# Studio tecnologico Identity Trust Fabric

## Diario delle modifiche

|  |  |  |
| --- | --- | --- |
| Data | Descrizione | Autore |
| 12/06/18 | Creazione documento. Scrittura dei capitoli Descrizione prodotto, scopo del documento e ITF, Permissionless e Permissioned blockchain, Ethereum | Simone Ballarin |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Scopo del documento

Nell’ottica di un prossimo sviluppo di un’estensione basata su blockchain dell’attuale servizio ora in produzione MonoKee, questo documento intende valutare in termini di benefici tecnici e tecnologici l’uso della tecnologia Ethereum quale base dell’Identity Trust Fabric (ITF).

## Riferimenti

1. Blockchain: The Dawn of Decentralized Identity (G00303143), Homan Farahmand per Gartner
2. Sito ufficiale Ethereum, [www.ethereum.org](http://www.ethereum.org)
3. bornonjuly4.me/2017/01/10/blockchain-what-is-permissioned-vs-permissionless
4. [www.cryptoitalia.org/bitcoin-il-51-attack](http://www.cryptoitalia.org/bitcoin-il-51-attack)
5. www.solidity.readthedocs.io

## Descrizione prodotto

Il progetto ha come scopo la creazione di un’estensione del servizio MonoKee basato su blockchain. L’estensione offre un sistema di Identity Access Management (IAM) composto da quattro principali fattori:

1. Identity Wallet (IW)
2. Service Provider (SP)
3. Identity Trust Fabric (ITF)
4. Trusted Third Party (TTP)

In sintesi l’estensione dovrà operare al fine di fornire la possibilità ad un utente di registrare e gestire la propria identità tramite l’IW autonomamente, mandare i propri dati (IPP) all’ITF la quale custodirà la sua identità e farà da garante per le asserzioni proveniente dai TTP. Inoltre il SP dovrà essere in grado con le informazioni provenienti da IW e ITF di verificare o meno l’accesso ai propri servizi.

## ITF – Identity Trust Fabric

Sulla base di un primo studio di fattibilità l’unico componente coinvolto nell’uso blockchain è l’Identity Trust Fabric. La sua principale funzione è quella di poter permettere ai vari Service Provider aderenti al servizio di poter verificare le informazioni rilasciate dai vari utenti tramite l’utilizzo del loro personale Identity Wallet (IW). Il componente mantiene al suo interno l’hash della chiave pubblica degli utenti (che rappresenta la loro identità) e le asserzioni fornite dai vari IW che possono essere potenzialmente certificate da una TTP (tramite una firma con la loro chiave privata). Le asserzioni devono poter essere modificate o eliminate in ogni momento, ovviamente ogni alterazione deve essere ogni volta certificata nuovamente. Anche da parte del TTP ci dev’essere la possibilità di revocare la certificazione di un’asserzione.

Una buona implementazione di una ITF deve possedere le seguenti caratteristiche:

1. Fiducia: il contenuto presente nella ITF deve essere solo quello autorizzato, non ci devono poter essere manomissioni malevoli da parte degli utilizzatori della rete. Ogni componente deve potere aver fiducia nella veridicità dei dati.
2. Garanzia: le regole logiche della ITF non devono poter essere manomesse. Deve essere possibile applicare le varie policy aziendali in ambito di gestione dei rischi.
3. Tracciabilità: ogni informazione e cambio di stato relativo alle identità e alle asserzioni deve poter essere tracciato e verificabile sia in termini cronologici sia in termini di provenienza.
4. Sicurezza: intesa come CIA. L’ITF deve rispettare i vincoli di confidenzialità, inalterabilità e disponibilità delle informazioni dentro lei contenute.
5. Scalabilità: l’ITF deve fornire un elevato grado di scalabilità soprattutto in un’ottica in cui il prodotto potrebbe essere reso disponibile ad un uso Consumer.
6. Efficienza: il funzionamento dell’ITF deve richiedere la minima quantità di risorse possibili.

