



Università degli Studi di Salerno
Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

MedChain: un approccio blockchain per il trattamento dei dati clinici

Relatori

Prof.ssa Genoveffa Tortora

Candidato

Stefano Pisciottano

Matricola: 0512107488

A chi ha creduto in me, anche quando io non l'ho fatto

Indice

Abstract	1
1 La Blockchain	4
1.1 Di cosa si tratta	4
1.2 Cenni storici	5
1.3 Utilizzi reali	7
1.4 Ledger	7
1.5 Distributed Ledger Technologies	9
1.5.1 DLT Permissionless	10
1.5.2 DLT Permissioned	10
1.5.3 Differenze	11
1.6 Meccanismi di consenso	12
1.7 I blocchi	15
1.8 Le transazioni	16
1.9 Sidechain o second level chain	18
1.10 Sicurezza	19
1.10.1 Funzioni Hash e timestamping	19
1.11 Smart Contracts	21
1.12 Tendenze future	26
2 Sistema sanitario in Italia	28
2.1 Situazione attuale	28
2.1.1 Rapporto con gli altri paesi europei	30
2.1.2 Fascicolo sanitario elettronico	32
2.1.3 Blockchain come soluzione dei problemi del settore sanitario	33
2.2 PharmaLedger	34
2.2.1 Obiettivi e casi d'uso	34
2.3 Hyperledger Fabric	36
2.3.1 Introduzione	36
2.3.2 Caratteristiche principali	37
2.3.3 Architettura	37

3 MedChain, la blockchain in ambito sanitario	41
3.1 Permessi e accesso	42
3.2 Struttura	44
3.2.1 Public key infrastructure e Membership Service Provider	44
3.2.2 Gestione della piattaforma	45
3.2.3 Crittografia	47
3.2.4 Amministrazione dei dati off-storage	47
3.3 Casi d'uso	48
3.3.1 Interazione dei vari attori all'interno del sistema	48
3.3.2 Caso d'uso paziente	48
3.3.3 Caso d'uso dottore	50
3.3.4 Creazione profilo utente	50
3.4 Prototipo del sistema	51
3.4.1 Dashboard paziente	51
3.4.2 Dashboard dottore	52
3.5 Ledger	54
3.5.1 Blockchain e World State	54
3.6 Implementazione	55
3.6.1 Prerequisiti	55
3.6.2 Fabric-samples	56
3.6.3 CouchDB	57
3.6.4 Rete ad alto throughput	57
3.6.5 Creazione dei vari componenti all'interno del sistema	62
3.6.6 Pazienti	63
4 Conclusioni	65
4.1 Innovazione apportata dalla blockchain	65
4.2 Trend futuri	66

Elenco delle figure

1.1	da sinistra a destra: Nick Szabo, Dorian Nakamoto, Craig Steven Wright, Hal Finney	5
1.2	Differenze tra logica centralizzata, decentralizzata e distribuita	9
1.3	Diversi tipi di DLT a confronto	10
1.4	Differenze tra i protocolli di consenso proof of work e proof of stake	14
1.5	Struttura dei blocchi all'interno della blockchain e collegamento tra di essi . .	16
1.6	Collegamento tra side chain e main chain attraverso un peg bidirezionale . . .	19
1.7	Esempio di uno smart contract scritto in Solidity	22
1.8	Probabilità di successo della doppia spesa, in funzione dell'hashrate dell'attaccante (q) e del numero di conferme n	23
1.9	Isolazione dei nodi all'interno della rete durante un attacco Eclipse	26
2.1	Ministero della salute: dati relativi al personale sanitario del 2017	29
2.2	Classifica finale Euro Health Consumer Index (Italia al 20° posto)	31
2.3	Costo dell'assistenza ospedaliera nei vari paesi europei	31
2.4	Conoscenza e utilizzo del FSE da parte dei cittadini e dei pazienti italiani .	32
2.5	Aziende farmaceutiche globali in collaborazione con il progetto PharmaLedger	34
2.6	Obiettivi principali del progetto PharmaLedger	35
2.7	Blockchain frameworks e tools di Hyperledger	36
2.8	Flusso delle transazioni in Hyperledger Fabric [25]	38
2.9	Processo di inoltro transazione e ricezione conferme da parte degli endorser .	39
3.1	Partecipanti all'interno della rete e relative autorizzazioni di accesso	42
3.2	Componenti di un'infrastruttura a chiave pubblica	44
3.3	Struttura dei meccanismi di accesso e di gestione della blockchain	45
3.4	Parti interessate nell'accesso al fascicolo sanitario elettronico di un paziente .	46
3.5	Salvataggio dei dati off-chain e meccanismo di controllo integrità dei dati . .	48
3.6	Use case diagram che coinvolge gli attori all'interno del sistema	49
3.7	Use case diagram paziente	49
3.8	Use case diagram dottore	50
3.9	Sequence diagram che mostra l'aggiunta di un paziente all'interno del sistema	51
3.10	Schermata di registrazione di un utente all'interno della rete	52
3.11	Richiesta cambio password in seguito al login da parte del paziente	52

3.12 Schermata di visualizzazione di tutti i medici che hanno richiesto l'accesso ai dati	53
3.13 Schermata di registrazione di un medico all'interno della rete	53
3.14 Schermata di accesso al sistema dopo aver effettuato la registrazione	54
3.15 Struttura del ledger, composto da un world state e dalla blockchain	55
3.16 Struttura delle cartelle nei Fabric samples	56
3.17 Funzione di aggiornamento all'interno del file high-throughput.go	58
3.18 Funzione di prelievo all'interno del file high-throughput.go	59
3.19 Funzione di taglio all'interno del file high-throughput.go	60
3.20 Funzione di cancellazione all'interno del file high-throughput.go	61
3.21 Creazione dei profili di connessione della rete per gli ospedali	62
3.22 Creazione dei profili degli utenti	63
3.23 Registrazione dei dati dei clienti (esclusivamente il ruolo di paziente)	64

Abstract

Il settore sanitario risulta essere compromesso da un grave problema: le informazioni riguardanti i pazienti, i dati e i referti clinici sono suddivisi in vari sistemi non comunicanti, obsoleti e costruiti anche in maniera differente. Questa mancanza di comunicazione tra vari livelli e strutture porta a un'inevitabile frammentazione dei dati e inconsistenza di essi, causando mancanze che si possono ripercuotere sui pazienti stessi. In questa tesi si pone come obiettivo l'analisi, la progettazione e la realizzazione di un sistema in grado di assicurare la corretta gestione dei dati attraverso l'integrazione, l'affidabilità e la sicurezza che caratterizzano la blockchain. Verrà analizzata la tecnologia, il suo funzionamento e la sua evoluzione, i punti di forza, le debolezze e caratteristiche principali, per poi focalizzarsi sul settore sanitario e sugli attuali problemi che lo caratterizzano. Sarà implementata una blockchain per il salvataggio dei dati dei pazienti, alla quale questi possono accedere liberamente e in qualunque momento, assumendo così pieno controllo dei propri referti. Con questa ricerca si mostrerà una soluzione permissioned basata su Hyperledger Fabric, in cui i pazienti sono al centro del sistema e sono consapevoli della gestione delle informazioni. La tesi mostra il potenziale dell'applicazione della blockchain nell'ambito sanitario e come può essere utile per la condivisione dei dati in maniera attendibile.

Introduzione

Introduzione

La blockchain nasce nell'ottobre del 2008 con la pubblicazione del white paper da parte di Satoshi Nakamoto. Questa tecnologia appartiene alla categoria delle *Distributed Ledger Technologies* e si basa sul concetto di decentralizzazione, immutabilità e sicurezza. La blockchain, come dice il nome stesso, è una catena di blocchi in cui ogni blocco è collegato al precedente tramite un valore hash, fino ad arrivare al blocco genesi. Attraverso la crittografia, i meccanismi di consenso e il registro distribuito permette di assicurare la fiducia tra le varie parti. I blocchi sono formati da transazioni, che consentono di scambiare gli asset tra le parti interessate. Questa tecnologia nasce per regolamentare gli scambi di valute digitali, ma le sue applicazioni non si limitano soltanto a questo. Sono infatti svariati gli ambiti in cui tale tecnologia trova applicazione. In questa tesi sarà analizzata la struttura della blockchain e in particolare si ci focalizzerà sul settore sanitario, che presenta alcuni problemi nella gestione dei dati dei pazienti, dati spesso sensibili.

L'applicazione della blockchain in ambiente sanitario può:

- aiutare i pazienti a gestire in maniera consapevole i propri dati;
- permettere di tenere traccia delle cure somministrate ai pazienti, delle diagnosi, delle malattie pregresse;
- garantire una corretta applicazione delle cure.
- garantire la corretta gestione dei dati, rendendoli affidabili e sempre disponibili.

La tesi si pone come obiettivo la realizzazione di un'applicazione basata su blockchain, una soluzione che utilizza *Hyperledger Fabric* come framework open source per la creazione di un sistema permissioned in grado di garantire un certo livello di privacy e riservatezza per i pazienti. Si parlerà inizialmente della blockchain, e saranno analizzate tutte le sue caratteristiche. In seguito si tratterà la situazione del settore sanitario e verrà proposta la soluzione al problema della gestione dei dati attraverso l'utilizzo della blockchain. Grazie al sistema proposto i pazienti saranno in grado di evitare sprechi di tempo, e sarà garantito il controllo dei propri dati. Infine saranno messi in risalto i vantaggi annessi all'utilizzo di tale soluzione e saranno analizzati gli svantaggi.

L'elaborato di tesi è articolato in 4 capitoli.

- Nel capitolo 1 viene analizzata la storia della blockchain, dalla sua nascita nel 2008 fino a oggi. Si distingue tra soluzioni centralizzate, decentralizzate e distribuite e tra blockchain *permissioned* e *permissionless*, analizzando pregi e difetti di queste due strutture. In seguito vengono presentati vari algoritmi di consenso per l'approvazione delle transazioni nella blockchain (ad esempio Proof of Work e Proof of Stake) e viene presentata la struttura dei blocchi, come sono collegati e da quali elementi sono composti. Dopo aver parlato anche delle transazioni si analizzano le sidechain (strutture di supporto per le blockchain) e si spiegano i meccanismi che garantiscono la sicurezza nelle blockchain, come crittografia e funzioni hash. In ultima analisi si presentano i vari attacchi che possono minare questa tecnologia.
- Nel capitolo 2 ci si occupa di presentare un resoconto sulla situazione attuale che caratterizza il settore sanitario. Si parla del Sistema Sanitario Nazionale e della posizione dell'Italia, anche in confronto agli altri paesi europei. Questo settore in Italia ricopre una grande fetta di mercato in quanto produce più del 9% del PIL nazionale. Vengono quindi valutati i problemi che caratterizzano questo settore per quanto riguarda la gestione dei dati, che a volte risultano essere inconsistenti e ridondanti, causando spreco di risorse. Viene inoltre presentata l'azienda Pharmaledger, che ha mosso i primi passi in questo senso andando a proporre una blockchain per il supporto del settore sanitario. Il progetto mira a occuparsi di 3 casi d'uso in particolare ma può essere esteso a molti altri. Infine viene presentato il framework modulare per lo sviluppo di blockchain *Hyperledger Fabric*, che consente di creare una piattaforma permissionless e personalizzabile grazie alla modularità delle componenti.
- Nel capitolo 3 viene proposta una soluzione al problema presentato in precedenza. In particolare si vuole creare una blockchain per gestire, mantenere, salvare e recuperare i dati dei pazienti in maniera attendibile, sicura e immediata. La blockchain si avvale delle caratteristiche di Fabric, che consente di mantenere un alto grado di privacy anche grazie alla presenza di canali per la comunicazione privata tra le organizzazioni. In questo sistema l'utente ha il pieno controllo delle proprie informazioni, può modificare i propri dati e decidere se consentire o negare l'accesso alle proprie informazioni, sia ai medici che agli altri partecipanti.
- Nel capitolo 4 si mostrano le conclusioni del lavoro e si ribadisce che la blockchain potrebbe essere una soluzione valida per la gestione delle informazioni sensibili. Si analizzano poi i pro e i contro della soluzione proposta e si analizzano gli sviluppi futuri, in quanto la blockchain è una tecnologia ancora giovane e ci sono molti studi e ricerche a riguardo.

Capitolo 1

La Blockchain

Quando si parla di blockchain, non esiste una sola definizione, bensì ce ne sono diverse, come anche diverse interpretazioni e punti di vista. Può comunque essere vista come una struttura dati condivisa e "immutabile", una catena di blocchi contenenti le transazioni che vengono validate a seconda dei protocolli previsti dai meccanismi di consenso. In generale, le blockchain fanno parte della famiglia delle Distributed Ledger, registri distribuiti che vengono acceduti in lettura e scrittura dai nodi della rete. Siccome si tratta di un registro distribuito e non è presente un ente centrale che governa e gestisce il sistema, è necessario che i nodi, per poter validare e apportare modifiche al sistema, raggiungano il consenso.

1.1 Di cosa si tratta

Le caratteristiche essenziali delle tecnologie blockchain sono la trasparenza, la scalabilità, la tracciabilità e l'immutabilità delle transazioni, la sicurezza garantita dalla crittografia asimmetrica.

Si tratta di un database strutturato in blocchi che sono tra loro collegati. I nodi che ne fanno parte hanno il compito di verificare la validità delle transazioni portando alla creazione di una rete condivisa da tutti. Queste transazioni possono essere considerate immutabili (se non in alcuni casi specifici), infatti una volta approvata o validata una transazione, sarà salvata in maniera permanente all'interno del registro e sarà resa visibile a tutti i nodi.

Il mondo è basato sullo scambio di beni, che acquisiscono valore in base alla loro scarsità. Con l'avvento di internet c'è stato un problema per quanto riguarda ciò, siccome questi beni, una volta digitalizzati, sono stati resi disponibili e duplicabili da una molteplicità di persone. Questo ne ha intaccato la rarità a causa della possibile replicazione e diffusione. Se il dato viene invece crittografato (divenendo così un cripto asset) e scritto sul registro allora si preserva la sua unicità. La blockchain aiuta in tal senso consentendo di riprendere il concetto di scarsità, e far sì che i dati siano unici e non replicabili. [1]

1.2 Cenni storici

La nascita della blockchain è avvenuta nel 2008, quando Satoshi Nakamoto nell'articolo "Bitcoin: A Peer-to-Peer Electronic Cash System" [2], ha gettato le basi matematiche per la criptovaluta bitcoin. Nonostante la fama acquisita dal creatore, ancora oggi non è stata attribuita un'identità a esso e non si sa neanche se sia un uomo, una donna oppure un gruppo di persone. Tutta via ci sono delle ipotesi a riguardo, e a tal proposito sono finite sotto i riflettori quattro personaggi importanti, possibili identità di Satoshi: Dorian Nakamoto, Hal Finney, Craig Wright e Nick Szabo.



Figura 1.1: da sinistra a destra: Nick Szabo, Dorian Nakamoto, Craig Steven Wright, Hal Finney.

La tecnologia introdotta da Satoshi comunque, non è solo alla base di tutte le criptovalute, ma ha trovato ampia applicazione in altri settori, come quello sanitario, economico, agroindustriale. Ha anche aperto le porte a nuove applicazioni come gli smart contracts. Quando si pensa a Bitcoin si pensa anche alla blockchain, infatti sono soluzioni strettamente correlate. Per convenzione, quando si fa riferimento a Bitcoin con l'iniziale maiuscola si allude alla tecnologia e alla rete in generale, mentre, quando si fa riferimento alla valuta in sé, si usa la lettera minuscola. Si tratta comunque di un sistema di pagamento e di scambio valutario alla base della quale c'è una blockchain. Bitcoin ha introdotto un aspetto fondamentale in questo ambito: la decentralizzazione. Il fatto che ogni transazione non debba essere approvata da un ente centrale o da una terza parte infatti, rappresenta uno dei punti più importanti, che hanno contribuito all'evoluzione del sistema.

Il Bitcoin è nato quindi alla fine del 2008 con il rilascio del white paper (documento informativo per incentivare un progetto) nel quale venivano spiegate le funzionalità e gli intenti del progetto. Bitcoin viene concepito come una moneta virtuale peer-to-peer senza terze parti o intermediari. L'idea è quella di sovvertire gli ordini e le regole del mondo bancario, che in quel periodo presentava una particolare crisi. Nel 2009 la rete Bitcoin inizia a funzionare e la valuta assume un valore iniziale inferiore a un centesimo di dollaro. A utilizzare Bitcoin v0.1.0 (prima versione ufficiale del software) era un ristrettissimo gruppo di persone, forse solo due (Satoshi e Hal Finney) o poco più. In quell'anno il progetto era molto ristretto e non sembrava mostrare le potenzialità che vediamo oggi. Tuttavia iniziò a diffondersi all'interno della community dei cypherpunk, e ad essere utilizzato come forma di pagamento anonimo.

Nel giro di poco tempo il numero dei suoi utilizzatori aumentò in modo significativo, tanto che già l'anno successivo avvennero due passaggi fondamentali.

Il primo pagamento effettuato in bitcoin è avvenuto nel maggio del 2010, con l'acquisto di un bene materiale: sono state acquistate due pizze per la cifra di 10.000 BTC.

Alcuni mesi dopo avvenne il primo scambio di valuta su un exchange e il prezzo di Bitcoin si aggirava attorno agli 0,06 dollari. Il 2011 è stato un anno molto importante per la valuta, in quanto c'è stato un forte incremento del valore di questa, che ha toccato un picco massimo di 32 dollari. Rispetto al valore iniziale il prezzo in circa un anno era già cresciuto del 53%. È possibile che proprio questo primo enorme boom abbia fatto decollare l'interesse nei confronti di Bitcoin. [3]

Nel 2012 Bitcoin raggiunge la capitalizzazione di mercato di un miliardo di dollari. A partire dal 2014 l'attenzione, inizialmente tutta su Bitcoin, inizia a spostarsi verso la tecnologia sottostante. Iniziano a nascere piattaforme che sfruttano alcuni principi fondanti di Bitcoin: Ethereum, una piattaforma orientata alla creazione di smart contract che cerca anche di superare quelli che sono i limiti di velocità e flessibilità di Bitcoin; Ripple, una piattaforma nata nel 2012 per facilitare i pagamenti interbancari in diverse valute, che raccoglie l'adesione da parte delle prime banche.

Da questo punto in poi si inizia a fare riferimento a "Blockchain 2.0" per riferirsi a un nuovo modo d'impiego della blockchain. L'obiettivo di questo progetto era quello di fornire a coloro che sono al di fuori dell'attuale sistema monetario un modo affidabile e sicuro per immagazzinare fondi lontano da occhi indiscreti. Inoltre, gli utenti potrebbero monetizzare le proprie informazioni personali senza che nessuno se ne accorga. Questo sistema avrebbe ipoteticamente anche il potenziale per risolvere il problema della disuguaglianza sociale, cambiando il modo in cui la ricchezza è ridistribuita. Nel 2017 il parlamento del Nevada ha liberalizzato completamente la blockchain.

La sezione 4 del Senate Bill 398 stabilisce varie disposizioni relative all'uso della tecnologia blockchain. Ad esempio impedisce alle autorità locali di imporre tasse sulla blockchain, di richiedere una licenza per l'utilizzo della tecnologia o qualsiasi altra richiesta per il suo utilizzo. [4]

Nell'aprile del 2019, è stato presentato il primo manufatto artigianale Made in Italy nella mostra dell'artigianato. La particolarità è che le fasi produttive sono state monitorate tramite blockchain.

