Authority

2.7 Attacchi

2.7.1 Selfish Mining Attack

2.7.2 Double Spending Attack

2.8 Blockchain Pubbliche/Private

2.9 Etherium

2.9.1 Smart Contract

2.9.2 Solidity

2.9.3 Gas

2.9.4 Dapps

Capitolo 3

Tecnologie utilizzate

3.1 Besu

3.1.1 IBFT

IBFT Methods

3.1.2 Free Gas Network

3.1.3 API Methods

3.2 Truffle

3.2.1 Compile

3.2.2 Test

3.2.3 Deploy

3.3 Node js

3.3.1 Web3

Capitolo 4

Caso d'uso

4.1 Problema Iniziale

Il progetto blockchain è stato ideato e sviluppato come proposta di soluzione ai problemi nella gestione del tracciamento, invio, certificazione e mantenimento di dati riguardanti "spostamenti". Questi spostamenti sono informazioni(LOG) inviate da un qualunque ente che metta a disposizione della società o di un qualunque servizio che preveda dei mezzi pubblici o privati quali ad esempio autobus, treni, taxi etc... Queste informazioni vengono ora raccolte, analizzate e utilizzate su di modelli di storage a fogli di calcolo (Ad Esempio EXCEL - Programma Micrsoft). I fogli di calcolo offrono alcuni vantaggi come la semplicità con cui vengo creati, scritti e salvati benché offrano un'interfaccia poco user friendly('facilmente utilizzabile') guardando tutti i possibili attori, mentre il progetto si focalizza sui difetti come la pubblicazione/condivisione delle informazioni o l'interrogazione di queste in quanto utilizzando semplici fogli non vengono proposte alcune regole di struttura e organizzazione,e ci si pone particolare importanza alla sicurezza e all'affidabilità di queste informazioni e che non vengano modificate durante la condivisione, quindi la possibile certificazione di essi.

4.2 Soluzione

La soluzione proposta si offre di risolvere tutti i problemi sopra elencati come la certificazione, salvataggio e interrogazioni di informazioni. Viene ideata una blockchain privata che avrà funzione di ente (decentralizzato) certificatore, questa non utilizza nessuna moneta creando una Free Gas Network e che con l'ausilio di appositi smart contract(scritti e caricati autonomamente) ci permette di salvare un codice che andrà ad identificare un determinato gruppo di spostamenti come un codice Hash che usufruendo della struttura e utilizzo della blockchain non potrà essere modificato, ciò implica che si potrà sempre verificare la correttezza del gruppo di spostamenti richiesti ricreando e controllando il loro codice.

La memorizzazione dei dati viene invece affidata ad un database relazionale, utilizzando nel progetto il DBMS(DataBase Managenent System) MySql, che

ci permette di salvare grandi quantità di dati con una efficiente organizzazione gestita con la creazione di tabelle così da essere facilmente interrogabile in futuro.

L'interfaccia comune è gestita con un server sviluppato tramite tecnologia node js che con una single Page Application avrà la funzione di interfaccia user friendy con funzioni di memorizzazione per i log degli spostamenti su Database, calcolo e salvataggio dei loro codici hash sulla blockchain al tempo stimato e quando necessario, cioè quando e ovunque verranno richiesti dei dati e avverrà l'interrogazione del DataBase sarà reso obbligatorio il controllo di questi con il codice sulla blockchain. Inserendo questo WebServer intermedio o server proxy si andrà ad eliminare il passaggio di dati non propriamente protetto e rende partecipi tutti i singoli attori dell'attività.

4.2.1 Problematiche

Utilizzando delle nuove tecnologie sorgono comunque nuove problematiche che non sono state affrontate nello sviluppo in quanto non inerenti ai fini del progetto.

Una prima problematica si sviluppa utilizzano un server proxy. Avendo un singolo server di accesso al database a alla blockchain sequesto non viene correttamente protetto e costantemente controllato è soggetto ai classici attacchi come un DDOS - Distributed Denial of Service in cui si utilizzano molteplici messaggi fittizi (come un inizio di HandShake per una connessione TCP) per far si che il server non possa sostenere tutti i servizi e essendo l'unico punto di accesso bloccherebbe l'intero accesso alla rete blockchain.

