UNIVERSITÀ DEGLI STUDI DI TRENTO

Corso di Laurea in Informatica

La gestione di ruoli e permessi sulla blockchain Ethereum

Supervisore
Prof. Alberto Montresor

Laureando
Matteo Golinelli

Anno accademico 2018/2019

Digicando

Fornisce servizi di:

- Anticontraffazione, certificazione e tracciabilità di prodotti
- Analisi statistica sugli acquirenti dei prodotti
- Recensioni affidabili e certificate

Architettura

Attuale

L'architettura dell'implementazione attuale del servizio di Digicando è *client-server*

Futura

L'obiettivo di Digicando è spostare il servizio di anticontraffazione sulla *Blockchain Ethereum*

Motivazioni

L'utilizzo della tecnologia blockchain fornisce un'infrastruttura in grado di registrare e certificare i trasferimenti di proprietà di un prodotto in maniera:

- Sicura
- Immodificabile
- Trasparente
- Aperta al controllo di chiunque

Svantaggi

L'utilizzo della blockchain introduce una serie di problematiche:

- Mancanza del concetto di segretezza
- Scalabilità della blockchain stessa
- Possibile introduzione di leggi che limitino le possibilità di utilizzo delle criptovalute

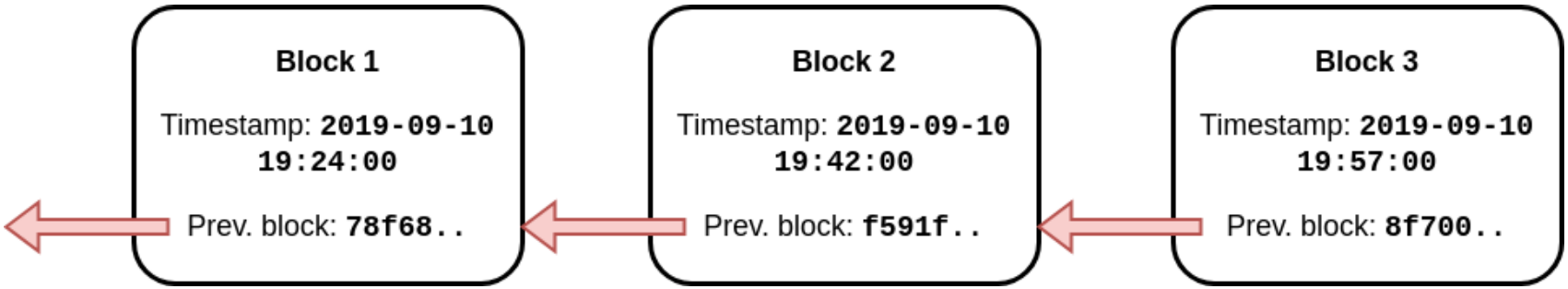
Blockchain

- La Blockchain è una struttura dati *distribuita*
- Ha come obiettivo il mantenimento di un registro immodificabile e crescente
- Il registro è composto da blocchi, connessi utilizzando la crittografia

Blockchain

Ogni blocco contiene:

- Un hash crittografico del blocco precedente
- Un timestamp
- Dati riguardanti delle transazioni



Blockchain

La blockchain è:

Distribuita

Si dice distribuita in quanto ogni partecipante alla rete ne condivide una copia identica

Immodificabile

Una blockchain è immodificabile per definizione, in quanto ogni modifica invaliderebbe crittograficamente l'intera struttura

Ethereum

- Ethereum è una blockchain open-source con criptovaluta nativa chiamata *Ether*
- L'Ether è una moneta digitale non è controllata da alcun governo o azienda
- Ethereum è programmabile:
  - E' possibile sviluppare ed eseguire applicazioni decentralizzate (*DApp*)
  - Beneficiano della criptovaluta e della tecnologia blockchain