Nel 2016 la blockchain acquisisce tanta attenzione mediatica e diventa un fenomeno su larga scala, tanto che la stampa inizia a parlarne e viene considerata sempre di più nella sua individualità, così inizia a discostarsi da Bitcoin. Come conseguenza di ciò, inizia ad aumentare la consapevolezza delle aziende sulla Blockchain e partono numerosi progetti. Alla fine del 2017 si entra nella cosiddetta "fase di disillusiono dell'hype cycle".[5] Si comincia a dubitare dei limiti di questa tecnologia a causa della mancata realizzazione delle promesse fatte. Le criptovalute si trasformano sempre maggiormente in uno strumento di speculazione finanziaria e continuano a oscillare nel prezzo, presentando una volatilità molto alta. Il mining all'interno della rete di Bitcoin (il processo per la validazione dei blocchi) presenta alcuni limiti al crescere della rete, come la lentezza nell'effettuare una transazione o il consumo elevato

di energia. Nel 2018 è avvenuto un netto calo in termini di capitalizzazione, tanto che la community conia un nuovo termine per definire questo momento: “crypto winter”. [6]

1.3 Utilizzi reali

La blockchain è una struttura dati nota per tenere traccia degli scambi di denaro digitale effettuati con Bitcoin o con altre criptovalute. Questa non è, però, l'unica applicazione possibile, in quanto la tecnologia può essere sfruttata in diversi settori, come quello alimentare, quello sanitario o quello immobiliare. All'interno dei blocchi non per forza devono essere contenuti dati, possono anche essere presenti script ad esempio. Spesso si tratta di codice semplice che serve a specificare il destinatario, la quantità di token da trasferire e a provare l'identità del mittente. Per questo scopo si usano linguaggi orientati ai contratti come Solidity o Liquidity. Dal lancio di Bitcoin sono nate oltre 5000 criptovalute, tra le quali si è distinta Ether, nata dal progetto di Ethereum sviluppato nel 2015 e pensato per l'utilizzo degli smart contract e per oltrepassare i limiti di Bitcoin.

La diffusione della blockchain come struttura alla base di diverse tecnologie ha portato all'emanazione di varie leggi, anche per la regolamentazione della stessa. In Italia il 23 gennaio 2019 è stato approvato un emendamento al Senato che include formalmente per la prima volta nel nostro ordinamento la tecnologia della blockchain e gli smart contract. In particolare, vengono definite le "tecnologie basate su registri distribuiti e smart contract" nella seguente maniera:

- Si definiscono "tecnologie basate su registri distribuiti" le tecnologie formate da un registro distribuito, replicabile e consultabile istantaneamente. Questo registro, basato su crittografia, deve mantenere la decentralizzazione e garantire la sicurezza per quanto riguarda registrazione, convalida e aggiornamento dei dati.
- Si definisce "smart contract" un programma che opera su tali tecnologie e la cui esecuzione vincola automaticamente due o più parti nel rispetto di alcune condizioni. Gli smart contract soddisfano il requisito della forma scritta previa identificazione informatica delle parti interessate, attraverso un processo avente i requisiti fissati dall'Agenzia per l'Italia Digitale con linee guida da adottarsi entro 90 giorni dall'entrata in vigore della legge di conversione del decreto legge. [7]

1.4 Ledger

Con il termine Distributed Ledger Technologies (DLT) si fa riferimento a "libri mastri", registri elettronici distribuiti geograficamente su un'ampia rete di nodi. I dati risiedenti all'interno di questi registri sono protetti da attacchi informatici siccome vengono validati in accordo a vari protocolli accettati da ciascun partecipante. La gestione di tali registri è decentralizzata e l'archiviazione di informazioni criptate è basata su algoritmi di consenso che coinvolgono tutti o parte dei partecipanti. Si tratta di meccanismi che permettono ai nodi di concordare sull'insieme di transizioni valide.

I libri mastri esistono sin dall'antichità, usati per testimoniare le transazioni economiche, con il compito di registrare i pagamenti, i contratti tra più parti oppure per spostare beni o proprietà. Nell'ultimo periodo, i computer hanno fornito il processo di gestione e manutenzione dei libri mastri con grande comodità e velocità. Oggi, le informazioni archiviate sui computer stanno diventando crittograficamente protette, veloci e decentralizzate.

La blockchain in particolare può essere vista come evoluzione di centralized e decentralized Ledger, e si configura come distributed Ledger.

Centralized Ledger

La struttura centralizzata permette di far riferimento e affidamento a un organo centrale, che funge da gestore e organizzatore della struttura stessa (figura 1.2 a). La fiducia deve essere riposta nell'autorità, nell'autorevolezza del soggetto o sistema che rappresenta il "centro" dell'organizzazione. Queste architetture presentano problemi di affidabilità in quanto dispongono di un solo server centrale che funge da moderatore dell'intero sistema. Questo rappresenta un "single point of failure" in quanto, in caso di problemi con il server centrale, c'è la possibilità che tutto il resto del sistema smetta di funzionare.

Decentralized Ledger

Se la struttura centralizzata propone una logica "uno a molti", quella decentralizzata si discosta un po' da questa filosofia, riproponendola però, a livello locale. In questo caso non ci sarà una singola entità a possedere e diffondere l'informazione ma ce ne saranno diverse. Come si può vedere dalla figura 1.2 b, sono presenti varie unità centrali che gestiscono diversi nodi. Si avranno tanti soggetti che fungeranno da server, in modo da aggirare il problema del single point of failure in quanto un guasto a uno dei soggetti centrali della rete può essere mitigato dalla presenza delle altre entità con ruolo analogo.

Distributed Ledger

I Distributed Ledger sono database decentralizzati e condivisi che possono essere distribuiti o replicati sui nodi di una rete peer-to-peer. In quanto database, è presente un meccanismo d'inserimento e salvataggio dei dati, che sono salvati in maniera continua in accordo a criteri di consenso, e l'aggiunta di voci al registro è protetta da meccanismi come proof-of-work o proof-of-stake.[8]

Il registro distribuito si propone come evoluzione dei precedenti in quanto supera i limiti della centralizzazione. I libri mastri centralizzati sono più inclini agli attacchi informatici, mentre quelli distribuiti sono intrinsecamente più difficili da attaccare perché tutte le copie distribuite devono essere attaccate contemporaneamente affinché un attacco abbia successo. Essendo difficili da manipolare e attaccare, i libri mastri distribuiti consentono un'ampia trasparenza.

Inoltre, riducono le inefficienze operative, accelerano la quantità di tempo necessaria per il completamento di una transazione e sono automatizzati (funzionano perennemente), il che riduce i costi complessivi per le entità che li utilizzano.

Uno dei concetti chiave per quanto riguarda i sistemi distribuiti è infatti la resistenza ai malfunzionamenti; ogni componente di un sistema distribuito può smettere di funzionare indipendentemente dalle altre, senza causare necessariamente il crollo dell'intero sistema o di altre parti di esso.[9] Questi sistemi eliminano in via definitiva la centralizzazione in modo tale che ogni nodo della rete sia connesso con gli altri, senza dipendere da alcun entità (figura 1.2 c).

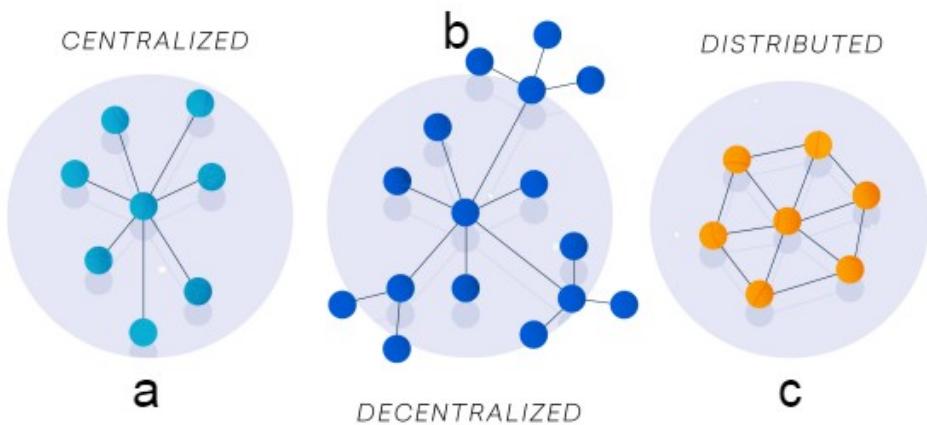


Figura 1.2: Differenze tra logica centralizzata, decentralizzata e distribuita

1.5 Distributed Ledger Technologies

Ci sono varie modalità per gestire il consenso distribuito, che cambiano a seconda di vari parametri, come ad esempio le politiche di lettura e scrittura delle transazioni, o la convalida di queste all'interno del registro. Ci sono infatti diverse prassi di accesso e scrittura del registro distribuito che determinano la suddivisione tra DLT. I diversi registri distribuiti competono per primeggiare, e le loro implementazioni variano base alla struttura dati utilizzata, alla tolleranza ai guasti e al consenso. Queste differenze influenzano ogni DLT in termini di costi, latenza, sicurezza e prestazioni. Una delle principali differenze è il tipo di struttura dati utilizzata e i dati memorizzati. In figura 1.3 sono analizzate le strutture della blockchain, del tangle, dell'hashgraph e della sidechain. La Blockchain si basa su una struttura di dati a elenchi collegati in cui ogni elemento dell'elenco è un blocco di transazioni. La sidechain estende questa struttura ponendo in parallelo più elenchi tra di loro collegati. D'altra parte, tangle e hashgraph utilizzano un DAG (directed acyclic graph) come struttura dati sottostante. In tangle, ogni nodo del DAG è una transazione, mentre in hashgraph un nodo rappresenta un evento, che può contenere più transazioni al suo interno. Esistono quindi differenti tipi di DLT che ti contraddistinguono per varie caratteristiche, ma noi ci concentreremo su soluzioni permissioned e permissionless.

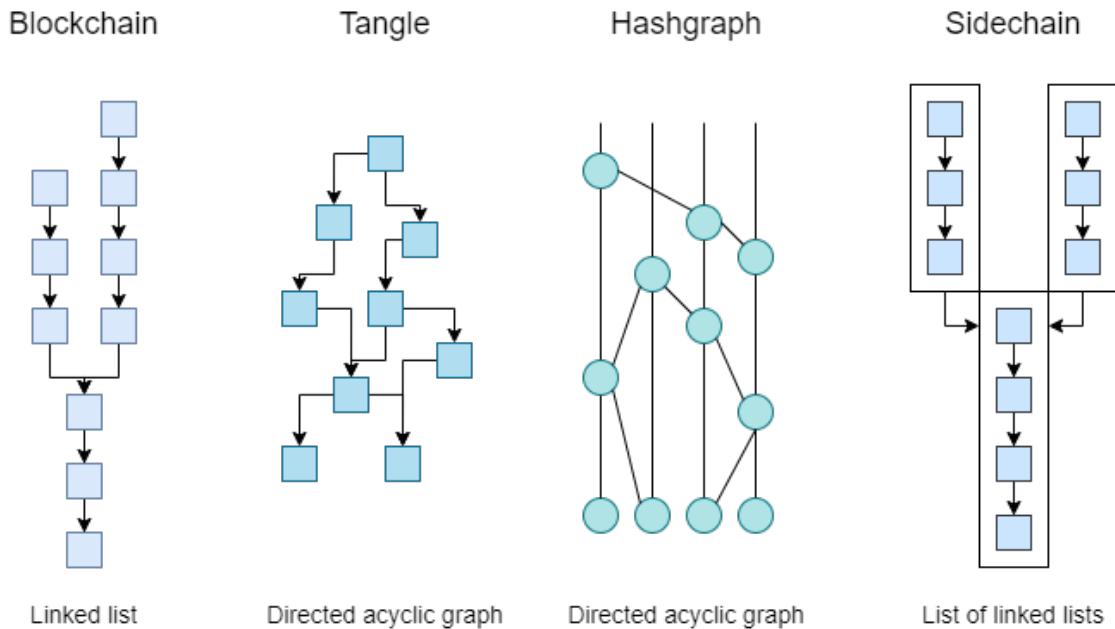


Figura 1.3: Diversi tipi di DLT a confronto

1.5.1 DLT Permissionless

L'esempio per eccellenza quando si parla di DLT permissionless è Bitcoin. Questa rete infatti, è aperta a tutti, non ha una “proprietà” o un attore di riferimento ed è concepita per non essere controllata. L'obiettivo è proprio quello: permettere a ciascun utente del sistema di contribuire all'aggiornamento dei dati sul Ledger e di disporre, in qualità di partecipante, di tutte le copie immutabili di tutte le operazioni. In una blockchain pubblica chiunque può avere i permessi di modifica, lettura e scrittura sul registro, e quindi, chiunque può entrare a far parte della rete. Nonostante il gran numero di nodi presenti e la possibilità data a qualsiasi soggetto di mettere le mani sui dati, una soluzione del genere garantisce comunque un elevatissimo livello di sicurezza, dato in gran parte dalla crittografia a chiave pubblico-privata. Una volta conquistato il consenso per la convalida di un'operazione, non c'è modo che questa possa essere cambiata, nessuno è nella condizione di impedire che le informazioni vengano aggiunte al ledger. Una rete del genere potrebbe essere molto utile, ad esempio, per trattare dati che hanno la necessità di rimanere immutabili nel tempo e che richiedono la massima sicurezza, come i contratti di proprietà oppure i testamenti.

1.5.2 DLT Permissioned

Le Permissioned Ledger invece non rispecchiano questa libertà e apertura, sono soluzioni che mirano a restringere il numero possibile di partecipanti, controllandone le politiche di accesso. Le autorizzazioni di scrittura e modifica dei blocchi sono centralizzate mentre quelle di lettura sono pubbliche o limitate a un gruppo specifico di utenti. Il sistema rimane protetto da verifiche crittografiche rinunciando però alla decentralizzazione, rendendo così la blockchain

una soluzione "privata". Il sistema di approvazione dei record non è affidato, come nel caso delle soluzioni permissionless, alla maggioranza dei nodi, ma a un numero ristretto di essi, identificati come affidabili. Questo tipo di soluzione trova ampio utilizzo nel campo delle istituzioni, di grandi imprese che devono gestire filiere con una serie di attori, di banche, società di servizi. In questo caso le Permissioned Ledger rispondono alle necessità di un aggiornamento diffuso su più attori che possono operare in modo indipendente, ma con un controllo limitato a coloro che sono autorizzati. Le blockchain private, in quanto tali, devono poter far affidamento su reti protette e largamente testate per limitare i possibili attacchi. Infatti l'impenetrabilità della rete ne garantisce in larga parte la sicurezza. Poiché il proprietario della rete ha l'autorità di modificare le regole di base della Blockchain, può bloccare determinate transazioni in base a regolamenti stabiliti. Il fatto che solo gli utenti invitati e autorizzati possano accedere alla rete garantisce una maggiore privacy per gli utenti e garantisce la riservatezza dei dati memorizzati su di essa.

1.5.3 Differenze

Mentre le blockchain pubbliche non sono soggette ad alcuna "autorità centrale" e quindi ogni nodo nella rete ha potenzialmente lo stesso potere degli altri, le blockchain private lo sono, creando così una rete chiusa alla quale possono partecipare solo nodi autorizzati. Variano quindi le autorizzazioni di lettura, scrittura e modifica da parte dei singoli nodi.

Entrambi i tipi di blockchain presentano punti a favore e sfavore, ma in generale si ritiene più all'avanguardia la soluzione permissionless che, nonostante l'assenza di controllo, permette comunque un alto grado di sicurezza e affidabilità, garantendo anche la distribuzione su tutti i nodi. Le blockchain pubbliche quindi rispecchiano il concetto di decentralizzazione che sta alla base delle blockchain stesse, quelle private sacrificano questa caratteristica in cambio di spazio, velocità e riduzione dei costi. Nel primo caso avremo una maggiore sicurezza a scapito della scalabilità e, a volte, della privacy; nel secondo caso avremo invece una minore sicurezza che va ad appannaggio di una migliore scalabilità e garanzia della privacy. Anche se sembra strano pensare che una soluzione priva di controllo da parte di un'autorità possa essere più sicura di una soluzione gestita da un'organizzazione centrale che governa gli accessi, c'è da tener conto che è proprio la pluralità il punto di forza della soluzione permissionless. In un contesto in cui chiunque può partecipare al processo di convalida, dove non esistono criteri preselettivi, l'assenza di centralizzazione e la distribuzione delle informazioni su tutti i nodi fanno sì che il sistema si discosti da possibili attacchi alla figura centrale.

In contesti architetturali di tipo permissioned, invece, nei quali solo alcuni soggetti preselezionati sono incaricati di convalidare i dati scritti sul registro distribuito, l'unica garanzia che si può avere è correlata all'interesse collettivo della piattaforma.

Mentre alle prime chiunque può prenderne parte (si pensi alla blockchain Bitcoin ove non vi sono restrizioni o condizioni di accesso), per le altre non è così; vengono utilizzate maggiormente per società private, istituzioni finanziarie, imprese e banche.

1.6 Meccanismi di consenso

La parola “consenso” si riferisce a una convergenza verso un interesse comune. Per consenso si intende l’interesse comune che i sistemi devono avere per il raggiungimento di un obiettivo. Gli agenti devono raggiungere un accordo su un certo interesse (un valore o un’azione per esempio) a seconda del loro stato. Per mantenere i nodi malevoli lontani da possibili attacchi alla rete, l’idea è quella di introdurre costi computazionali per addebitare i peer che deviano dal comportamento predefinito. Con l’aumento della popolarità delle criptovalute, i requisiti di scalabilità e prestazioni sono cambiati in modo significativo. Sono emersi quindi i limiti delle blockchain di prima generazione ed hanno portato a un’analisi più approfondita. Il meccanismo di consenso Proof of work ad esempio presenta limiti per quanto riguarda scalabilità e spreco di risorse, anche se appropriate modifiche alla procedura possono garantire livelli di scalabilità impegnativi con minore spreco di energia. Il consenso PoW insieme ai protocolli caratterizzati da un’elezione del leader basata sullo sforzo formano la classe degli algoritmi di consenso proof-of-X (PoX)[10]

I protocolli PoX si affidano a un processo probabilistico di elezione del leader. In ambienti senza autorizzazione ogni nodo ha la possibilità di diventare un leader semplicemente dimostrando di aver fatto un lavoro, che può essere di natura computazionale, monetaria o di archiviazione oppure può essere uno sforzo per affermarsi sulla rete blockchain.

Proof of work

Il meccanismo di consenso Proof of Work è quello utilizzato dalla blockchain di Bitcoin. Si chiama così perché richiede un certo tipo di sforzo computazionale da parte dei nodi (chiamati miner) che compongono la rete. I miner hanno il compito di convalidare le transazioni Bitcoin eseguendo il software su computer solitamente performanti. Questi risolvono un difficile problema matematico per verificare una transazione e guadagnare denaro. Svolgono questo compito monitorando e contribuendo alla rete Bitcoin, in un processo chiamato "mining". Per il raggiungimento del consenso quindi, è richiesta una "difficoltà", che viene usata come parametro da rispettare affinché un blocco sia considerato valido. Questo parametro può essere modificato dalla rete per far sì che la velocità con la quale i blocchi vengono aggiunti rimanga costante.