Una caratteristica che rende sicura la blockchain pubblica è la molteplicità di nodi, questa con una blockchain privata come la nostra va a decadere con il discendere del numero di nodi; utilizzando un meccanisco di conseso basato su PoA(Proof of Autority) si ha infatti bisogno di un minimo di 4 nodi per essere resistente al problema bizantino.

Per evitare l'appesantimento della Blockchain si è pensato di salvare su di essa solo un codice identificativo(codice Hash) per un gruppo di Log. Questo implica che con l'aumentare dei log identificati da un singolo codice hash diminuisca la sicurezza che questo apporta infatti sarà più facilmente utilizzabile un attacco come l'attacco del compleanno che ha come obiettivo quello di generare una collisione cioè di trovare dei dati fittizi ai Log originari che però generano lo stesso codice Hash, questi dati fittizi potranno essere quindi sostituiti nel DB ma verranno comunque considerati certificati dal sistema in quanto produrranno lo stesso codice.

4.3 Attori

Il caso d'uso per il progetto blockchain prevede la partecipazione di diversi attori quali:

Un Terminal User o utente finale è un comune dipendente di un ente che partecipa alla blockchain il quale ha il compito di comunicare i propri spostamenti/Log o qualunque informazione di cui si preveda il salvataggio;

Gli Admin sono dei dipendenti di enti partecipanti che vengono segnati dagli stessi come amministratori che quindi posseggono particolari oneri come il possesso e la trasmissione di una chiave privata;

Il proprietario/gestore della blockchain avrà il compito di gestire l'intera blockchain privata con l'amministrazione che ne segue, come la supervisione dei nodi presenti, il loro funzionamento e la loro caratterizzazione come validatori.

4.4 Scenario di utilizzo

Il Progetto prevede uno scenario di utilizzo diverso seguendo la distinzione degli attori. Per l'utilizzo si prevede che ad ogni ente partecipante al progetto gli venga assegnato un account, cioè una copia di chiavi privata e pubblica che serviranno per interagire con la blockchain, inoltre ogni ente dovrà inserire i propri dipendenti nel Database e specificare il ruolo di essi, se admin o Terminal User.

Un Terminal User, una volta effettuato l'accesso, viene portato ad un'interfaccia in cui può inserire la città che sarà selezionata come Start dello spostamento e in seguito viene spostato in una seconda interfaccia da cui può terminare lo spostamento o annullarlo, se annullato potrà cominciare un nuovo spostamento dalla precedente interfaccia, il completamento di questo avverrà solo se compila i campi necessari quali la città di Termine e la distanza percorsa indicata in Kilometri.

Un admin, una volta effettuato l'accesso, potrà a differenza di un Terminal User effettuare delle query/ interrogazioni riguardo gli spostamenti compiuti, inserendo una data otterrà tutti gli spostamenti che sono stati certificati da una transazione inserita in quella determinata data, da qui potrà anche accedere ai dettagli della transazione o del blocco che la contiene riferendoci alla blockchain, inoltre, quando il sistema lo richiede, ha il compito di inserire la Private Key dell'utente(ente) che verrà utilizzata per la scrittura su blockchain.

Capitolo 5

Sviluppo

5.1 Schema Progetto

5.2 Blockchain Ibrida

5.3 Smart Contract

5.3.1 Boxing

5.4 WebApp

5.4.1 Single Page Application

5.4.2 Input

Inserimento in un DB

Inserimento nella Blockchain

5.4.3 Output

Report di Controllo

Monitor Blockchain

Capitolo 6

Sviluppi futuri

6.1 Analisi costi

6.2 Immissione nella blockchain pubblica

6.3 Blockchain pubblica come certificazione

6.4 Svilupo full Blockchain

Elenco delle figure