## Permissionless e Permissioned blockchain

Al fine di poter valutare la fattibilità dell’utilizzo di Ethereum quale blockchain sottostante all’ITF è necessario avere in mente le due principali categorie di blockchain: **permissionless** e **permissioned**.

### Permissioned blockchain

Una permissioned blockchain pone dei vincoli sulla partecipazione alla rete. Soli i nodi autorizzati possono partecipare all’algoritmo di consenso dei blocchi. Le autorizzazioni possono essere date singolarmente, quindi i vari nodi possono avere o meno le seguenti possibilità:

* lettura dei blocchi;
* scrittura dei blocchi;
* esecuzione di codice (se prevista dalla blockchain);
* verifica dei nodi.

### Permissionless blockchain

Una permissionless blockchain è una rete in cui qualsiasi nodo può partecipare al processo di verifica dei blocchi. Ogni nodo ha tutte le precedenti quattro proprietà.

## Ethereum

Ethereum è una piattaforma decentralizzata pubblica ed open-source basata sulla creazione di SmartContract. Permette la creazione di applicazioni che operano su blockchain in modo che non ci sia alcuna possibilità di downtime, censura, frodi o interferenze da terze parti. Rappresenta una dei principali esempi di rete permissionless.

La piattaforma è stata rilasciata nel corso del 2014 ed è mantenuta dalla Ethereum Foundation. Questo fa di Ethereum una delle più longeve blockchain disponibili. Ciò comporta la presenza di una documentazione abbastanza nutrita rispetto ai competitor e di un buon numero di strumenti già disponibili.

### Strumenti

Come già citato la relativa maturità della tecnologia ha portato alla creazione di alcuni utili strumenti.

Truffle: è una suite di development e testing. Permette di compilare, buildare ed effettuare la migrazione degli SmartContract. Inoltre ha funzioni di debugging e di scripting. La suite offre la possibilità di effettuare test degli SmartContract sia in Javascript (con l’utilizzo di Chai), sia in Solidity.

Ganache: è uno strumento rapido che permette di creare e mantenere in locale una rete blockchain Ethereum personale. Può essere usata per eseguire test, eseguire comandi e per operazioni di controllo dello stato mentre del codice esegue.

Metamask: è uno plugin disponibile per i browser Chrome Firefox Opera che permette di interfacciarsi alla rete Ethereum senza la necessità di eseguire in intero nodo della rete. Il plugin include un wallet con cui l’utente può inserire il proprio account tramite la chiave privata. Una volta inserito l’account il plugin farà da tramite tra l’applicazione e la rete.

## Valutazione applicabilità soluzione Ethereum

Al fine di poter valutare correttamente da ogni punto di vista l’applicabilità di una soluzione basata su Ethereum quale base della componente ITF, si procede ad analizzare in maniera analitica le sei caratteristiche presentate nel capitolo ‘ITF – Identity Trust Fabric’.

### Fiducia

Questa caratteristica è ottenuta da Ethereum da una combinazione di diversi fattori quali:

* utilizzo di incentivi economici, il pagamento per effettuare operazioni
* utilizzo di prove di interesse (Proof of Interest)

Le prove di interesse possono essere di due tipi:

* Proof of Stake, l’esibizione di un interesse;
* Proof of Work, l’uso di potenza di calcolo per risolvere un problema matematico.

Queste metodologie fanno in modo che solo chi realmente interessato possa influenzare l’algoritmo di consenso dei blocchi. Questo rende minore la possibilità di un 51% attack (nota 4). C’è comunque da ricordare che un attacco di questo tipo e praticamente impraticabile.

Per queste ragioni si ritiene una rete Ethereum completamente soddisfacente per quanto riguarda l’aspetto fiducia, al pari di una rete di tipo permissioned.