Questo meccanismo presenta però degli svantaggi, infatti risulta essere lento in termini di velocità di elaborazione delle transazioni, con un numero di sette transazioni al secondo. Inoltre, sono richieste grandi quantità di energia per il processo di mining. Per esempio, il mining di solo un bitcoin ha avuto un costo medio di più di 10.000 euro nel 2018. Per questo motivo, i miner cercano di migliorare le loro possibilità unendosi in gruppi chiamati "pool". In alcuni casi sono sfruttati i servizi di cloud, che permettono anche all’utente medio di partecipare a un processo così oneroso in termini di risorse.

Un altro svantaggio del processo è che i mining pool grandi hanno più potenza di calcolo a loro disposizione, e quindi maggiori possibilità di effettuare il mining di blocchi validi, mettendo i miner più piccoli in svantaggio. Esistono protocolli con il compito di migliorare velocità e scalabilità di Bitcoin, rimediando ai limiti posti da questa rete, come Lightning

Network. Questo è un protocollo posizionato al di sopra della rete Bitcoin, progettato per togliere la pressione di un gran numero di transazioni dalla blockchain centrale di Bitcoin, in modo tale da alleggerire il carico.

Anche se altri meccanismi e algoritmi possono essere superiori alla Proof of Work in quanto velocità di esecuzione e spreco di risorse, quest'ultimo rimane quello più affermato e testato nel tempo contro gli attacchi. Quindi è molto probabile che l'algoritmo PoW venga continuamente migliorato dagli sviluppatori per far fronte alle sue carenze. Per porre rimedio al consumo di energia ad esempio, si è discusso molto sulla fonte di energia utilizzata, e si è pensato di introdurre energia rinnovabile per alimentare la rete. La crescente necessità di potenza di calcolo inoltre, genera un pesante problema ambientale, non solo dovuto al consumo di energia, ma anche alla produzione di rifiuti elettronici. Ci sono ad esempio dispositivi creati e utilizzati appositamente per il processo di mining, che quindi non possono essere riutilizzati e rimangono un problema per l'ambiente. [11]

Proof of Stake

Il consenso Proof of Stake ha lo stesso obiettivo dell'algoritmo Proof of Work, ma, a differenza di questo, non ci sono miner coinvolti nel processo. Al posto dei miner, che devono svolgere un complicato sforzo computazionale, i partecipanti in questo caso devono detenere una certa quota di token. Per essere coinvolti nel processo di verifica delle transazioni infatti, devono mettere una quantità di moneta "in gioco", in un portafoglio collegato alla blockchain. Questo processo è noto come "staking". Un validatore in questo caso può creare blocchi in maniera proporzionale al denaro messo in staking, e anche l'elezione per la validazione si basa sulla natura della posta in gioco. Questo meccanismo però genera un problema: una possibile centralizzazione dovuta al possesso di una grande quantità di denaro da parte di un singolo nodo o da un gruppo ristretto di essi. I produttori di blocchi di alcune monete possono esercitare una grande quantità di potere se il numero di produttori di blocchi è basso, e se quindi sono loro a poter convalidare tutte le transazioni. Tuttavia, c'è la possibilità (come per la moneta EOS) che un produttore venga revocato se agisce contro gli interessi della rete. Infatti in EOS, se un produttore non riesce a convalidare blocchi per più di 24 ore, viene sostituito da uno di riserva. Con la Pos però, è richiesta meno potenza di calcolo della PoW e quindi ha meno impatto sull'ambiente. L'alternativa Pos presenta un altro problema noto come "nessuna posta in gioco"[12]. Nel consenso Pow la generazione di due blocchi in maniera quasi simultanea (a caso di un ritardo per esempio), porta confusione sul decidere quale blocco debba essere validato. Nella Pos, siccome non ci sono risorse di calcolo in gioco, si può procedere con la validazione di entrambe le parti generando così una fork, con possibili blocchi in conflitto tra di loro. Questo rallenta il consenso nella rete e indebolisce il sistema, rendendo più facili gli attacchi di vario tipo. In un confronto tra proof of work e proof of stake (figura 1.4) si può notare come quest'ultimo sia meno dispendioso, non siano necessarie risorse onerose per la validazione dei blocchi e come gli utenti abbiano il pieno controllo dei propri token.

	Proof of work	Proof of stake
Energy consumption	High	Low
Required tools	Mining equipment	No equipment
Security	High	Untested
Decentralized vs Centralized	Tends to centralized	Users can remain in control of their tokens

Figura 1.4: Differenze tra i protocolli di consenso proof of work e proof of stake

Delegated Proof of Stake

Il consenso Delegated Proof of Stake (DPos) nasce come variante del consenso Proof of Stake. Si basa sul fatto che i partecipanti della rete scelgono dei delegati, che vengono eletti per verificare le transizioni e generare i blocchi. Questo sistema permette una rapida e precisa individuazione dei problemi e la conseguente sostituzione di delegati (anche detti testimoni) che corrompono il consenso. Quando questi però ricevono una ricompensa per la validazione delle transizioni, solitamente la dividono con i portafogli che hanno contribuito a eleggerli.

Proof of Authority

Lo schema di consenso Proof of Authority (PoA) è progettato per essere una soluzione pratica ed efficiente, rivolta soprattutto alle blockchain private. Questo protocollo funziona in maniera differente rispetto ai precedenti. Infatti PoA fa uso delle identità reali dei partecipanti per ottenere la convalida. Quindi chi ha il compito di validare i blocchi mette in gioco la propria reputazione come garanzia di onestà. Questo meccanismo, si basa su un numero limitato di validatori, cosa che offre un chiaro vantaggio: l'elevata scalabilità della blockchain. Inizialmente, i validatori vengono scelti in maniera casuale, e l'inclusione di nodi avviene attraverso un sistema di votazione. Sono i validatori precedentemente autorizzati a scegliere quali nodi selezionare, in modo tale da limitare l'inclusione di nodi malevoli. Ogni validatore poi, può firmare al massimo uno di una serie di blocchi consecutivi durante il proprio turno di convalida. L'identità è quindi al centro di questo meccanismo di consenso e risulta essere una cosa molto preziosa. Ogni validatore deve condividere e confermare la propria, e non può permettersi di compiere atti volti alla minaccia della rete, perché questo potrebbe intaccare o distruggere la reputazione acquisita. L'identità, che viene messa in gioco, può fungere da grande equalizzatore per preservare la rete.

1.7 I blocchi

I blocchi sono i componenti principali di una blockchain, e la correlazione tra di essi va appunto a formare la catena. Nel pacchetto dati di ogni blocco è contenuto l'hash del blocco precedente, valore che permette di collegarli e verificarne l'autenticità, come visibile in figura 1.5. Siccome l'hash cambia anche con una piccola variazione dell'input, una volta creato un blocco con relativo hash, non può più essere modificato. I blocchi sono costituiti comunque da transazioni, elementi fondamentali nel mondo delle criptovalute per lo scambio di valore. Perché un nuovo blocco di transazioni sia aggiunto alla blockchain è necessario appunto che sia controllato, validato e crittografato. Per ciò è indispensabile svolgere un complicato lavoro computazionale dovuto alla risoluzione di un complesso problema matematico (mining). Il tipo di remunerazione che si ottiene dal processo di mining varia a seconda delle regole e del tipo di governance definite dalla blockchain. In questa è contenuta tutta la storia delle transazioni a partire dalla genesi, ovvero dalla creazione del primo blocco. Oltre alle transazioni e all'hash, che permette di concatenare un blocco a quello precedente, ogni blocco contiene un timestamp (marca temporale) e un nonce (number only used once), ossia un numero che un miner deve scoprire per poter risolvere un blocco nella blockchain. L'insieme di questi fattori permette di garantire l'integrità della blockchain fino al primo blocco. Abbiamo visto che il consenso consiste nell'accordo della maggioranza dei nodi della rete sulla validità delle transazioni in un blocco, che una volta validato viene aggiunto permanentemente alla catena senza possibilità di modifica, grazie al fatto che cambiando anche un solo valore cambierebbe tutto l'hash (e non sarebbe più consistente con il valore precedente). Il protocollo alla base di Bitcoin limita la grandezza dei blocchi a 1 MB, e ogni blocco contiene all'incirca 4000 transazioni. I blocchi vengono aggiunti in media ogni 10 minuti e questi numeri sono molto bassi, specialmente se comparati con quelli offerti da sistemi già affermati come il circuito VISA. Sono state pensate varie soluzioni per porre rimedio al problema della lentezza, ad esempio aumentare la grandezza dei blocchi. Si è pensato di aumentare la dimensione di ogni blocco a 2 MB in seguito al raggiungimento dell'accordo dei nodi della rete (oltre al 95%), oppure di aumentare le dimensioni del 4,4% circa ogni 97 giorni. Kyle Croman [13] ha analizzato il throughput effettivo della rete (numero di blocchi che si propagano nel tempo), giungendo alla conclusione che se il tasso di transazione supera il throughput del 90%, il 10% della rete non è in grado di rimanere al passo. Questo potrebbe portare a inefficienza e alla possibilità di negare i servizi agli utenti e ridurre la potenza di mining. Sono consigliati due parametri in tal caso per garantire che almeno il 90% dei nodi abbia un throughput sufficiente: la dimensione dei blocchi non dovrebbe superare i 4 MB e l'intervallo di propagazione del blocco non dovrebbe essere inferiore a 12 secondi, così da garantire il massimo utilizzo della banda di rete.

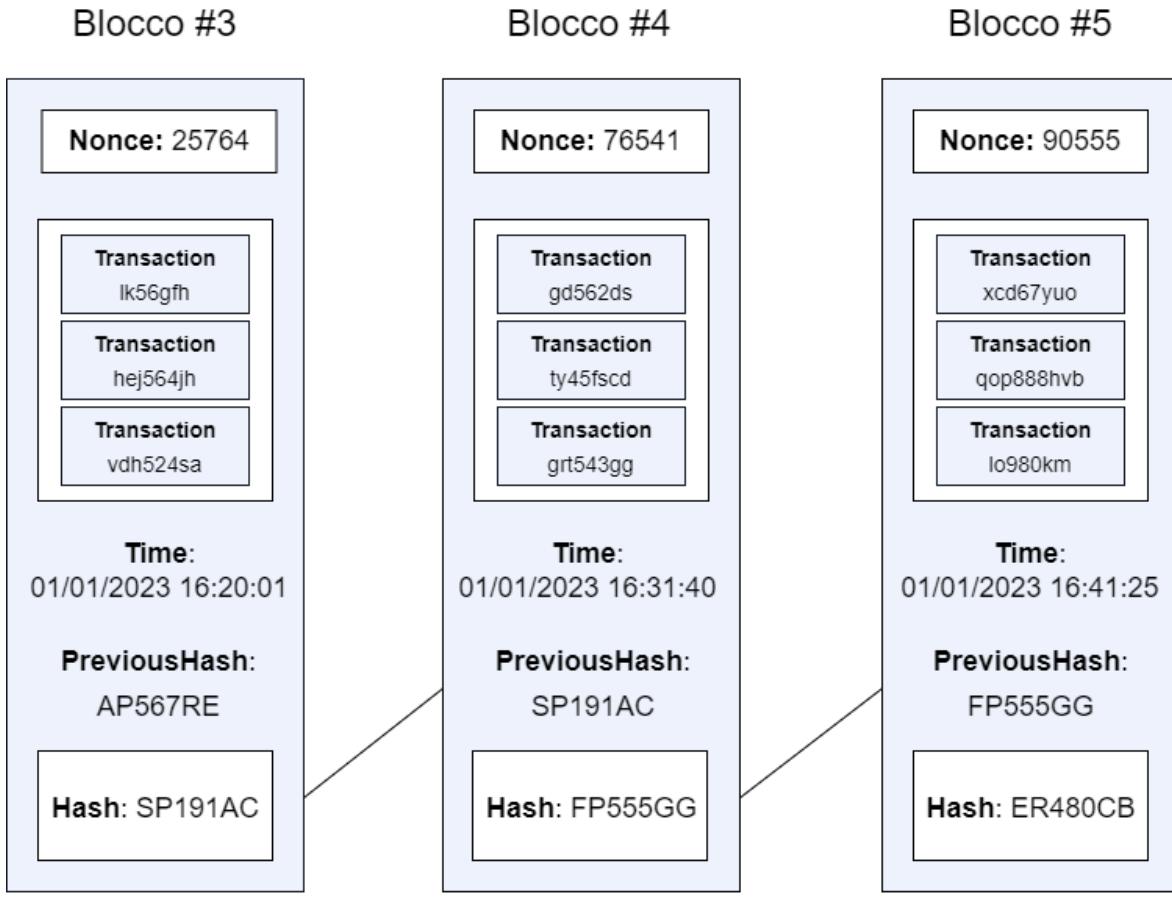


Figura 1.5: Struttura dei blocchi all'interno della blockchain e collegamento tra di essi

1.8 Le transazioni

Le transazioni sono un mezzo per garantire lo scambio di asset tra parti interessate. I componenti che le caratterizzano sono:

- Sender (persona che avvia il processo di transazione);
- Transaction (transazione come processo);
- Receiver (persona che ottiene la transazione);

Si inizia con la creazione di una transazione, che subito dopo viene firmata digitalmente con una o più firme, al fine di dimostrare l'autorizzazione a spendere una certa quantità di criptasset in accordo alla transazione stessa. Viene quindi inviata alla rete e verificata dai nodi che la propagheranno a tutti gli altri; infine, viene validata da un nodo miner e aggiunta a un blocco registrato sulla blockchain.[1]

Per avviare una transazione è fondamentale l'utilizzo di chiavi crittografiche, ossia un insieme variabile di dati inseriti in input ad un algoritmo di crittografia adatto a questa mansione. Le chiavi crittografiche possono essere classificate come simmetriche oppure asimmetriche, in base al loro utilizzo. Una chiave simmetrica è una singola chiave usata per entrambe le

operazioni (criptare, decrittare) in uno schema di crittografia. In generale si fa affidamento sul fatto che la chiave privata sia conosciuta solo da una delle due parti, cosa che rende lo schema pressoché sicuro. Le chiavi asimmetriche, d'altra parte, vengono usate negli schemi che richiedono una maggiore sicurezza e affidabilità, in cui sono necessarie chiavi diverse per ogni operazione. Esempi di ciò sono gli schemi che usano coppie di chiavi pubbliche/private, in cui la sicurezza dello schema dipende dalla garanzia che la chiave privata sia nota solo ad una delle parti. Viene quindi usata la chiave pubblica per cifrare i dati, siccome chiunque ha accesso a essa, e la chiave privata per decifrare il messaggio. Il mittente, per inviare i dati al destinatario, li cifra con la chiave pubblica di quest'ultimo. Sarà poi compito del destinatario, una volta ricevuto il messaggio, decifrare il messaggio utilizzando la sua chiave privata.

Ogni utente, quando entra a far parte di una blockchain, crea un proprio portafoglio digitale (wallet). Una volta creato questo wallet, viene fornito delle due chiavi crittografiche. Per ogni coppia di chiavi viene inoltre generato matematicamente, dal sistema, anche un indirizzo pubblico che spesso è associato ad un QR Code, per rendere più agevoli le operazioni.

Cosa avviene nello specifico? Per inviare dati o informazioni, il mittente, dopo aver acceduto al proprio portafoglio, deve essere a conoscenza della propria chiave privata, e di quella pubblica del destinatario. In realtà si tratta dell' l'indirizzo derivato dalla chiave pubblica del ricevente e non della chiave stessa. A questo punto il ricevente deve aver aperto già un proprio wallet e deve disporre del codice da inviare al mittente, ovvero un codice alfanumerico oppure un codice QR. Successivamente si può procedere alla creazione della transazione, specificando l'importo desiderato. Insieme a ciò può essere presente del codice contenuto all'interno di uno script, che specifica altre informazioni. Sono presenti in realtà due script: il primo si chiama Unlocking Script (o anche scriptSig) ed è quello di sblocco, che determina il modo in cui il destinatario dovrà confermare di essere realmente lui. In questo script sono contenute la sua firma digitale e la sua chiave pubblica. Il secondo script, chiamato Locking Script (o scriptPubKey) è quello di blocco, che determina le condizioni di spendibilità di quei bitcoin inclusi nella transizione creata dal mittente. Bisogna anche tener conto del fatto che tutto lo storico delle transazioni è salvato nella blockchain, quindi quando un utente desidera inviare del denaro deve dimostrare la capacità di spendita, ovvero dimostrare che è in reale possesso di quella quantità.[1]

Quando un nodo richiede la verifica di una transazione, questa viene propagata sulla rete, che ne legittima la validità in modo da garantire l'inserimento all'interno di un blocco in attesa di validazione. Tale processo, cui si attribuisce il nome di verifica indipendente, precede quello di validazione effettiva che si avrà con la validazione del blocco ed è eseguito da ciascun singolo nodo.

1.9 Sidechain o second level chain

La sicurezza e la gestione della governance nelle blockchain sono temi sempre più popolari e discussi. Le blockchain pubbliche traggono forza da una serie di punti che vanno consolidando sempre maggiormente nel tempo. Quelle private, pur rinunciando ad alcune caratteristiche peculiari delle blockchain, rispondono molto bene ad alcuni requisiti. La necessità di trovare un compromesso tra queste due realtà ha portato alla creazione di un nuovo modello: quello delle sidechain o second level chain.

Una soluzione che permette di creare un nuovo livello, diverso da quello della catena principale, in grado di aiutare il sistema a essere più veloce ed efficiente fungendo da livello di scambio. Questo al di fuori della blockchain principale.

Grazie alle Sidechain si crea questo meccanismo di scambio per la gestione delle transazioni che, siccome avviene "off-chain", permette di togliere carico dalla catena principale. Gli asset elaborati sulla chain secondaria possono poi essere riportati su quella principale.

Il compito delle second level chain è quindi quello di gestire gli scambi tra soggetti attivi sulla blockchain su spazi secondari ed esterni. Si possono così sfruttare i vantaggi delle blockchain permissionless senza rinunciare ai pregi di quelle permissioned, che possono essere utilizzate anche solo per specifici punti o funzioni. Queste chain, affiancate a quelle maggiori, sono in grado di garantire l'interscambiabilità degli asset con un compromesso tra apertura, accessibilità, efficienza e controllo degli accessi. La chain secondaria può avere dei protocolli e meccanismi di consenso diversi da quella primaria ed è collegata a essa tramite un peg bidirezionale (figura 1.6). Secondo un articolo pubblicato sul "Journal of Network and Computer Applications", [14] esistono principalmente tre scelte progettuali per l'implementazione del peg bidirezionale per il trasferimento di asset dalla mainchain alla sidechain e viceversa. La prima soluzione, centralizzata, prevede un'entità fidata che si occupa di trattenere i fondi (e di gestirne l'eventuale blocco o sblocco); la seconda soluzione multi firma rappresenta un miglioramento rispetto alla soluzione centralizzata e prevede il controllo da parte non di una singola entità ma di un intero gruppo (i fondi vengono sbloccati solo in accordo alla maggioranza delle entità); l'ultima soluzione basata sulla verifica leggera dei pagamenti (SPV) permette a un client di provare che la transazione è stata inserita in un blocco legittimo della chain più lunga, senza il bisogno di scaricare tutta la catena a partire dal blocco genesi. Questi lightweight client sono tenuti a scaricare solo le intestazioni dei blocchi dell'intera blockchain, che hanno dimensioni molto più ridotte rispetto al blocco stesso, rendendo questa soluzione performante.