Problema

- Durante la fase di sviluppo dell'applicazione si è data priorità allo sviluppo del core dell'applicazione
- Alcune caratteristiche e funzioni sono state tralasciate per il futuro
- La principale limitazione è l'impossibilità di gestire i contratti da più indirizzi
  - Costringe a mantenere una chiave privata condivisa
  - Limita la flessibilità del sistema nell'aggiunta e rimozione degli utenti autorizzati

Problema

- Inizialmente la gestione dei ruoli è stata realizzata mediante un sistema di *whitelisting* all'interno dei contratti stessi
- Irrigidisce l'architettura nel momento di eventuali aggiornamenti dei contratti
- Aumenta la complessità dei contratti interessati, costretti a svolgere compiti non correlati al loro scopo

Requisiti

I requisiti del sistema più importanti individuati insieme all'azienda sono:

- Il sistema deve lavorare interamente su blockchain
- Il sistema deve essere dinamico sulla creazione di nuovi ruoli
- I ruoli devono poter essere gestiti in maniera gerarchica

Soluzione

La soluzione individuata prevede:

- Due contratti esterni, in sostituzione del sistema di whitelisting degli indirizzi
- Un approccio basato su claim
  - Gli utenti possono attribuire claim ad altri utenti
  - Le claim conferiscono particolari privilegi

Tecnologie utilizzate

Le tecnologie utilizzate per lo sviluppo del sistema sono open-source e sono:

- *Solidity*: linguaggio di programmazione per la blockchain Ethereum
- *Truffle*: ambiente di sviluppo, framework per il testing, insieme di risorse
- *Ganache*: strumento per la creazione di blockchain personali per il testing

Contratti

ClaimsRegister

- Implementa un registro che memorizza i ruoli emessi, cioè le *claim*
- Implementa una funzione per il controllo di validità delle singole claim ed una per la loro rimozione

Contratti

ClaimsManagement

Permette di gestire delle strutture organizzative gerarchiche e complesse

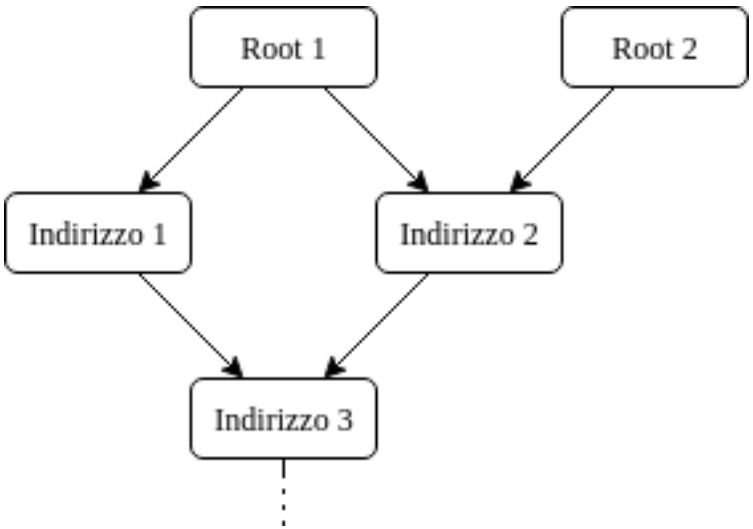


Figure: Un esempio di gerarchia complessa

Registro

Il registro è un `mapping`, cioè una tabella hash nel linguaggio Solidity. Memorizza un valore associato a tre chiavi:

- *Emitter*: l'indirizzo che emette il ruolo
- *Receiver*: l'indirizzo che riceve il ruolo
- *Claim*: la chiave di un ruolo

```
// emitter => receiver => claim => value
mapping(address => mapping(address => mapping(bytes32 =>
    bytes32))) public registry;
```

Integrazione

Fasi del processo di integrazione:

- Studio dell'implementazione attuale
- Individuazione dei contratti che fanno uso del sistema di whitelisting
- Sostituzione del vecchio sistema con il nuovo
- Testing approfondito del sistema integrato

Conclusioni

- Il sistema sviluppato, composto dai due contratti, ha risolto il problema
- Il sistema di Digicando così integrato risulta più flessibile, funzionale e performante