## Garanzia

Lo studio in nota 1 evidenzia come questo rappresenti un punto critico. Infatti riporta che il raggiungimento di questo obiettivo è fortemente condizionata dall’efficacia dell’algoritmo di consenso e dai nodi presenti nella rete. Prosegue facendo notare che la presenza di nodi malevoli oltre che mettere a rischio l’algoritmo di consenso può compromettere anche il corretto funzionamento. Infatti trattandosi di una blockchain pubblica ogni nodo è in grado di visionare il contenuto di ogni singolo contratto, inclusi i dati e i metodi presenti. Per quanto riguarda i dati questo potrebbe non essere un problema in quanto si può immagazzinare una versione codificata del dato. Per quanto riguarda i metodi questo non è possibile, questo potrebbe rendere in grado ad un attaccante di trovare eventuali bachi e criticità dell’ITF.

## Tracciabilità

Lo studio Gartner evidenzia come in una rete permissionless la tracciabilità temporale non sia possibile, in quanto in una rete distribuita ogni nodo può avere un concetto di tempo proprio. Questo però non risulta possibile in nessun approccia risolutivo all’ITF basato su blockchain. Infatti le reti permissioned applicano timestamp a livello di blocco e non di transazione, anche ammettendo che ci sia un concetto di tempo comune tra i nodi, le transizioni rimarrebbero temporalmente non tracciabili. La cosa potrebbe per mettere ad un blocco di alterare l’ordine delle transazioni.

Questo problema in una rete permissioned può essere risolto creando blocchi immutabili e ogni volta si voglia fare una modifica si dovrà creare un nuovo blocco. In questo modo ci sarà solo una transazione di creazione blocco il cui timestamp coinciderà con il timestamp del blocco.

Questo approccio in Ethereum rimane comunque impraticabile. Attualmente non sono note ulteriori tecniche per la tracciabilità temporale in Ethereum. Per questa ragione l’attribuzione di un riferimento temporale dovrà essere effettuato lato client, con i conseguenti limiti di sicurezza.

## Security

La confidenzialità dei dati anche se non presente nativamente in Ethereum e facilmente ottenibile immagazzinando nei contratti solo un hash dei dati.

L’integrità dei dati invece è garantita dalla prova di lavoro che utilizza la blockchain come già ribadito nella sezione Fiducia.

La disponibilità invece è garantita dalle caratteristiche di distribuzione di ogni blockchain.

Un ulteriore punto di considerazione da fare è che chiunque ha la possibilità di vedere il contenuto di ogni SmartContract incluso il codice dei metodi. Questo come già detto può comportare la possibilità da parte di un attaccante di individuare eventuali errori logici. Ogni contratto dovrà comunicare con gli altri attraverso chiamate a metodi pubblici, in quanto non c’è in Ethereum nessun concetto di visibilità dei metodi di tipo protected o package. Questo rende possibile da parte di qualsiasi utente della rete di utilizzare questi metodi in maniera malevole. Questo tipo di problematica è facilmente superabile applicati i dovuti pattern Solidity quali WhiteList Pattern e Owner Pattern. L’applicazione dei pattern però comporterebbe un notevole aumento in termini di complessità e costo soprattutto in presenza di logiche di accesso variegate e dinamiche. Inoltre, in caso di liste di utenti autorizzati l’immagazzinazione di queste liste potrebbe risultare oneroso in termini di costi.

## Scalabilità

Ethereum per poter applicare l’algoritmo del consenso fa utilizzo di una prova di lavoro. Questa deve essere fatta in occasione di ogni transazione. La prova consiste nella risoluzione di un problema crittografico la cui difficoltà è dinamica in base a diversi fattori della blockchain quali valore dell’Ether, numero di utenti, numero di transazioni, etc. Inoltre si nota come anche in lettura ci sia una lentezza che difficilmente potrebbe essere ritenuta accettabile da un utente medio. Per avere prova di questo fatto si può prendere in esame una qualsiasi Dapp presente al seguente link www.stateofthedapps.com.