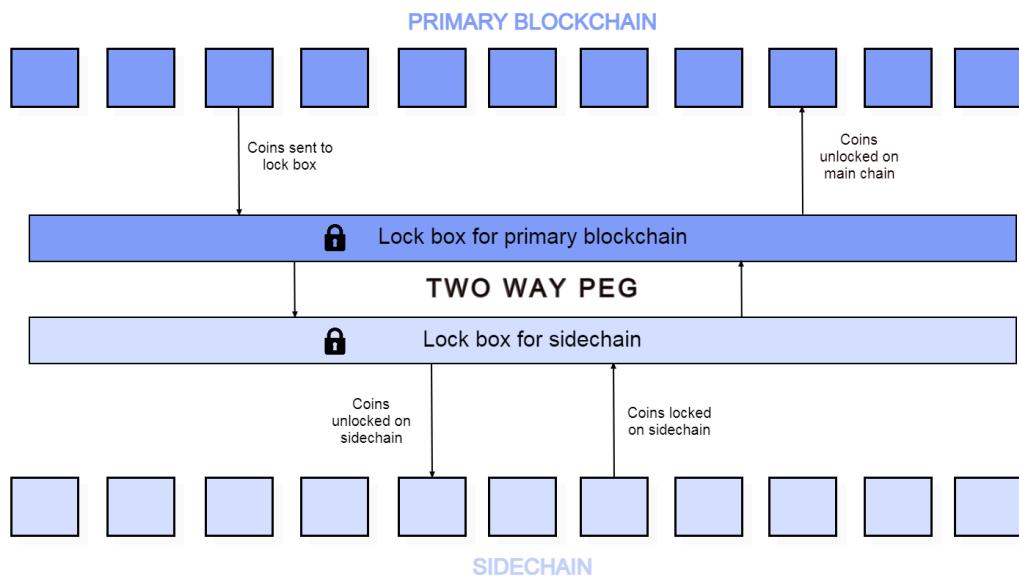


Figura 1.6: Collegamento tra side chain e main chain attraverso un peg bidirezionale

1.10 Sicurezza

Una delle caratteristiche più importanti della blockchain è la sicurezza, che viene garantita da alcuni meccanismi come timestamping e tecniche crittografiche. La crittografia, dal greco kryptos, nascosto, e graphein, scrivere, è la scienza che si occupa dello studio delle scritture segrete. Con la digitalizzazione dei dati c'è stata sempre una maggiore necessità di proteggere le informazioni sensibili, quali ad esempio carte di credito, coordinate bancarie eccetera. Anche nelle blockchain risulta di notevole rilievo la crittografia, che si serve di funzioni dette funzioni hash.

1.10.1 Funzioni Hash e timestamping

La parola Hash deriva dall'inglese “to hash”, sminuzzare. Si tratta di funzioni one way che consentono di mappare dei dati di lunghezza arbitraria in una stringa di lunghezza fissa chiamata hash o digest. Questo digest è strettamente legato al messaggio in entrata poiché ogni messaggio genera un digest univoco.[15]

Queste funzioni si rivelano utili in applicazioni come la firma digitale perché consentono di provare l'autenticità del messaggio. Il ricevente di un messaggio, per assicurarsi della sua autenticità, calcola la funzione di hash sui dati ricevuti e confronta l'output con il digest fornитогli. Se i due valori non corrispondono allora significa che i dati sono stati manomessi o comunque non sono arrivati integri.

Le funzioni hash sono però funzioni non iniettive: a messaggi diversi potrebbe teoricamente corrispondere lo stesso valore di hash. In questo caso si parla di collisioni; lo scopo principale è quello di minimizzare le collisioni rendendo la funzione hash più efficiente e permettendo così un'affidabilità maggiore.

Per soddisfare i requisiti di sicurezza richiesti è essenziale che vengano rispettate le seguenti proprietà:

- Determinismo: ogni funzione di hash applicata allo stesso input deve restituire sempre lo stesso output.
- Non invertibilità: il digest non deve fornire alcuna informazione sul messaggio originale e non deve essere possibile recuperarlo.
- Effetto cascata: la variazione di un singolo carattere nell'input deve produrre in output una stringa totalmente differente.
- Resistenza alle collisioni: trovare due digest uguali a partire da stringhe diverse deve essere estremamente improbabile.
- Veloce da calcolare: il calcolo del digest a partire da qualsiasi input deve essere un'operazione efficiente e veloce.

La maggior parte delle blockchain usa la funzione di hash Secure Hash Algorithm 256 (SHA-256), che restituisce un digest di 32 byte. Il funzionamento di SHA-256 può essere descritto nella seguente maniera:

- Vengono aggiunti dei bit all'input in maniera tale da ottenere un messaggio che risulta congruo a 448 modulo 521.
- Si aggiunge un unsigned int di 64 bit che rappresenta la lunghezza del messaggio iniziale.
- Vengono creati alcuni buffer e vengono inizializzati con valori esadecimali.
- La sequenza di bit risultante viene divisa in blocchi da 512 bit.
- Avviene la fase di compressione, che è suddivisa in cicli. In ogni ciclo, l'algoritmo prende in input i valori dei blocchi e i buffer insieme a una costante ed al termine dell'ultima iterazione il valore dei registri sarà l'output della funzione di hash.

Siccome il digest è composto da 32 byte, il numero di possibili valori che si possono ottenere applicando la funzione di hash è di 2^{256} , una cifra altissima, considerando che si stima essere il numero di atomi presenti nell'universo conosciuto. Generare le collisioni in un insieme così ampio è quindi pressoché impossibile.

Ad esempio, applicando SHA-256 alla stringa "funzione one way" si ottiene il seguente digest:

6A4F2295812B44AA3FED66A0636574956FFD7CCBCB89E19D072022BBFE563A50

Applicando la stessa funzione ma mettendo la lettera maiuscola all'inizio si ottiene:

8D7005B5169C900D7C5D9DFC19EF8E87022FB3F1A2864FE7267E0175BBF21389

Si noti come l'hash di "Funzione one way" sia completamente differente nonostante ci sia stata una sola piccola variazione.[16]

Per quanto riguarda la marca temporale, questa consente di associare una data e un'ora certe e legalmente valide a un documento informatico. Consente quindi di validare un documento, un'informazione o un dato in maniera che questo non possa più essere messo in discussione. La marca è costituita da una sequenza specifica di caratteri che identificano in modo univoco, indelebile e immutabile una data e/o un orario per fissare e accettare l'effettivo avvenimento di un dato evento. La rappresentazione della data è sviluppata in un formato che ne permette la comparazione con altre date e permette di stabilire e definire un ordine temporale. La pratica dell'applicazione di tale marca è detta “timestamping”. Bisogna comunque tenere conto del fatto che la marca temporale non garantisce l'autenticità della fonte, ma serve solo a verificarne la validità da un certo periodo in poi.

1.11 Smart Contracts

Alla base di alcune blockchain, come quella di Ethereum, uno smart contract è un programma che, al verificarsi di determinate condizioni, esegue azioni predefinite. Questi programmi possono facilitare lo scambio e il trasferimento di qualsiasi bene (ad es. azioni, valuta, contenuto, proprietà). Risiedono nella struttura blockchain e vengono attivati insieme alle transazioni. Gli smart contracts possono essere immaginati come protocolli digitali utilizzati per facilitare e far rispettare la negoziazione di un contratto legale. Infatti, in un contratto reale se una parte non rispetta gli accordi si può passare per altre vie legali, meccanismo che non avviene con i contratti intelligenti grazie al fatto che eliminano le terze parti e fungono essi da unici regolatori dei contratti stessi. Grazie alla blockchain, stanno diventando sempre più popolari poiché possono essere utilizzati più facilmente attraverso questa struttura rispetto alla tecnologia disponibile al momento della loro invenzione. Il primo a menzionarli è stato Nick Szabo, giurista e crittografo che ne ha fatto menzione in un documento del 1995, ed ha poi esteso e sviluppato il concetto due anni dopo, nel 1997.

Grazie a questo tipo di accordi si potrebbe, ad esempio, sostituire avvocati e banche che sono stati coinvolti in scambi di asset a seconda di aspetti predefiniti. Possono essere utilizzati anche per controllare la proprietà degli immobili, ponendo fine a tutte quelle controversie a livello legale che si consumano ogni giorno. Possono gestire accordi riguardanti proprietà sia materiali (ad es. case, automobili) che immateriali (ad es. azioni, diritti di accesso). Vengono scritti in vari linguaggi di programmazione come Solidity (linguaggio tipizzato staticamente e di alto livello), Vyper (basato su Python), Rust (linguaggio focalizzato sulla sicurezza) e molti altri.

Vantaggi nell'utilizzo di questi contratti risiedono ad esempio nella sicurezza e nell'affidabilità, dati dalla blockchain sulla quale risiedono, che consente di ridurre al minimo le possibilità di manomissione del contratto. Quest'immutabilità può essere al contempo un punto a sfavore; va ricordato infatti che questo tipo di contratto è pur sempre una porzione di codice scritta dall'uomo (sono ammissibili errori) e risulta necessaria una conoscenza informatica. Un codice che presenta bug e inconsistenze, reso poi immutabile, può rappresentare un bel problema siccome le debolezze di questo possono essere sfruttate da eventuali attaccanti. Non è sempre detto poi che eseguire questi contratti è meno dispendioso rispetto a seguire

```

1 pragma solidity >=0.4.22 <0.7.0;
2 contract EtherBank {
3     mapping (address => uint256) public balances;
4     function deposit() external payable {
5         require(balances[msg.sender] + msg.value >= balances[msg.
6             sender]);
7         balances[msg.sender] += msg.value;
8     }
9     function withdraw(uint256 amount) external {
10        require(amount <= balances[msg.sender]);
11        balances[msg.sender] -= amount;
12        msg.sender.transfer(amount);
13    }

```

Figura 1.7: Esempio di uno smart contract scritto in Solidity

procedure tradizionali. In figura 1.7 è presente un esempio di uno smart contract scritto in Solidity: nella prima riga si trova il range di versioni del linguaggio compatibili, in seguito vengono esposte le funzioni di prelievo (withdraw) e deposito (deposit) per un conto in banca.

Il problema del double spending

Quando si pensa a una blockchain, si pensa a un'architettura sinonimo di sicurezza, affidabilità, trasparenza e immutabilità che permette di scambiare asset unici e non replicabili. Ciò nonostante, la singolarità di questi asset potrebbe venir meno, come nel caso del “double spending”.

Le valute digitali non nascono con la blockchain, ma sono introdotte ben prima dell’arrivo del white paper di Satoshi Nakamoto. Tutti i progetti però incappavano in uno dei problemi più difficili: quello del "double spending".

Il digitale può, in maniera molto semplice, essere replicato anche a costi irrisoni, ponendo un problema evidente a livello di asset perchè ne compromette l’unicità, rendendo magari possibile una doppia spesa.

Nelle “digital currency” il compito di evitare che uno stesso asset venga speso due o più volte è affidato alle banche, che rendono consistente il valore in entrata e quello in uscita nel nostro conto, in funzione del tipo di servizio di pagamento o scambio utilizzato. Questo problema del double spending è stato affrontato attraverso sistemi di tracking.

La risoluzione risiede nella identità della moneta. La crittografia che accompagna il bitcoin e in generale le diverse declinazioni della blockchain permette di gestire l’identità della criptovaluta, con un suo specifico codice ID, un suo nome e cognome e una sua storia. Se Alice acquista un pasto con la moneta A (ID: 850567), viene contrassegnato il passaggio della moneta A dal wallet di Alice a quello di Bob in cambio di un bene. A questo punto la transazione verrà diffusa ai nodi della rete. Inoltre alla moneta A (dotata di identificativo e altre informazioni) si aggiungeranno le notizie riguardanti la transazione, ad esempio che è servita per pagare un determinato bene a una persona chiaramente identificata. La stessa moneta

A si arricchirà di altre informazioni nel momento in cui Bob la userà per altre transazioni e nuovamente quando il beneficiario di questa nuova transazione a sua volta la utilizzerà per un nuovo acquisto. Alice non potrà più disporre di questa moneta e non potrà disporre nemmeno di una copia di essa. In realtà un attacco di doppia spesa non è così frequente e probabilmente come si potrebbe pensare, ed è maggiormente pericoloso su reti più piccole, non testate con completezza e insicure. Potrebbe verificarsi anche in caso di attacchi al 51%, ma sono attacchi comunque di durata limitata e in seguito alla cessazione di un attacco del genere si ritornerebbe ad avere una chain corretta e verrebbero alla luce eventuali inconsistenze. Per eseguire con successo un attacco di doppia spesa l'attaccante A ha bisogno di ingannare un possibile nodo V per fargli accettare la transazione $TR(v)$ che non sarà più in grado di annullare successivamente. A questo punto A crea un'altra transazione $TR(a)$ che ha lo stesso input della prima ma rimpiazza l'indirizzo di destinazione, che non sarà più quello di V ma di un nodo sotto il controllo di A. Se queste transazioni vengono inviate nello stesso momento hanno possibilità (anche se piccole) di essere confermate in un blocco da validare. Il compito di questo tipo di attacco è quindi quello di convincere dapprima un peer che la transazione è avvenuta con successo, per poi convincere tutto il network ad accettare altre transazioni in modo tale che al nodo non resta né il prodotto né le monete, che vengono invece gestite dall'attaccante. In figura 1.8 viene mostrata la possibilità di successo di un attacco di doppia spesa in funzione dell'hashrate e del numero di conferme ottenute.

q	1	2	3	4	5	6	7	8	9	10
2%	4%	0.237%	0.016%	0.001%	≈ 0					
4%	8%	0.934%	0.120%	0.016%	0.002%	≈ 0				
6%	12%	2.074%	0.394%	0.078%	0.016%	0.003%	0.001%	≈ 0	≈ 0	≈ 0
8%	16%	3.635%	0.905%	0.235%	0.063%	0.017%	0.005%	0.001%	≈ 0	≈ 0
10%	20%	5.600%	1.712%	0.546%	0.178%	0.059%	0.020%	0.007%	0.002%	0.001%
12%	24%	7.949%	2.864%	1.074%	0.412%	0.161%	0.063%	0.025%	0.010%	0.004%
14%	28%	10.662%	4.400%	1.887%	0.828%	0.369%	0.166%	0.075%	0.034%	0.016%
16%	32%	13.722%	6.352%	3.050%	1.497%	0.745%	0.375%	0.190%	0.097%	0.050%
18%	36%	17.107%	8.741%	4.626%	2.499%	1.369%	0.758%	0.423%	0.237%	0.134%
20%	40%	20.800%	11.584%	6.669%	3.916%	2.331%	1.401%	0.848%	0.516%	0.316%
22%	44%	24.781%	14.887%	9.227%	5.828%	3.729%	2.407%	1.565%	1.023%	0.672%
24%	48%	29.030%	18.650%	12.339%	8.310%	5.664%	3.895%	2.696%	1.876%	1.311%
26%	52%	33.530%	22.868%	16.031%	11.427%	8.238%	5.988%	4.380%	3.220%	2.377%
28%	56%	38.259%	27.530%	20.319%	15.232%	11.539%	8.810%	6.766%	5.221%	4.044%
30%	60%	43.200%	32.616%	25.207%	19.762%	15.645%	12.475%	10.003%	8.055%	6.511%
32%	64%	48.333%	38.105%	30.687%	25.037%	20.611%	17.080%	14.226%	11.897%	9.983%
34%	68%	53.638%	43.970%	36.738%	31.058%	26.470%	22.695%	19.548%	16.900%	14.655%
36%	72%	59.098%	50.179%	43.330%	37.807%	33.226%	29.356%	26.044%	23.182%	20.692%
38%	76%	64.691%	56.698%	50.421%	45.245%	40.854%	37.062%	33.743%	30.811%	28.201%
40%	80%	70.400%	63.488%	57.958%	53.314%	49.300%	45.769%	42.621%	39.787%	37.218%
42%	84%	76.205%	70.508%	65.882%	61.938%	58.480%	55.390%	52.595%	50.042%	47.692%
44%	88%	82.086%	77.715%	74.125%	71.028%	68.282%	65.801%	63.530%	61.431%	59.478%
46%	92%	88.026%	85.064%	82.612%	80.480%	78.573%	76.836%	75.234%	73.742%	72.342%
48%	96%	94.003%	92.508%	91.264%	90.177%	89.201%	88.307%	87.478%	86.703%	85.972%
50%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Figura 1.8: Probabilità di successo della doppia spesa, in funzione dell'hashrate dell'attaccante (q) e del numero di conferme n

Attacco al 51%

Uno degli attacchi più discussi inerenti alle blockchain è quello del 51%, attacco in cui un nodo o un insieme di essi prende il controllo della rete ottenendo la maggioranza del potere computazionale, in modo tale da poterla controllare. Si tratta di uno degli attacchi più temuti all'interno del panorama blockchain e rappresenta un grande rischio per l'affidabilità del sistema. Attacchi di questo tipo non hanno avuto successo nella rete Bitcoin perché si tratta di un progetto molto ampio e si ingrandisce molto velocemente, quindi non è semplice ottenere la maggioranza della rete anche perché richiederebbe un grande dispendio economico. In realtà nel giugno 2014 il pool di mining GHash.IO ha conquistato più del 51% di tasso di hash di Bitcoin per un'intera giornata. Non si immaginava una possibilità del genere, soprattutto nella rete di Bitcoin data la portata della stessa, ma in realtà il pool di mining riuscì ad ottenere la maggioranza, anche se un mese dopo la quota di nel tasso di hash della rete scese a poco più del 38%. In questo caso Ghash si è impegnato nel fare un passo indietro, ma restava il timore che un'altra organizzazione raggiungesse nuovamente la quota di maggioranza. Questo incidente non ha rappresentato un vero e proprio attacco, infatti il compito dell'organizzazione non era quello di falsare i blocchi oppure spendere due volte gli stessi bitcoin, ma comunque nella maggior parte dei casi questi attacchi falliscono o non hanno a lungo successo. La rete Bitcoin però non è l'unica esistente e le reti più piccole, meno affidabili e meno sicure, sono maggiormente vulnerabili a questo attacco. Si sono verificati numerosi attacchi al 51% concreti su reti più piccole, tra cui Ethereum Classic. Grazie al suo volume e all'accordo della community di mantenere integra la rete, Bitcoin è rimasto al sicuro fino ad oggi. Ipoteticamente parlando però, se un'entità nella rete controlla più del 50% della potenza di calcolo, potrebbe provare a estrarre blocchi minati a partire da un blocco corrente (formando così una sorta di catena parallela). Quando viene pubblicata, questa catena concorrente supererà per lunghezza la catena originale, lasciando fuori tutte le transazioni dalla fork e, di conseguenza, sabotando l'integrità della rete. Il responsabile di una tale azione potrebbe, grazie al potere acquisito, rifiutarsi di accettare la validazione di blocchi da parte di altri partecipanti, in modo da ridistribuire il compenso in modo non equo. Addirittura potrebbe lasciare fuori dalla rete possibili concorrenti inserendo il loro indirizzo all'interno di una blacklist. Perché un attacco del 51% si verifichi l'entità in questione necessita di abbastanza potenza di hash per estrarre con successo i blocchi su una copia della main chain, copia che continua a funzionare in parallelo all'originale. Il fattore chiave per ottenere una tale potenza di calcolo sta nel comprare molte risorse per raggiungere la quota desiderata, quindi se un attore malintenzionato non è in grado di pagare per prendere il controllo di una rete, un attacco al 51% non può verificarsi. La velocità con cui un attacco può raggiungere la catena onesta può essere rappresentata dalla seguente equazione in cui p rappresenta la probabilità che il nodo onesto validi per primo il blocco, q è la probabilità che gli attaccanti estraggano il nuovo blocco più velocemente dei minatori onesti e la probabilità che gli attaccanti raggiungano i nodi onesti a partire da uno scarto di z blocchi è q_z .[17]

$$q_z = \begin{cases} 1 & p \leq q \\ (q/p)^z & p > q \end{cases}$$

Eseguire questo tipo di attacco su Bitcoin ha un costo stimato di poco più di 15 miliardi di dollari. Altre altcoin, basate su un sistema più piccolo e meno sicuro invece sono molto più a rischio. Basterebbe noleggiare una grossa (e magari anche dispendiosa) attrezzatura per il mining per poter mettere a segno un massiccio attacco. La rete di Ethereum Classic è stata violata da alcuni hacker che hanno sottratto alla rete una quantità di valuta nativa molto elevata (per un valore superiore al milione di dollari).

