STUDIO SU SISTEMI DISTRIBUITI E DECENTRALIZZATI BASATI SU BLOCKCHAIN

Luca Vecchi - Matricola 830718

13 Aprile 2017

Il presente lavoro è frutto di un progetto presso il Dipartimento di Informatica dell'Università di Milano sotto la supervisione del Dott. Trentini Andrea Mario ed è lo studio dell'applicazione di concetti quali *blockchain*, *Dapp*, attraverso la realizzazione di un software "smart-contract" finalizzato alla gestione dei diritti d'autore sulle opere dell'ingegno. Scopo della presente ricerca è stato costruire un'organismo autonomo per la gestione delle opere autoriali che non possa essere soggetto a censura, frodi, manomissioni e, per mezzo della criptovaluta, un autore possa ottenere un guadagno economico dalla registrazione ed utilizzo dell'opera stessa da altri utenti.

L'idea della registrazione e gestione delle opere autoriali mediante sistemi informativi centralizzati non è nuova e possono essere trovati in rete numerosi esempi di piattaforme e compagnie di *escrow*[1].

Gli escrow, come i sistemi centralizzati che permettono l'interazione tra più soggetti, proteggono le transazioni ma allo stesso tempo, poiché concentrano le informazioni in un unico point-of-failure, espongono le stesse al rischio di censura, frode e manomissioni.

Il presente lavoro si basa sul concetto di blockchain, database immutabile, distribuito, protetto da crittografia e accettato tramite consenso da un sistema decentralizzato di peer. La natura crittografica della blockchain permette di costruire particolari meccanismi e automazioni, tra cui gli *smart contract*, che consentono, attraverso una transazione, di trasferire digital asset da una persona ad un'altra.

La transazione non avviene tramite una terza parte fidata (come può essere una banca), ma viene presa in carico, anche nello stesso istante, da uno o più nodi, indipendenti l'uno dall'altro, chiamati *miner*, i quali, verificano la sua validità, raggruppandola con altre in blocchi concatenati tramite precise regole condivise chiamate *regole di consenso*.

Inizialmente la registrazione e verifica della proprietà di opere autoriali era garantita da enti centralizzati come la *SIAE*; con l'avvento della blockchain ed Ethereum, i quali permettono l'esecuzione di smart contract in una piattaforma decentralizzata, la gestione della proprietà delle opere avviene in modo imparziale e sicuro.

Nel presente lavoro si è voluto analizzare lo sviluppo di una *proof of concept*, simulata in una rete privata e isolata dalle blockchain in Internet.

Inizialmente il nodo che registrava le transazioni, eseguiva gli smart contract e minava i blocchi era gestito tramite l'applicativo *Testrpc*, il quale simula l'esecuzione di un *full-node*[2] rendendo la fase di sviluppo più veloce.

Tuttavia, è stato osservato che esso non consentiva di simulare situazioni reali creando una rete privata di nodi, di conseguenza è stato deciso di utilizzare *Geth*, software full-node

[1] Escrow o acconto di garanzia è un contratto tra una o più parti che stipulano un accordo nel quale un bene, materiale o immateriale, viene depositato presso una terza parte fidata (la compagnia di escrow) che veglierà sul rispetto delle clausole

[2] Full-node: indica le funzionalità di un nodo della blockchain, la cui principale è la capacità di minare nuovi blocchi. Si contrappone al *light-node* che non possiede questa funzionalità, ma consente solamente di connettersi e scambiare messaggi con altri nodi.

più in uso e completo.

Ciò ha permesso di assumere il pieno controllo delle operazioni di mining, della connettività di rete e predisporre di una console a linea di comando con cui è stato possibile testare varie funzionalità tra cui creare e distruggere account, firmare ed eseguire le transazioni, ispezionare lo stato delle transazioni e dei blocchi.

Lo smart contract è scritto in *Solidity*, un linguaggio ad oggetti Turing completo, che una volta compilato in bytecode permette l'esecuzione di codice all'interno della Ethereum Virtual Machine o EVM.

Per sviluppare gli smart contract si è optato per l'utilizzo di *Solidity browser*, un'interfaccia web che fornisce un ide di sviluppo e vari strumenti per l'interazione e testing dei contratti, il deploy, la compilazione e la verifica formale automatica del codice scritto.

Allo scopo di ottenere il massimo controllo sull'operazione di deploy, il codice sorgente è stato compilato tramite il compilatore *Solc* e, successivamente, importati i contratti direttamente dalla console di Geth.

Per il testing, sono stati sviluppati alcuni test di unità in *Mocha* che hanno permesso di approfondire il funzionamento del linguaggio Solidity.

Lo smart contract così implementato permette la registrazione e il trasferimento, anche temporaneo, della proprietà di un opera da uno specifico account Ethereum ad un altro. Un'opera è riconoscibile attraverso i metadati che un utente inserisce in fase di registrazione dell'opera. Ognuno connesso alla blockchain può verificare la storia della catena della proprietà e i vincoli pendenti su un'opera.

Sulla blockchain risiede il contratto *Digichain* che impersona un'organizzazione autonoma per la gestione delle opere alla quale un autore può caricare la propria opera sotto forma di metadati. L'utente può acquisire e registrare in un contratto l'utilizzo dell'opera versando un corrispettivo in Ether (la valuta di Ethereum). L'autore può verificare lo stato della cessione dell'opera e in particolari casi, Digichain provvede a rimuovere il contratto autonomamente restituendo il denaro a specifici soggetti.

E' stato implementato il pattern *System of Contracts* che permette la modifica del contratto Digichain già distribuito mediante un contratto di registrazione.

Inoltre è stato aggiunto il contratto *GlobalRegistrar* che implementa un sistema di risoluzione dei nomi. Un utente può registrare l'indirizzo di un contratto e associarli un nome simbolico.

Se da una parte lo sviluppo del progetto Ethereum sia ancora all'inizio e i recenti avvenimenti che l'hanno coinvolto pongono vari interrogativi su quanto possa essere stabile e sicuro sviluppare applicazioni decentralizzate su questo sistema, dall'altra, invece, le possibilità di sviluppare applicazioni su blockchain, i grandi investimenti finanziari ricevuti per il proseguimento dello sviluppo di Ethereum e le collaborazioni con grandi nomi del settore IT dimostrano una grande fiducia nel progetto e nelle tecnologie basate su blockchain.

La progettazione di smart contract in Ethereum è stata guidata sia dai limiti imposti dal nuovo linguaggio di programmazione e dal paradigma di Ethereum dove qualsiasi computazione ha un prezzo, sia da pattern orientati ai contratti che permettono di oltrepassare vari limiti come l'impossibilità di aggiornare il codice dei contratti, sia dall'assenza di meccanismi decentralizzati per creare eventi temporizzati.

L'applicazione può essere un punto di partenza per ulteriori ricerche riguardanti la sua interazione con i browser, che attraverso plug-in, permetterebbe la verifica automatica del diritto d'uso delle opere digitali contenute in un sito.