Attacco Sybil

Un attacco Sybil è una specifica minaccia in cui una persona prova ad assumere il controllo di un sistema creando diversi account o nodi. L'idea alla base di questo attacco è molto semplice e può essere paragonata alla creazione di diversi account da parte di una persona all'interno di un social, nel tentativo di gestire e controllare diversi nodi. Il termine deriva da "Sybil Dorsett", una donna ricoverata per disturbo dissociativo d'identità, chiamato anche sindrome d'identità dissociata. Creando sufficienti identità false i nodi malevoli potrebbero mettere in minoranza i nodi onesti prendendo il controllo della rete, e una volta fatto ciò potrebbero rifiutare di ricevere o trasmettere blocchi, bloccando altri utenti all'interno della stessa rete. Se con questo attacco si raggiunge comunque la maggioranza dei nodi allora si verifica un attacco al 51%, con tutte le conseguenze da esso derivanti. Per sventare attacchi del genere le blockchain si avvalgono degli algoritmi di consenso che contribuiscono a renderli poco pratici. Bitcoin ad esempio sfrutta un insieme di regole per la creazione di blocchi, ad esempio concede la possibilità di generare un blocco solo in relazione della potenza di calcolo reale di cui si dispone. In questo modo un potenziale attacco Sybil viene sventato nella maggior parte dei casi.

Attacco Eclipse

Si tratta di un attacco volto a interferire con i nodi di una rete. Come suggerisce il nome, questo attacco punta ad "eclissare" uno oppure un gruppo ristretto di nodi per comprometterlo e fargli avere una visione distorta della rete. Nella figura 1.9 si può vedere come i nodi rossi sono stati isolati dal nodo malevolo, che ne condiziona la visione dello stato globale. Questo tipo di attacco può sembrare a prima vista simile a quello precedente, ma l'attacco Sybil mira a compromettere tutto il network, mentre questo opera su un piccolo numero di nodi.

I nodi di Bitcoin che svolgono operazioni di mining hanno bisogno di attrezzature importanti e talvolta specializzate per la validazione di blocchi, mentre i nodi che non svolgono questo tipo di operazione sono gestibili con una minima potenza di calcolo. Un fattore limitante per molti nodi è, però, la larghezza di banda. Anche se Bitcoin risulta essere una rete molto numerosa, ci sono limiti di connessione tra nodi stabiliti a priori dal software (125 connessioni). Il dispositivo medio della rete quindi non è in grado di connettersi con un numero elevato di nodi, rendendo più semplice l'attacco. L'hacker farà in modo che tutte le

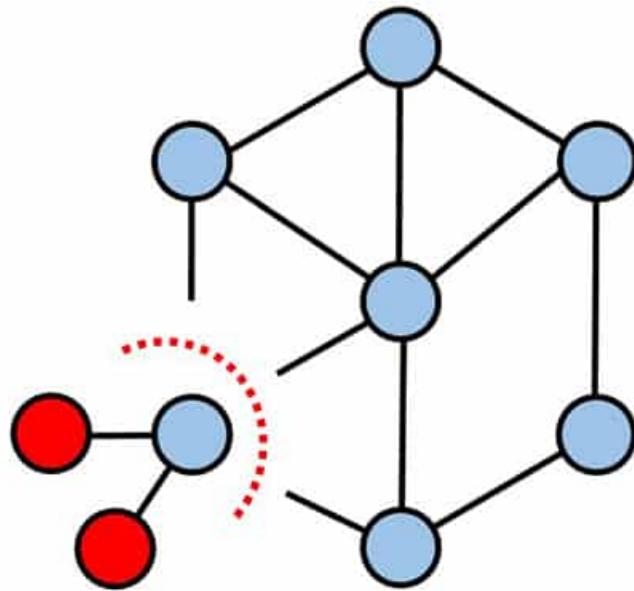


Figura 1.9: Isolazione dei nodi all'interno della rete durante un attacco Eclipse

connessioni del nodo target avvengano con nodi da lui controllati. Il primo passo è quello d'inondare il bersaglio con indirizzi IP, in modo tale da farlo connettere a uno di essi al riavvio del software; il riavvio può essere forzato (attacco DDoS) oppure l'attaccante può aspettare che avvenga senza interferire. Quando poi il target si connette nuovamente avrà una visione non consistente della rete e l'attaccante potrà forzare dati non corretti.

1.12 Tendenze future

Le blockchain sembrano essere applicabili in moltissime aree, specialmente quelle in cui si riteneva necessaria, fino a poco tempo fa, la presenza di una terza parte regolatrice. Atzori (2015) [18] suggerisce che la politica e l'intera società potrebbero essere ristrutturate a partire dalla blockchain. Molte funzioni potrebbero diventare obsolete se le persone iniziassero a organizzare e proteggere la società utilizzando piattaforme decentralizzate e sostiene inoltre che è possibile anche un decentramento dei servizi governativi attraverso la Blockchain.

Pertanto, la tecnologia blockchain consente di stabilire regole tra parti utilizzando la crittografia e di sostituire intermediari che sono stati necessari per stabilire la fiducia in passato. Il settore bancario e quello economico in generale si sentono minacciati dalla blockchain e dalle sue potenzialità. Questa paura è evidente considerando i vantaggi che il sistema apporta. Se le persone pagano oggi la merce con bonifico, il pagamento effettivo avviene dopo diversi giorni. Utilizzando la blockchain, questo pagamento può essere effettuato in tempo reale regolando il libro mastro, pur non rinunciando a caratteristiche importantissime come l'affidabilità. La

ricchezza può essere protetta in modo più efficace utilizzando la blockchain. Soprattutto nel terzo mondo, i proprietari terrieri hanno problemi a provare il possedimento dei propri beni se per esempio il governo locale mira a espropriarli. Queste minacce possono essere controllate integrando i titoli delle terre e dei beni nella blockchain, permettendo quindi di eliminare ogni tipo di discordia.

Nell'ambito dell'healthcare la blockchain può rivelarsi molto utile per consolidare la trasparenza nei meccanismi, la sicurezza nella gestione delle informazioni, l'interoperabilità fra sistemi e i rapporti tra diversi stakeholders in questo panorama. Modelli basati sulla blockchain possono portare ad una nuova formula di interconnessione basata sulla collaborazione ed interazione tra sistemi. L'obiettivo di una tale soluzione sarebbe ad esempio l'introduzione delle terapie digitali nel sistema sanitario nazionale (SSN). La blockchain è performante proprio in quei casi in cui è necessaria la condivisione e il salvataggio di dati in maniera sicura e immutabile, permettendo la trasparenza dei processi per il paziente, in modo tale da renderlo più consapevole e partecipe nella gestione e nel controllo dei dati.

In conclusione, adottare soluzioni blockchain-based permetterà un maggiore controllo, e con l'evoluzione di questa tecnologia si toccheranno sempre nuovi aspetti per rendere il progresso maggiore ed espandere la visione di questo panorama. Gli ambiti di applicazione sono vastissimi e gli studi in materia permettono uno sviluppo veloce del sistema. Se le persone continueranno a credere nelle capacità di questa tecnologia si arriverà a un cambio notevole in vari settori; la tecnologia si introduce in tutti gli ambiti più importanti risolvendo problemi e migliorando le tematiche attualmente prese in considerazione.

Capitolo 2

Sistema sanitario in Italia

2.1 Situazione attuale

Abbiamo analizzato la blockchain, le sue parti e i meccanismi che regolano questa tecnologia, che in poco tempo si è affermata in molti ambiti, andando a cambiare la concezione degli stessi (si pensi ad esempio al settore bancario dopo l'introduzione di criptovalute come Bitcoin), incrementando la sicurezza e rendendo più agevole l'accesso alle informazioni. Alcuni settori però risultano essere ancora poco sviluppati in tal senso, anche se cercano di rincorrere quella che è l'evoluzione del campo e, più in generale, del mondo tecnologico. Si pensi ad esempio al caso del settore sanitario. Particolare attenzione va posta ai dati, alle cartelle cliniche e tutto lo storico di un paziente. Ci si ritrova davanti a quello che potrebbe essere un grande problema: le informazioni non sono sempre reperibili nel momento e nel modo in cui si vorrebbe e si presenta un rischio di inconsistenza e incoerenza tra dati salvati nelle diverse applicazioni. L'integrazione e l'accessibilità di tutti i dati, suddivisi in differenti basi di dati, rappresenta una necessità per tutte le aziende, per garantire la possibilità di detenere il proprio patrimonio informativo e ridurre al minimo il rischio di dispersione o ambiguità.

Il servizio sanitario nazionale (SSN) è un insieme di servizi e strutture che hanno lo scopo di garantire assistenza ai cittadini e l'accesso libero all'erogazione delle prestazioni sanitarie.

Queste strutture si basano su alcuni principi organizzativi che mettono al centro del sistema l'individuo, che possiede una serie di diritti come quello di essere informato sulla malattia, come la libertà di scegliere il luogo di cura oppure il diritto alla riservatezza delle informazioni personali. Altri principi basilari sono ad esempio l'integrazione socio-sanitaria che prevede l'associazione di assistenza sanitaria e sociale quando il cittadino richiede una prestazione sanitaria e la collaborazione tra i vari livelli del SSN. Tutte le strutture (Stato, comuni, regioni) devono collaborare tra di loro (sempre rientrando nelle proprie competenze) per assicurare condizioni di salute omogenee su tutto il territorio e prestazioni sanitarie appropriate per ogni cittadino.

C'è da dire inoltre che errori in questo sistema di distribuzione e gestione di informazioni spesso delicate possono comportare seri problemi e perdite di tempo, fattori non ammissibili quando si parla di un ambito così delicato. Infatti, questo servizio ricopre una bella fetta di mercato in quanto attorno alla sanità si muovono più di 100 miliardi di euro, ovvero circa

il 9% del PIL nazionale. Questi dati mettono in luce l'importanza del settore e quanto sia decisivo il corretto funzionamento di questa macchina per l'erogazione di servizi. Per dare un'idea della quantità di persone coinvolte in questo sistema, facciamo riferimento ai dati del ministero della salute [19] relativi al 2017 (figura 2.1); si tratta di un numero complessivo di 603.375 dipendenti (numero pressoché simile agli anni precedenti), con un 66,8% composto da donne e il restante 33,2% da uomini.



Figura 2.1: Ministero della salute: dati relativi al personale sanitario del 2017

I sistemi sanitari si sono evoluti nel tempo secondo una logica scollegata, che prevede l'elaborazione di dati in ambienti differenti, creando una mancata collaborazione. Ogni parte gestisce un sottoinsieme dell'intero patrimonio informativo che a volte risulta essere anche replicato, non direttamente accessibile alla parte interessata e con poca sincronizzazione. Le strutture stesse poi si pongono come un ostacolo rispetto alle esigenze di integrazione e condivisione dei dati.

Questo comporta conseguenze sia dal punto di vista del rischio clinico che dal punto di vista del profilo normativo, oltre a quello della ricerca e della prevenzione. Questa inconsistenza tra dati, che risultano essere frammentati in diverse strutture e gestiti da diverse politiche di accesso, rende molto difficili alcuni processi, come presentare un quadro complessivo dello stato di salute del paziente (per assicurare la veridicità delle decisioni cliniche); assicurare la continuità dei processi in modo da evitare trascrizioni manuali oppure trasmissioni verbali; rispettare i requisiti dei regolamenti per soddisfare le esigenze di sicurezza e di accesso ai dati; disporre di un patrimonio informativo coerente e il più ampio possibile.

Va inoltre considerata la dipendenza che le strutture hanno rispetto ai fornitori ai quali bisogna rivolgersi per l'acquisizione di dati d'interesse. Questa dipendenza determina le cosiddette situazioni di “vendor lock-in”, con le conseguenti difficoltà dell'organizzazione di essere realmente indipendente rispetto ai fornitori anche in fase di evoluzione del sistema in-

formativo, di rilevanza anche sotto il profilo giuridico. Vale la pena allora ricordare le linee guida dell’Autorità Nazionale Anticorruzione, [20] che indicano come causa d’infungibilità “i costi della migrazione di dati (e documenti) informatici” e raccomandano quanto previsto dal “Quadro europeo di interoperabilità – Strategia di attuazione”.

In particolare si ritiene che “tutti i dati pubblici dovrebbero essere liberamente accessibili per l’utilizzo e il riutilizzo da parte di terzi” e che “le pubbliche amministrazioni devono rendere l’accesso e il riutilizzo dei loro servizi pubblici e dati indipendente da qualsiasi tecnologia o prodotto specifici”.

La frammentazione dei dati (che risultano circoscritti a determinati ambiti e salvati secondo regole sintattiche e semantiche differenti) è incentivata dall’evoluzione dei sistemi di assistenza e cura che tendono a migrare sempre maggiormente verso una deospedalizzazione, grazie anche all’espansione tecnologica di strumenti medici e vari dispositivi IOT. Queste divergenze tra i dati causano rallentamenti nella cura dei pazienti e nell’analisi della loro situazione, compromettendo la capacità e l’affidabilità nel recupero degli storici. In alcuni casi questo può portare a un complesso di informazioni ridondanti, confusionarie o addirittura incompatibili.

Facciamo un esempio: supponiamo un individuo faccia un incidente in auto. Questo deve essere ricoverato presso un ospedale. Allora è necessario che i medici facciano molte analisi, accertamenti e studi per indagare il caso proposto. Questi devono ricostruire la storia sanitaria del paziente analizzando anche eventuali malattie pregresse, disturbi, intolleranze o allergie e assunzione di farmaci. I dati in questione spesso sono ridondanti e frammentati: non esiste un fascicolo sanitario unico del paziente cui fare riferimento in maniera veloce, sicura e affidabile. Infatti, questi dati sono disseminati in vari database, non comunicanti e che magari presentano qualche incoerenza. Se, al contrario, questi dati fossero inseriti nella blockchain, verrebbe progressivamente costruita la storia clinica del paziente, che sarebbe una storia continua, affidabile, facilmente consultabile. Per altro il fascicolo potrebbe essere aggiornato in tempo reale e consultato contemporaneamente da tutte le strutture sanitarie. Si arriverebbe nel complesso a un sistema interconnesso, che sfrutta la decentralizzazione della tecnologia blockchain, oltre ai mille vantaggi che questa tecnologia può apportare.

2.1.1 Rapporto con gli altri paesi europei

L’Euro Health Consumer Index (EHCI) è uno strumento che permette di confrontare la situazione a livello sanitario in ben 35 paesi. Il confronto è effettuato sulla base di ben 46 parametri che includono aree come diritti dei pazienti, informazioni personali, accesso alle cure e risultati del trattamento attraverso l’utilizzo di farmaci. Secondo il rapporto del 2018 [21] i paesi più evoluti in ambito di assistenza sanitaria sono Svizzera (che si colloca al primo posto con un totale di 893 punti), Norvegia, Danimarca, Belgio e Finlandia. L’Italia presenta grandi variazioni a livello regionale (come la Spagna) e si colloca soltanto al 20º posto, con 687 punti. Il nostro paese presenta la più grande differenza di PIL pro capite tra le regioni; quelle più povere sono quelle meridionali e si arriva fino a un terzo rispetto ai dati della Lombardia

(che risulta essere la regione più ricca). In figura 2.2 osserviamo la classifica EHCI dei vari paesi europei, con la Svizzera che ottiene la prima posizione, mentre l'Albania l'ultima.

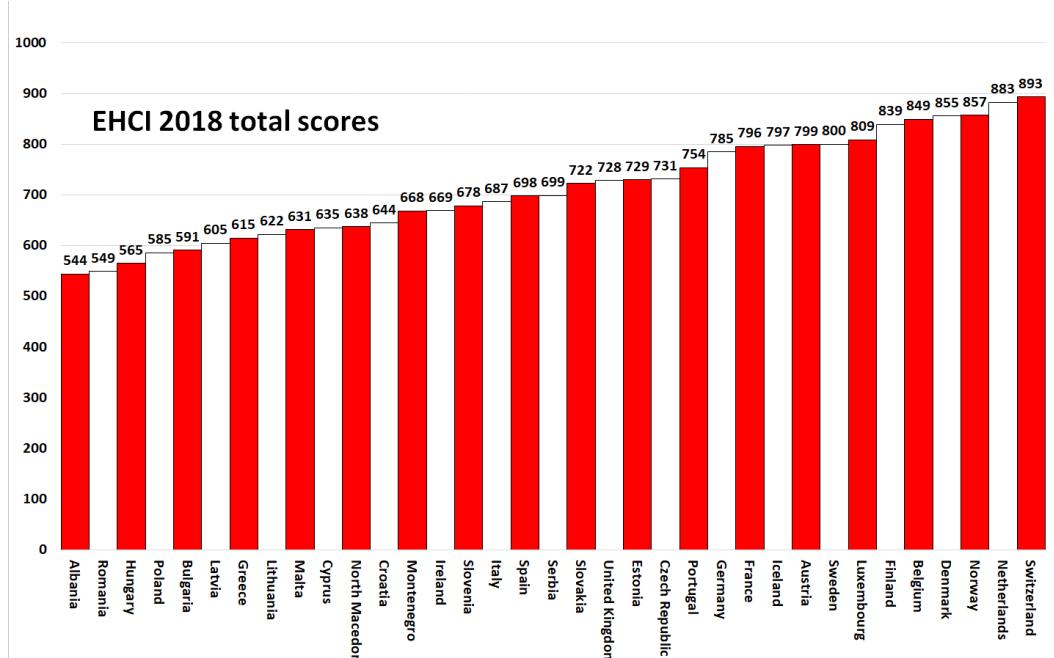


Figura 2.2: Classifica finale Euro Health Consumer Index (Italia al 20° posto)

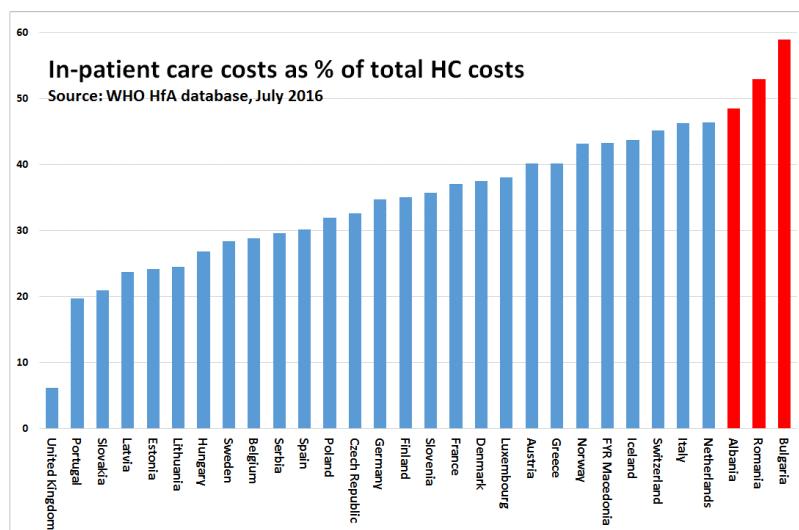


Figura 2.3: Costo dell'assistenza ospedaliera nei vari paesi europei

2.1.2 Fascicolo sanitario elettronico

Il 5º rapporto GIMBE sul servizio sanitario nazionale (11 ottobre 2022) stima che la spesa sanitaria presenterà delle riduzioni nel biennio 2023-2024 mentre si avverrà una stabilizzazione nel 2025, dovuta alla minore spesa causata in precedenza dalla pandemia. È necessario che il cittadino assuma un ruolo attivo nel sistema, diventando uno dei coordinatori delle proprie informazioni, attraverso l'accesso a internet e anche grazie al fascicolo sanitario elettronico (FSE). La figura 2.4 mostra come, dei cittadini intervistati, solo il 26% ha sentito parlare del FSE anche prima della pandemia. Il monitoraggio dei dati e del proprio stile di vita dovrebbe avvenire tramite applicazioni e tramite una comunicazione con il proprio medico anche per via digitale.

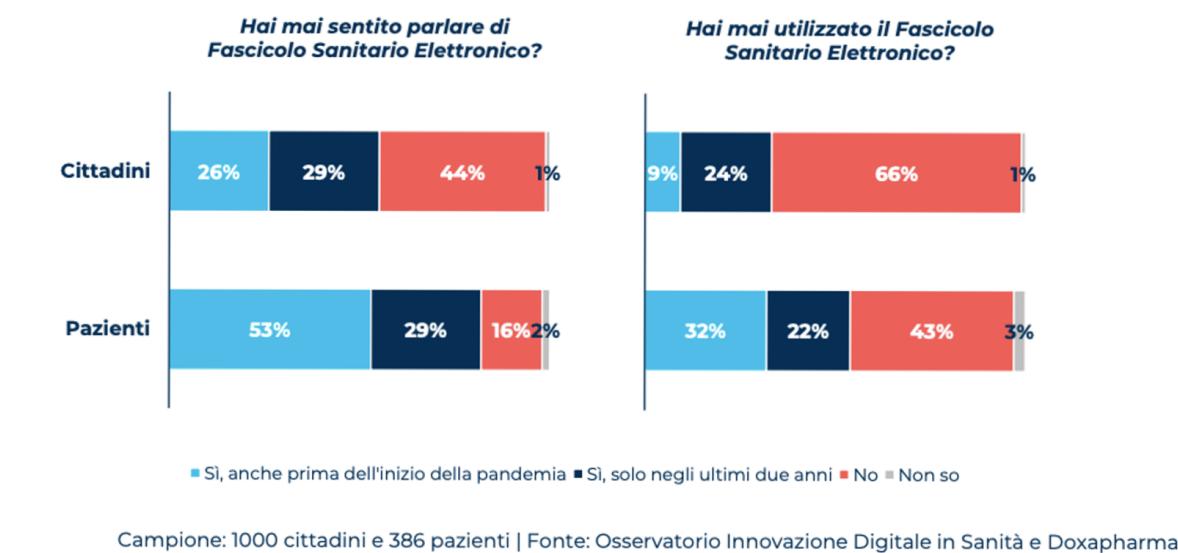


Figura 2.4: Conoscenza e utilizzo del FSE da parte dei cittadini e dei pazienti italiani

È stato stimato che oltre il 50% delle persone ritira il proprio referto medico presso la struttura (spesso delega qualcuno per farlo) e che il tempo medio per fare ciò si aggira attorno ai 45 minuti. Se questo fenomeno avvenisse per via digitale si ridurrebbe notevolmente il tempo in questione, che scenderebbe al di sotto dei 5 minuti. Non si tratta solo di una questione di tempo ma anche di denaro, infatti ciò (oltre a evitare ovvi rallentamenti) porterebbe a un notevole risparmio. Una copia della cartella clinica cartacea generalmente costa oltre i 15€: cifra con la quale l'ospedale copre le spese relative all'attività del personale e al materiale di cancelleria. Un approccio improntato su sistemi digitali rappresenta quindi anche un risparmio di denaro per i pazienti. Dal punto di vista tecnologico però il servizio sanitario nazionale è ancora arretrato. I vecchi sistemi non sembrano rispondere più correttamente alle esigenze apportate dalla tecnologia e, nonostante ciò, i fondi per investire in tal senso non sono elevatissimi. Molti sistemi sono lenti, arretrati e obsoleti e non gestiscono le informazioni e le richieste come invece dovrebbero. Spesso queste realtà sono anche diverse tra loro: macchine

costruite in maniera differente, con scopi differenti e capaci di elaborare dati spesso non combacianti. Da qui derivano i problemi di comunicazione, di affiliazione delle informazioni e di consistenza dei dati. La discordanza tra questi sistemi porta all'elaborazione di dati non coerenti, fattore che può inficiare sull'analisi e sulla diagnosi dello stesso paziente, e può aumentare in maniera considerevole i tempi di attesa e i costi relativi al trattamento. Grazie alla digitalizzazione delle informazioni e all'utilizzo di un fascicolo elettronico si eviterebbero inutili code, tempi per eventuali spostamenti e costi annessi, rendendo tutto il sistema più efficiente e meno dispersivo, andando maggiormente verso un'ottica in cui il paziente è al centro del discorso. Il piano nazionale ripresa e resilienza (PNRR) propone di ridistribuire il modello di assistenza all'interno del nostro territorio, per andare sempre più incontro ai pazienti. È stata prevista una riforma del modello organizzativo della rete di assistenza territoriale che si basa sull'incremento dell'assistenza presso domicilio, sulla creazione di nuove strutture sanitarie che mirano a migliorare l'accessibilità e su un'ottica one health per l'integrazione di un nuovo assetto istituzionale che contribuisca a migliorare questo ambito nel nostro territorio. Uno dei punti chiave di questa riforma è proprio la telemedicina: si vuole incentivare l'utilizzo di dispositivi elettronici, internet, software e sistemi digitali per facilitare l'erogazione di servizi e prestazioni sanitarie a distanza. Nel PNRR sono stati previsti aiuti per l'Europa per un totale di 191 miliardi di euro, da distribuire in vari settori e in vari modi. La 6^a missione è quella dedicata alla sanità, per la quale sono stati stanziati 15,6 miliardi di euro circa. Di questi, 7 miliardi stanziati per le strutture e per l'assistenza sanitaria territoriale mentre la restante parte per l'innovazione, la ricerca e la digitalizzazione del servizio sanitario nazionale. Il ministero dell'economia e delle finanze ha confermato che sono stati conseguiti nei tempi previsti tutti i traguardi e obiettivi indicati dal PNRR per il primo semestre 2022.[22]

2.1.3 Blockchain come soluzione dei problemi del settore sanitario

Appurato come gli attuali sistemi non sono in grado di garantire trasparenza, immutabilità, affidabilità e tracciabilità, analizziamo soluzioni in grado di porre rimedio a queste lacune. Sistemi improntati su tecnologie blockchain possono risolvere tali problematiche; si tratta infatti di uno strumento in grado di fornire la connessione, l'integrazione e l'affidabilità richiesta, con tutta la sicurezza che un tale sistema distribuito può garantire. Gli sforzi fatti per introdurre questa tecnologia nel settore della salute sono aumentati in maniera esponenziale negli ultimi anni e ci sono particolari aree della medicina che potrebbero essere fortemente influenzate da essa. IBM ha mostrato come il 70% dei leader nel settore sanitario prevede che il maggior impatto della blockchain consisterà nel miglioramento nella gestione delle sperimentazioni cliniche e della fornitura di un quadro decentralizzato per la condivisione delle cartelle cliniche elettroniche (EHR). Il controllo degli accessi, l'interoperabilità e l'integrità dei dati sono questioni che dovrebbero essere migliorate dalla blockchain in questo campo, ed Ethereum e Hyperledger Fabric sembrano essere le piattaforme/framework più utilizzate in questo dominio.

2.2 PharmaLedger

PharmaLedger è un grande progetto che riguarda l'ambito sanitario e parte proprio dal presupposto che questo settore risulta essere caratterizzato da varie mancanze e problemi, come quella dell'inconsistenza dei dati. Inoltre, il costo dell'assistenza sanitaria è in aumento negli ultimi anni a causa delle sempre più crescenti necessità di prevenzione e sostegno. L'aumento generale dei costi è caratterizzato anche da una crescita demografica (0,9% variazione annua), dal costo dello sviluppo di nuovi farmaci e dall'aumento delle malattie croniche. PharmaLedger è un consorzio sponsorizzato dall'Innovative Medicines Initiative (IMI) e dalla European Federation of Pharmaceutical Industries and Associations (EFPIA), che riunisce molte aziende farmaceutiche ed enti privati. Non sono solo aziende farmaceutiche però a collaborare con il gruppo, ma una serie di aziende tecnologiche, ospedali, istituti di ricerca, università e operatori sanitari. In figura 2.5 sono mostrati alcuni tra i principali collaboratori e partner del progetto.



Figura 2.5: Aziende farmaceutiche globali in collaborazione con il progetto PharmaLedger

2.2.1 Obiettivi e casi d'uso

L'obiettivo del progetto è la realizzazione di una blockchain per il supporto al settore sanitario, una piattaforma affidabile e sicura che garantisce il progresso e la risoluzione di problemi attraverso l'innovazione tecnologica. Il progetto mira a fornire una piattaforma basata su blockchain focalizzata su tre macro aree: la catena di approvvigionamento (Supply Chain), le sperimentazioni cliniche (Clinical Trials) e i dati sanitari (Health Data). Nonostante l'attenzione sui tre domini, PharmaLedger tiene a sottolineare che la piattaforma blockchain mira a servire qualsiasi caso d'uso nel settore sanitario. Uno dei casi d'uso di cui si occupa il progetto è la supply chain, l'insieme di processi che permette di portare i beni sul mercato scambian-
doli da fornitore a cliente. Questo è un processo molto complesso, che comprende varie figure

professionali, dall'approvvigionamento alla distribuzione. Anche in questo caso la blockchain può essere di fondamentale importanza in quanto potrebbe permettere di gestire e monitorare i vari processi logistici. Inoltre, in questi anni, si è vista una sempre maggiore falsificazione dei medicinali, situazione che può essere risolta dalla tecnologia tenendone traccia, dalla loro produzione alla distribuzione. La soluzione basata sulla catena di approvvigionamento mira proprio a ridurre le frodi e a rispettare le normative vigenti. PharmaLedger si occupa anche di fornire un sistema distribuito per la condivisione dei dati, data la logica a 'silos' con cui l'attuale sistema si sviluppa. Il consorzio si propone di sviluppare una soluzione blockchain trasparente, verificabile e tracciabile per il controllo degli accessi e la gestione delle cartelle cliniche nel settore della sanità. In figura 2.6 sono presentati gli obiettivi del progetto e si può notare come il fulcro sia proprio la realizzazione di una struttura per il monitoraggio dei principali casi d'uso proposti. Il desiderio del gruppo è quello di creare API che si fondono con la tecnologia, pur rimanendone distaccate, in maniera tale che ci sia un certo livello di indipendenza, nel caso in cui si volesse cambiare la struttura sottostante e non l'applicazione.

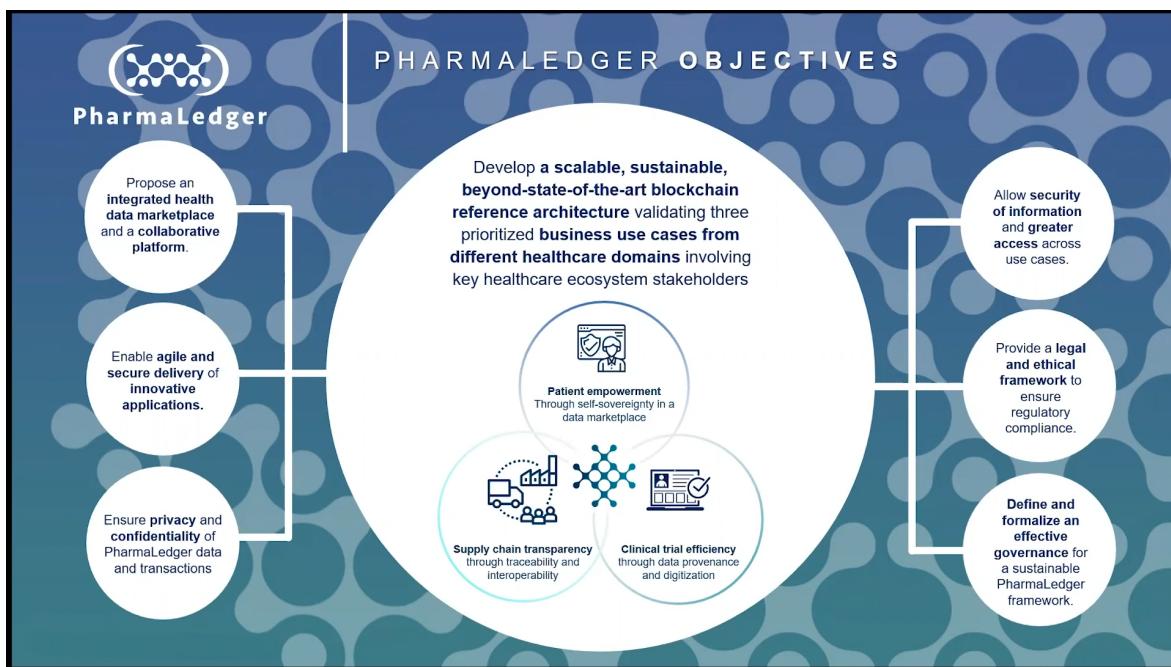


Figura 2.6: Obiettivi principali del progetto PharmaLedger

2.3 Hyperledger Fabric

2.3.1 Introduzione

Hyperledger è un progetto della Linux Foundation che nasce nel 2015, con l'obiettivo di portare trasparenza, scalabilità ed efficienza a un mercato già caratterizzato dalla presenza di soluzioni blockchain. Hyperledger coordina una comunità di organizzazioni e sviluppatori software che si occupano di implementare piattaforme, librerie e soluzioni di vario tipo. Il progetto è composto da organizzazioni leader in vari settori, tra cui quello bancario, finanziario, sanitario e tecnologico. È un progetto open source (tutto il codice implementato è disponibile online sotto la licenza Apache) e consiste in una serie di progetti software pensati per l'implementazione di blockchain a livello aziendale. Questa apertura garantisce una piena libertà e trasparenza a livello di implementazione, infatti tutti i progetti sono guidati dalla comunità. La fondazione funge solo da spina dorsale per servizi e progetti e gestisce la crescita della comunità, lasciando massima libertà di collaborazione, controllo e revisione a chiunque. Hyperledger si basa sulla fiducia, trasparenza e affidabilità che le blockchain possono portare all'interno del mercato aziendale. Molte organizzazioni infatti necessitano di condividere i dati all'interno di un database distribuito in cui entra anche in gioco la fiducia e la sicurezza nelle informazioni registrate. Secondo il rapporto "Time for Trust" la maggior parte delle associazioni migrerà verso l'utilizzo della blockchain entro il 2025. Questa tecnologia inoltre può potenzialmente apportare un valore di 1,76 trilioni di dollari all'economia globale entro il 2030. [23]

Hyperledger Foundation ospita un'ampia gamma di progetti open source e guidati dalla comunità che rientrano in vari ambiti (figura 2.7). La diversità di questi strumenti riflette il livello di apertura e libertà che il sistema fornisce e permette alle aziende di modellare strumenti in base alle proprie esigenze. Sono presenti più di 10 progetti tra cui Fabric, una permissioned blockchain promossa da IBM.

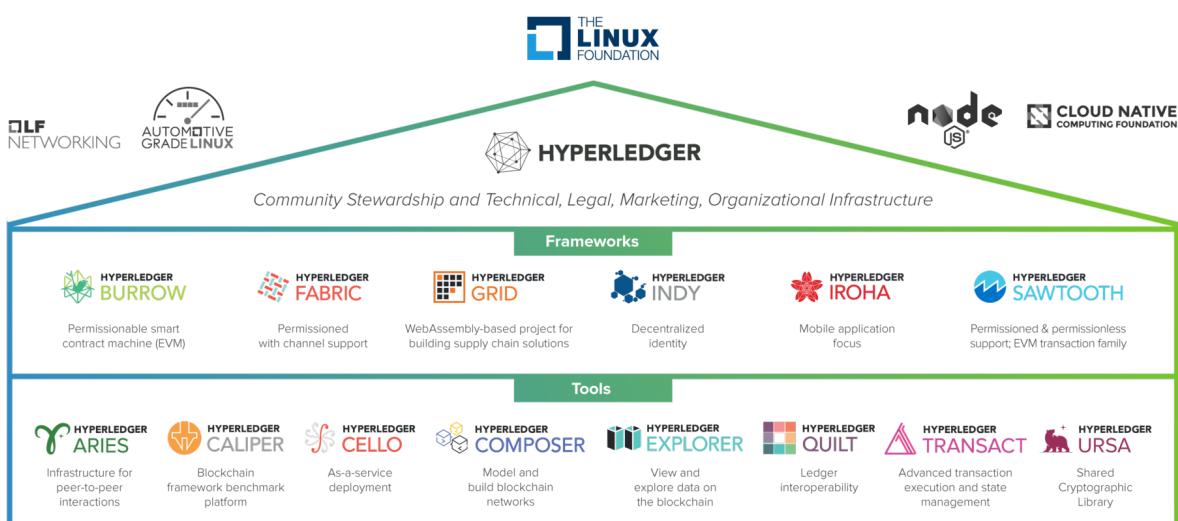


Figura 2.7: Blockchain frameworks e tools di Hyperledger

2.3.2 Caratteristiche principali

Fabric è un framework modulare per blockchain usato come punto di partenza per lo sviluppo di applicazioni a livello aziendale. Quindi Fabric si basa su un'architettura altamente modulare e consente di poter implementare smart contracts con qualsiasi modello desiderato. Sono poi presenti i "canali", che consentono di mantenere la comunicazione privata. Gli smart contracts sono detti anche chaincode e possono essere scritti con differenti linguaggi di programmazione come Go, Java e Node.js, oltre a essere presente un supporto per Solidity e per L'Ethereum Virtual Machine. Hyperledger Fabric è inoltre utilizzato in vari progetti nell'ambito healthcare, ad esempio per gestire la minaccia che ha rappresentato il Covid-19, per incrementare l'efficienza nella trasmissione dei dati clinici o per creare modelli di accesso ai dati critici (che devono restare privati). Fabric è formata da un ledger principale che può supportare vari tipi di DBMS, smart contracts programmabili in differenti linguaggi come già anticipato, un servizio di gossip per la diffusione dei blocchi e un service provider che ha il compito di dare un'identità digitale ai membri della rete. Infatti, quando si tratta di uso aziendale, alcune caratteristiche importanti sono proprio l'identità dei partecipanti, le elevate prestazioni della rete e la privacy e riservatezza delle transazioni e dei dati relativi alle transazioni commerciali. Grazie alla sua modularità, Fabric consente una grande capacità di adattamento e personalizzazione, che risiede ad esempio nel supporto per diversi protocolli di consenso collegabili. Non è presente inoltre una criptovaluta nativa, ciò implica che non sono necessarie le operazioni di mining, che tra l'altro sarebbero fonte di un grande dispendio energetico. Solitamente le blockchain presentano un'architettura order-execute in cui le transazioni vengono prima validate e ordinate, poi eseguite nell'ordine previsto su tutti i nodi. Fabric introduce un nuovo tipo di architettura per le transazioni: execute-order-validate. In questo processo le transazioni vengono validate ed eseguite, poi vengono ordinate tramite un protocollo di consenso per poi essere convalidate.[24]

2.3.3 Architettura

I requisiti che caratterizzano le blockchain variano: per alcune reti sarebbe meglio avere dei blocchi di dimensione maggiore e dei meccanismi di consenso molto rapidi, per altre un processo più lento rimarrebbe comunque accettabile. Per queste differenze inerenti a scalabilità, rapidità e sicurezza, Hyperledger utilizza una strategia a ombrello, che incoraggia il riutilizzo di elementi comuni attraverso la modularità. Questa architettura rende ogni componente indipendente e garantisce l'estensibilità e la flessibilità delle parti. Tutti i progetti Hyperledger seguono questa filosofia e vengono distinte le varie parti che caratterizzano le blockchain. È presente un livello di consenso responsabile della verifica della correttezza delle transazioni nei blocchi; un livello degli smart contracts in grado di determinare l'esecuzione delle transazioni in base ad alcune condizioni; un livello di comunicazione responsabile del trasporto di messaggi e varie APIs per permettere ai clienti di interfacciarsi con la blockchain. Sono presenti poi un servizio dedicato all'identificazione e un servizio di policy che di occupa di verificare e gestire le politiche di sistema.

Transazioni

I meccanismi di consenso nei vari progetti cambiano e in Hyperledger Fabric il consenso viene suddiviso in tre fasi: endorsement, ordering e validation. La prima parte è basata sulle politiche che permettono di approvare una transazione, la seconda fase consente di ordinare le transazioni mentre la fase finale è quella di convalida dei risultati. Le varie applicazioni possono utilizzare diversi modelli di endorsement, ordering e validation a seconda delle loro necessità e in particolare, l'API del servizio di ordering, consente di trattare processi di consenso basati su BFT (byzantine fault tolerance).

I nodi si dividono in:

- Client: agiscono per conto degli utenti finali e si occupano della creazione e dell'avvio delle transazioni.
- Peer: mantengono una copia del libro mastro e sono di due tipi. Esistono infatti gli endorser che hanno il compito di approvare le transazioni e i committer che verificano e approvano i risultati ottenuti, prima che le transazioni vengano salvate sulla chain.
- Orderer: si occupano di ordinare le transazioni approvate in blocchi, per poi spedirle ai committer.

Il processo di gestione delle transazioni è illustrato in figura 2.8

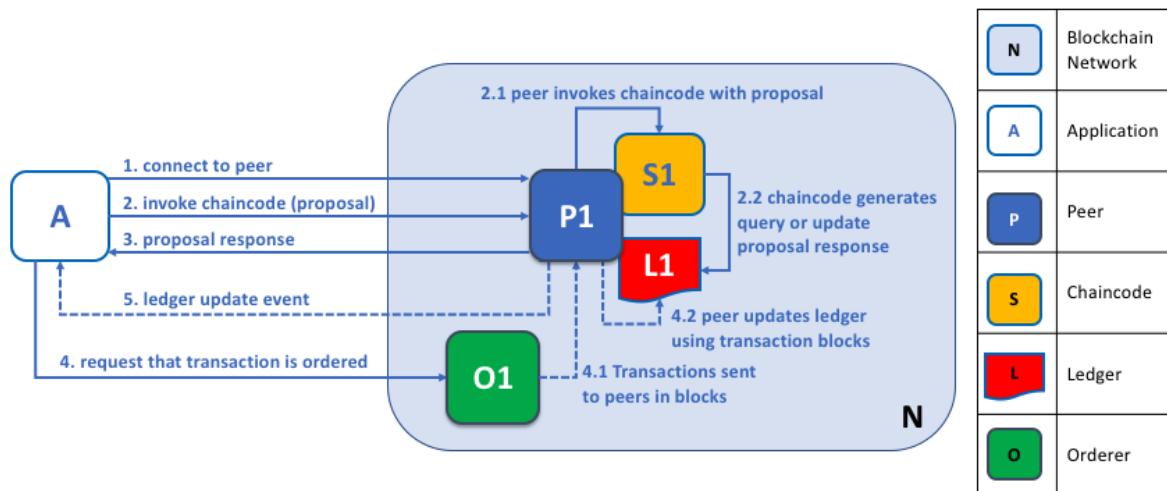


Figura 2.8: Flusso delle transazioni in Hyperledger Fabric [25]

Un cliente inoltra la transazione creata a un certo numero di endorser e ognuno di essi la esegue in una sandbox, utilizzando le regole prestabilite per la verifica della correttezza. A questo punto il client attende che si raggiungano un certo numero di approvazioni e in seguito invia la risposta agli orderers, che eseguono a loro volta il servizio di ordinamento. Questi nodi dapprima raggiungono un consenso riguardo all'ordine delle transazioni e poi suddividono il tutto in blocchi. Alla fine i blocchi vengono consegnati ai peer, che si impegnano nella registrazione dei dati sulla blockchain. Più nello specifico, quando il client invia una transazione agli endorser, questi verificano la firma del client e simulano la transazione invocando il

chaincode corrispondente e la copia del world state che hanno in locale. Quindi gli endorser calcolano le dipendenze tra la versione letta e gli aggiornamenti di stato (readSet e writeSet). Il readSet contiene le coppie chiave-valore delle chiavi lette sulla transazione mentre il writeSet le coppie di chiavi modificate.

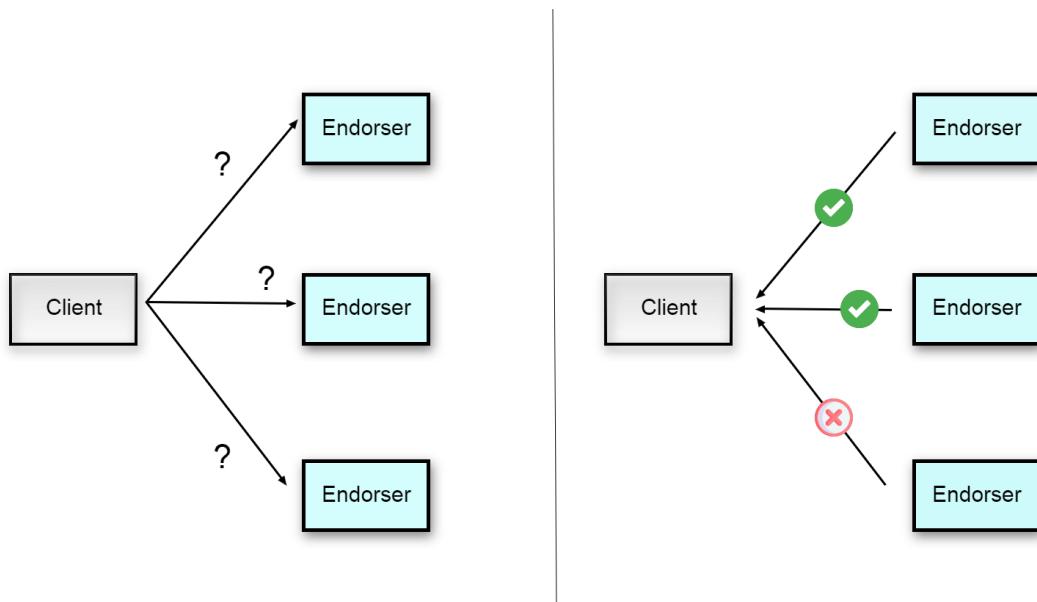


Figura 2.9: Processo di inoltro transazione e ricezione conferme da parte degli endorser

Ci sono due possibili azioni che il client può compiere: eseguire una transazione per interrogare il ledger oppure per aggiornare i dati presenti su di esso. Nel primo caso, quando il client riceve una risposta dall'endorser, il flusso termina. Nel secondo caso il processo continua con le altre due fasi, sino ad arrivare all'ordinamento delle transazioni e al loro salvataggio all'intero della blockchain.

Chaincode

Un chaincode è un programma (scrivibile con Go, Java e Node.js) che implementa un'interfaccia. Questo in genere gestisce la logica di business, quindi è simile a uno smart contract. Nel mondo di Hyperledger ci sono sostanzialmente due tipi di smart contracts:

- Gli smart contract installati inseriscono la logica di business ai validatori prima che la rete venga avviata.
- Gli Smart contract on-chain fanno parte del libro mastro e inseriscono la logica di business come una transazione.

Quattro progetti Hyperledger supportano gli smart contracts: Fabric, Burrow, Iroha e Sawtooth.

In Fabric, i chaincode includono degli smart contract che vengono installati nei nodi della rete da membri autorizzati e permettono di eseguire le azioni previste nel codice. Questi

vengono invocati dai client per la generazione delle transazioni e vanno a leggere o scrivere le coppie chiave valore del worldstate. Il world state è una componente del ledger e ne descrive il suo ultimo stato. Lo stato creato da un chaincode ha visibilità ristretta a esso, quindi non può essere acceduto direttamente da un altro. Ce ne sono due tipi principali: chaincode di sistema e di applicazione. I chaincode di sistema gestiscono le transazioni relative al sistema e si occupano ad esempio del controllo del life-cycle o delle varie policy, mentre gli altri governano lo stato delle applicazioni sul ledger, includendo i record di dati e gli asset digitali. Un chaincode inizia con un pacchetto contenente i meta dati del contratto (nome, versione, firme per assicurarne la validità) che, dopo essere stato creato, viene installato sui nodi delle controparti. A questo punto un membro della rete può attivare il codice inviando un'apposita transazione alla rete. Se la transazione viene approvata, allora il chaincode entra in azione e può svolgere il suo compito.

Canali

I canali in Hyperledger Fabric sono una componente fondamentale. Fabric è infatti un grande network che collega tutte le sue parti, e i canali vanno a creare una sorta di sotto rete per far sì che due o più membri comunichino in maniera privata. Ogni transazione all'interno della rete viene eseguita su un canale, e ogni parte deve essere autenticata. Quando un peer vuole aggiungersi a una comunicazione, per far sì che si autentichi, un provider di servizi (MSP) gli fornisce un'identità. Per la creazione di un nuovo canale invece, entra in gioco un client SDK, che chiama un chaincode di sistema dando il via alla richiesta. Viene creato così un registro per il canale che serve a memorizzare le informazioni riguardanti esso, ad esempio le varie politiche. Nonostante qualsiasi di questi peer possa appartenere a più canali e pertanto mantenere più ledger, nessun dato di un ledger può passare da un canale a un altro. Questa separazione è definita dal chaincode di configurazione, dall'MPS e dal protocollo di gossip di diffusione dei dati. Questo isolamento dei dati consente ai membri della rete di interagire tra loro scambiando transazioni private e confidenziali.

Capitolo 3

MedChain, la blockchain in ambito sanitario

La blockchain rappresenta la possibilità della più efficace innovazione per la sanità. Questa permetterebbe di trasformare le informazioni rendendole maggiormente adattabili in un'ottica distribuita anziché centripeta. Per quanto non sviluppato adeguatamente però, il settore sanitario non è rimasto immune alle dinamiche innescate dalla blockchain, tecnologia così dirompente. In ambito sanitario, la blockchain è stata indicata come la panacea di quasi tutti i mali, dalla difficoltà di scambio dati all'integrazione delle cartelle cliniche, passando per il tracciamento dei farmaci e la gestione dei trial clinici.

In questo campo, a causa di business model non sempre chiarissimi e difficoltà d'innovazione, la perplessità su tale approccio è molto marcata e non sono in molti ad avere adeguate competenze per poter comprendere le potenzialità della tecnologia. Quindi questo settore può riorganizzare i processi gestionali affidandosi alla blockchain per lo sviluppo di nuove soluzioni per una serie di casi d'uso.

L'obiettivo di questo capitolo è proprio quello di presentare l'implementazione di una blockchain che permetta ai medici e al personale sanitario di memorizzare i dati in una struttura immutabile e sicura, rendendoli così fruibili in qualsiasi momento. L'idea è quella di creare una piattaforma decentralizzata che consente uno scambio sicuro, veloce e trasparente dei dati clinici. Attraverso la tecnologia blockchain si propone la creazione e la gestione di un fascicolo sanitario elettronico incentrato sull'utente, in grado di mantenere un'unica versione dei suoi dati, in maniera del tutto sicura e interconnessa. Il sistema proposto sarà in grado di consentire agli utenti di dare accesso a diversi stakeholders del campo in questione, garantendogli o negandogli i permessi di lettura e scrittura del fascicolo sanitario. L'interazione utente-dato avviene in maniera affidabile e sarà essa stessa parte delle transazioni registrate sul ledger. La piattaforma si basa inoltre sulla privacy e sulla modularità garantite dal framework che ne sta alla base: Hyperledger Fabric.

3.1 Permessi e accesso

Grazie ad Hyperledger Fabric, si possono stabilire vari livelli di permessi e di accesso. In questo modo, risulta esserci più controllo all'interno della struttura e ogni utente può decidere i permessi da concedere o negare, quando farlo e per quanto tempo. Siccome la piattaforma è basata su Fabric, si tratta di un'architettura privata, in cui solo alcuni utenti possono accedere. In questo sistema, popolato da varie figure, ognuna ha il suo compito e i suoi permessi. Uno schema rappresentativo di questa situazione è quello in figura 3.1, in cui notiamo come risulta essere il paziente ad avere il controllo, avendo la libertà di decidere ed essere sempre informato riguardo ai suoi dati clinici.

Attori	Permessi
Pazienti	Permesso lettura del proprio fascicolo sanitario elettronico. Permesso scrittura e aggiornamento dei propri dati. Convalida permessi ad un medico o un'istituzione per leggere/scrivere sul FSE. Revoca permessi lettura/scrittura. Convalida permessi di lettura ad utenti di terze parti (parenti ad esempio).
Medici	Permesso di lettura e scrittura del fascicolo sanitario, ove consentito. Richiesta permesso lettura/scrittura per altri componenti.
Istituti di ricerca	Permesso di lettura del fascicolo sanitario elettronico, per estrarre dati necessari alle operazioni di ricerca e sviluppo.
Aziende assicurative	Permesso di lettura del fascicolo sanitario elettronico.
Autorità governative	Permesso di lettura del fascicolo sanitario elettronico.

Figura 3.1: Partecipanti all'interno della rete e relative autorizzazioni di accesso

Una qualsiasi persona, per poter entrare a far parte del sistema, deve recarsi all'ospedale, presso un centro per effettuare una visita, oppure registrarsi direttamente online. Sarà così possibile registrare il suo account nel sistema in modo tale da farne parte. Non tutti possono entrare in qualsiasi modo all'interno della rete però. Per entrare come medico, come istituzione o come uno degli altri stakeholders infatti, bisogna autenticarsi e mostrare la validità della propria identità.

E nel caso in cui il paziente dovesse sentirsi male, e non potesse accedere alla cartella clinica, concedendo l'autorizzazione al medico? In tal caso uno stretto parente dovrebbe disporre dell'accesso a essa, oppure il medico dovrebbe inviare una transazione di richiesta alla rete e dovrebbe essere approvata dalla maggioranza dei nodi, in modo tale da concedergli l'autorizzazione almeno di lettura della cartella. Un'altra soluzione sarebbe quella di mettere a disposizione dati strettamente necessari, consultabili in caso di sola emergenza, a cui poter fare riferimento. Per ridurre i tempi di attesa poi, nel caso di prenotazione di visite, scelta di orari e medici, si potrebbe accedere direttamente alla piattaforma online, compiendo questo tipo di scelte e gestendo tutto il proprio profilo di prenotazioni passate. La scelta di passare al digitale in maniera quasi del tutto definitiva ridurrebbe non solo gli sprechi di tempo, ma permetterebbe di eliminare tutte quelle "scartoffie" che vengono generate durante le visite, durante gli incontri, successivamente alla diagnosi in seguito a una visita.

3.2 Struttura

3.2.1 Public key infrastructure e Membership Service Provider

Tutti i partecipanti della rete, per diventare tali, necessitano di provare la propria identità in modo tale da essere capaci di effettuare transazioni all'interno del sistema e svolgere i propri compiti. Hyperledger Fabric utilizza una PKI (Public Key Infrastructure) per verificare le identità in maniera affidabile. Una PKI è proprio un insieme di processi e mezzi tecnologici che permettono di verificare e garantire l'identità di un utente. Ognuno degli attori, conserva la propria identità digitale in un certificato digitale X.509. Queste identità sono di fondamentale importanza perché permettono di stabilire le autorizzazioni esatte sulle risorse e l'accesso alle informazioni che gli attori hanno all'interno della piattaforma. La PKI è composta da una certification authority che permette di rilasciare i certificati digitali alle parti e, quando un certificato viene revocato, entra in gioco la CRL (Certificate Revocation List) che detiene la lista dei certificati non più validi. La CA quindi, ha il compito all'interno del sistema di rilasciare l'identità creando una coppia di chiavi pubbliche e private che può essere utilizzata per le operazioni crittografiche. In figura 3.2 viene mostrata l'organizzazione di un'infrastruttura a chiave pubblica e i suoi componenti.

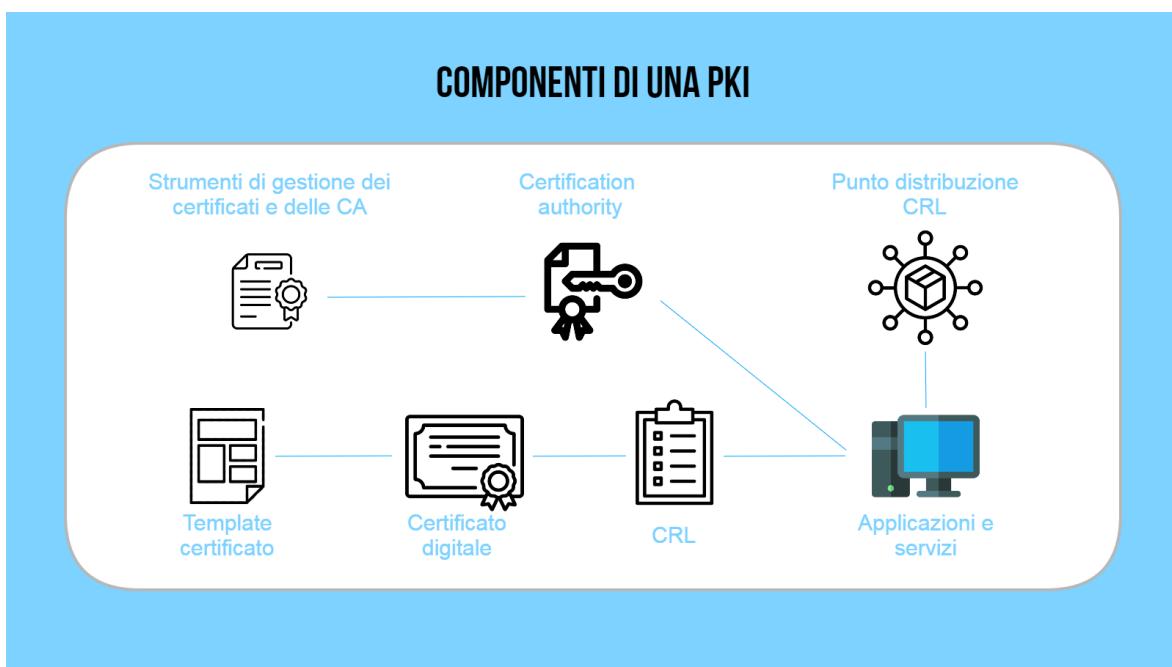


Figura 3.2: Componenti di un'infrastruttura a chiave pubblica

L'MSP (Membership Service Provider) viene utilizzato per riconoscere le varie identità all'interno della rete e per verificare che le transazioni siano effettuate da entità valide. È il Membership Service provider che ha il compito di gestire l'identità come ruolo all'interno del sistema, con i privilegi che quell'attore può possedere. La piattaforma che si intende proporre quindi, è una blockchain basata su Hyperledger Fabric, in grado di garantire gli accessi agli utenti secondo determinati schemi e modi, anche a seconda dei casi.

3.2.2 Gestione della piattaforma

La gestione dei processi segue il flusso proposto in figura 3.3, nella quale si può notare che tutto inizia a partire dal paziente. Questo si registra alla piattaforma ottenendo così un identificativo unico. Quando il paziente si reca in ospedale per effettuare una visita, i dottori svolgono tutti gli accertamenti, chiedendo al paziente accesso alla propria cartella clinica per la lettura e scrittura dei dati, che sarà salvata sulla blockchain. Attraverso un meccanismo per cui il paziente deve dare il proprio consenso e attraverso un processo del tutto crittografato, i medici ottengono le informazioni richieste e, dopo aver stilato una diagnosi, aggiornano con l'accordo del paziente i dati all'interno della cartella digitale, per poi salvarla nuovamente sulla blockchain. Gli unici ad avere accesso alla struttura per poter richiedere i dati quindi, non sono solo i medici, ma sono presenti altri stakeholders. Infatti le parti interessate posso essere varie, come gli istituti di ricerca che si occuperebbero di utilizzare i dati per scoprire nuove cure, per prevedere nuovi trend, nuove malattie. Potrebbero essere interessate anche le aziende assicurative, in caso di incidenti, in caso di lesioni e per la ricostruzione delle dinamiche dei sinistri. Una rappresentazione di questi possibili stakeholders è quella proposta in figura 3.4. Senza contare il fatto che la chiave di accesso alla cartella digitale potrebbe essere data a un parente stretto, per velocizzare i meccanismi in caso di emergenza.

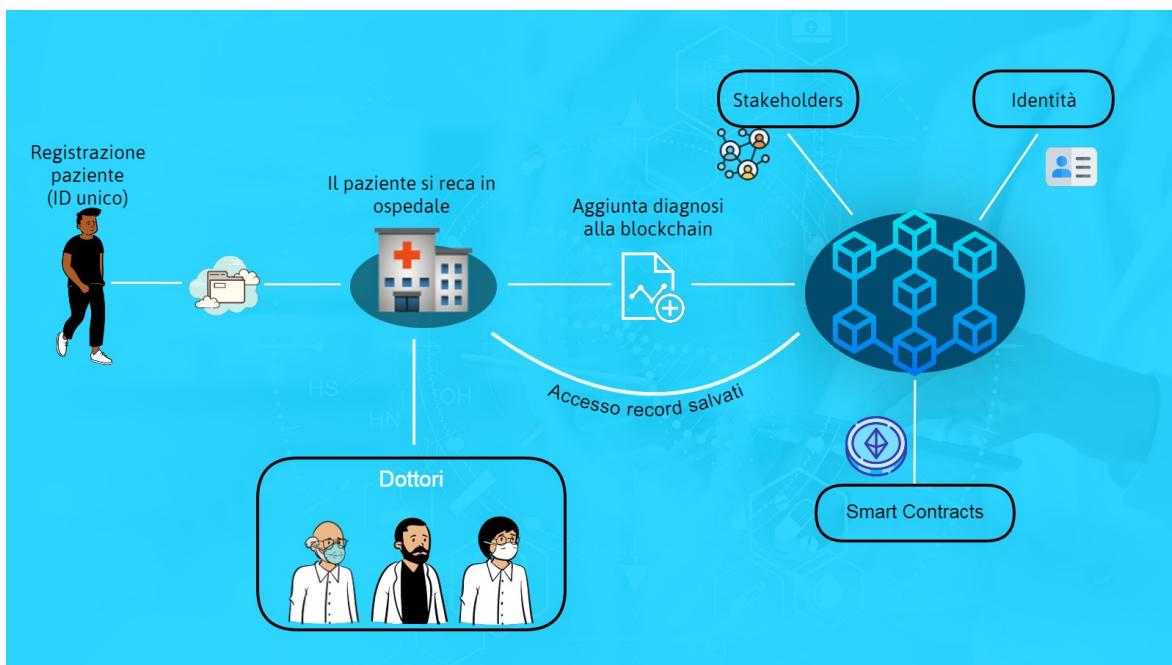


Figura 3.3: Struttura dei meccanismi di accesso e di gestione della blockchain

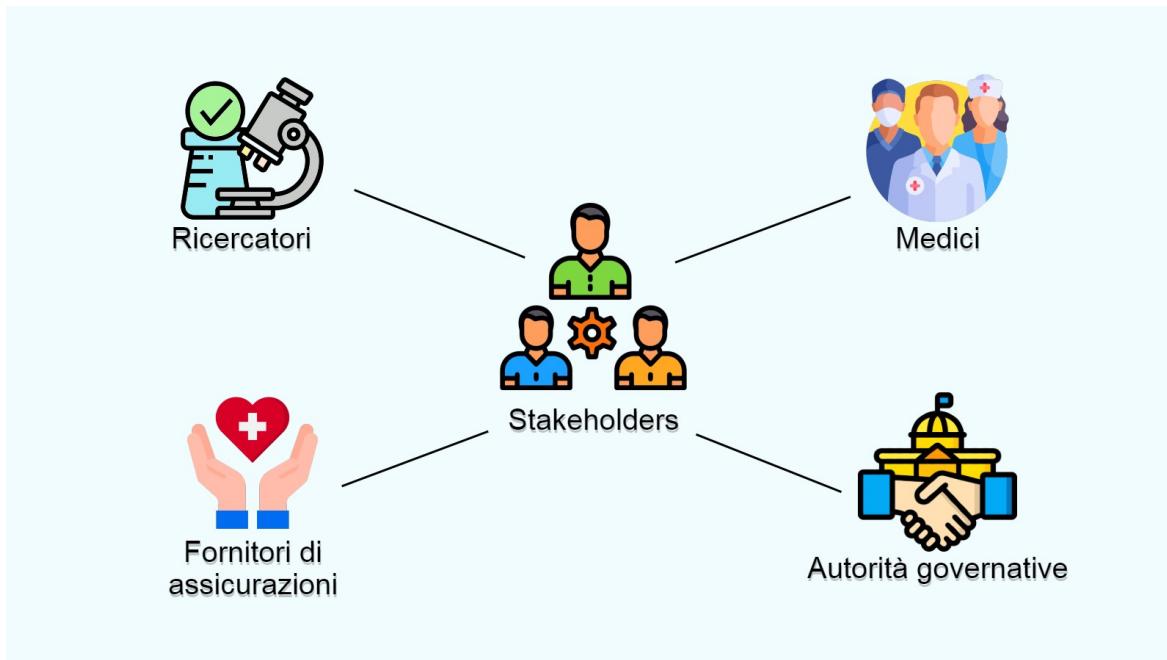


Figura 3.4: Parti interessate nell’accesso al fascicolo sanitario elettronico di un paziente

La piattaforma quindi si basa sull’utilizzo di una blockchain, che come abbiamo visto risulta essere caratterizzata dall’immutabilità dei record. È proprio questa caratteristica a far sì che i dati e le cartelle cliniche rimangano salvate nel registro, senza la possibilità che vengano manomesse o modificate. Quando un paziente si registra alla piattaforma, fornendo così tutti i dati necessari, la cartella corrispondente entra a far parte del sistema, venendo così permanentemente salvata sul ledger. Quando un medico o un professionista sanitario deve modificare dati o aggiornarli richiede l’accesso alla cartella clinica e il paziente deve consentirgli l’accesso per far sì che visualizzi i dati in questione. Dopo aver aggiornato la cartella, viene effettuata una transazione per la condivisione dei dati nella rete, e in seguito all’approvazione della transazione, viene salvata la cartella sul ledger. La cartella non potrà essere salvata nella stessa posizione che assumeva precedentemente in quanto i blocchi vengono validati, collegati e restano immodificabili. Un blocco viene così generato, seguendo i meccanismi di gestione delle transazioni proposti da Hyperledger Fabric, che seguono la logica execute-order-validate. I blocchi della blockchain contengono comunque le transazioni e ogni blocco presenta un hash del blocco precedente, per essere collegato ad esso. I blocchi presentano un header con alcune informazioni ed un body, in cui sono contenute le transazioni. Sono poi presenti informazioni come il numero del blocco, l’identificativo del validatore, l’hash del blocco e del blocco precedente.

3.2.3 Crittografia

L'organizzazione dei dati è molto importante e mantenerli sicuri è una necessità siccome stiamo trattando informazioni molto sensibili. La piattaforma utilizza anche meccanismi crittografici per assicurare l'affidabilità nella gestione dei record. In particolare, viene utilizzato un meccanismo di crittografia asimmetrica per assicurare la correttezza dei dati. Questi, non sono salvati direttamente sulla chain, ma essa funge da puntatore alla zona di memoria in cui sono conservati in maniera cifrata, in modo tale da non permettere infiltrazioni o manomissioni.

Crittografia a chiave pubblica/privata

La crittografia asimmetrica, con l'utilizzo di chiave pubblica e privata, permette lo scambio di informazioni in maniera del tutto sicura, evitando le problematiche di scambio della chiave annesse alla crittografia simmetrica. Il mittente, per mandare un messaggio, lo cifra con la chiave pubblica del destinatario, che poi sarà responsabile di decifrare il messaggio. Nel caso in cui il paziente deve garantire l'accesso alla sua cartella, questa viene decifrata con la propria chiave privata e successivamente l'identificativo del medico che ha modificato la cartella viene aggiunto all'asset, in maniera tale da testimoniare l'evento.

3.2.4 Amministrazione dei dati off-storage

Per quanto riguarda la struttura interna della blockchain, si tratta di una blockchain affiancata da uno storage off-chain, un posto in cui salvare i dati dei pazienti, le cartelle cliniche vere e proprie. Infatti, come anticipato nel capitolo precedente, il settore della sanità ricopre una fetta molto importante di tutto il mercato e applicando questo sistema si dovrebbe gestire una quantità considerevole di dati. Utilizzando un meccanismo del genere si ridurrebbe il carico dalla catena principale, in modo tale da salvare solo gli hash dei dati su di essa, rendendo le operazioni più veloci e meno onerose. I dati una volta registrati vanno a far parte dello storage off-chain. Questo archivio di dati viene gestito dal governo e dalle organizzazioni competenti. Quando i dati vengono salvati viene al contempo generato un hash del fascicolo in questione attraverso un algoritmo di hashing (si propone l'utilizzo del secure hash halgorithm 256) e il digest generato viene salvato sul ledger in via definitiva. Quando si richiede l'accesso a un fascicolo e quindi si vogliono riportare alla luce i dati, viene comparato l'hash dei dati decifrati con l'hash salvato sul ledger, per verificare che questi non siano stati compromessi. Se i due valori non corrispondono risulta esserci una compromissione e quindi non viene concesso l'accesso, altrimenti i dati vengono inviati al richiedente. Il processo appena descritto è illustrato e analizzato in figura 3.5. Quando un utente deve concedere i permessi di accesso a qualche medico oppure istituzione si attiva il meccanismo di crittografia a chiave asimmetrica che regola il sistema. Per far sì che il dato venga decriptato, l'utente deve infatti utilizzare la sua chiave privata.

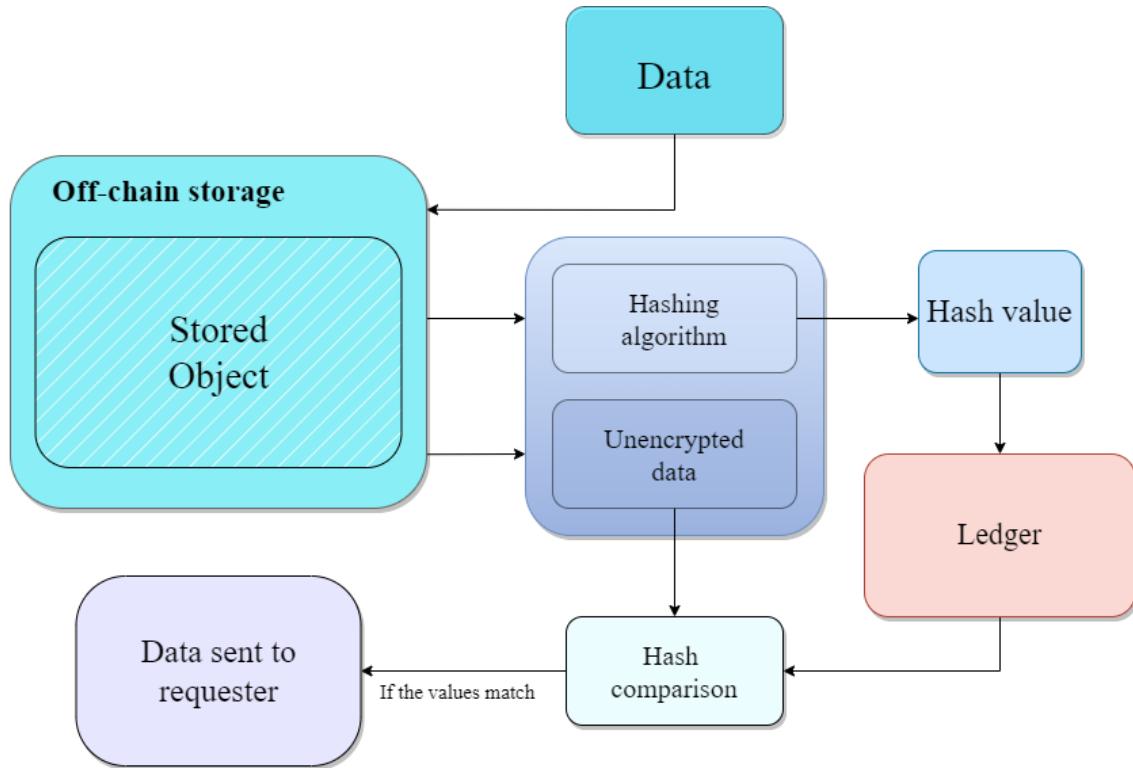


Figura 3.5: Salvataggio dei dati off-chain e meccanismo di controllo integrità dei dati

3.3 Casi d'uso

3.3.1 Interazione dei vari attori all'interno del sistema

Di seguito, in figura 3.6 viene proposto un diagramma dei casi d'uso in cui sono coinvolti i principali attori del sistema. Il paziente necessita della creazione dell'account, così successivamente può effettuare il login, visualizzare i dati e confermare le richieste di accesso da parte di medici e stakeholders. Il medico, dalla sua parte, può permettersi di visualizzare e modificare i record se concesso dal paziente, accedendo alla propria schermata dopo aver effettuato il login. Anche le varie istituzioni possono visualizzare i dati, sempre con il consenso del paziente.

3.3.2 Caso d'uso paziente

Il paziente, dopo aver effettuato il login alla piattaforma, può eseguire varie azioni. Innanzitutto può visualizzare le proprie informazioni, lo storico dei propri dati ed effettuare eventualmente delle modifiche oppure degli aggiornamenti. Il sistema, essendo incentrato sull'utente, gli permette di avere il controllo sui dati e sugli accessi, garantendogli maggiore sicurezza e consapevolezza. L'utente può poi visualizzare l'insieme dei medici che hanno richiesto l'accesso in lettura o scrittura alla propria cartella clinica e convalidare o revocare questi permessi. La figura 3.7 rappresenta questa situazione.

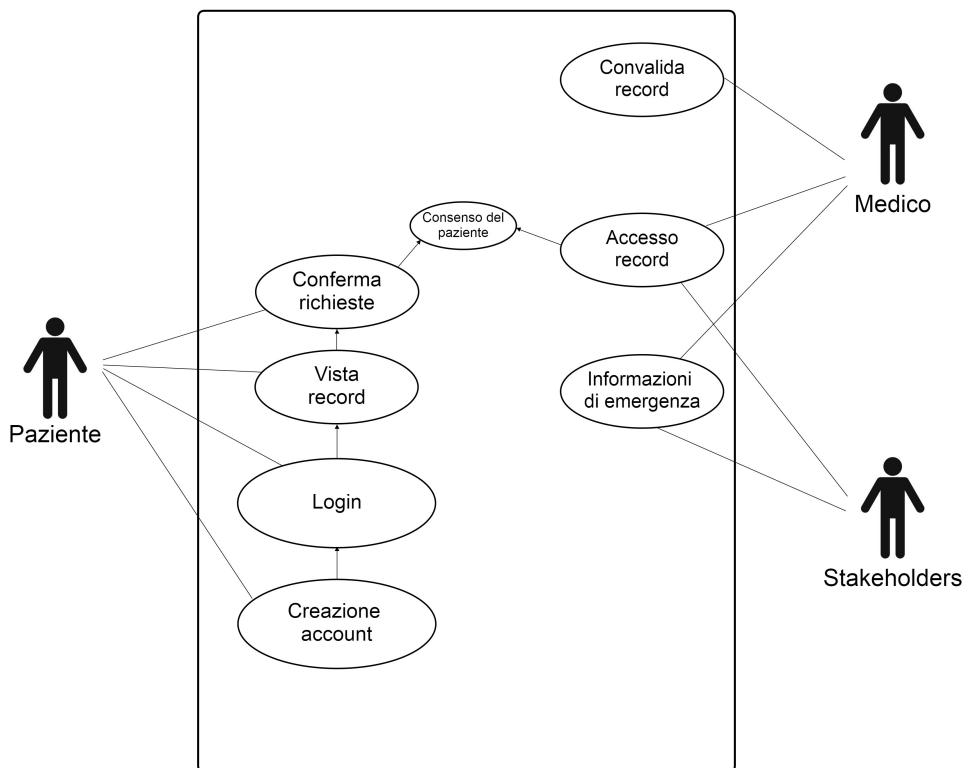


Figura 3.6: Use case diagram che coinvolge gli attori all'interno del sistema

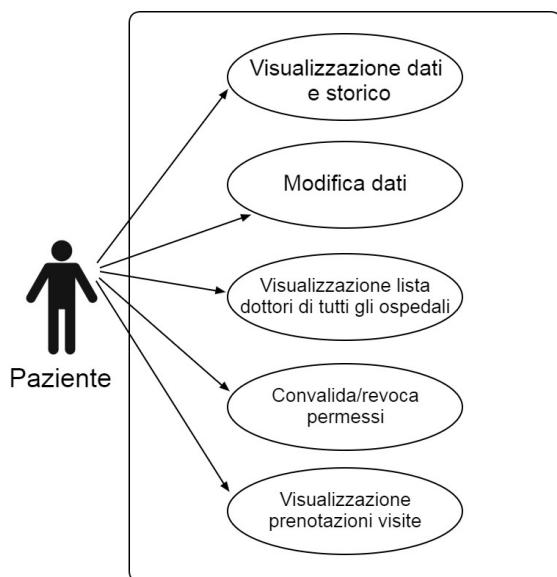


Figura 3.7: Use case diagram paziente

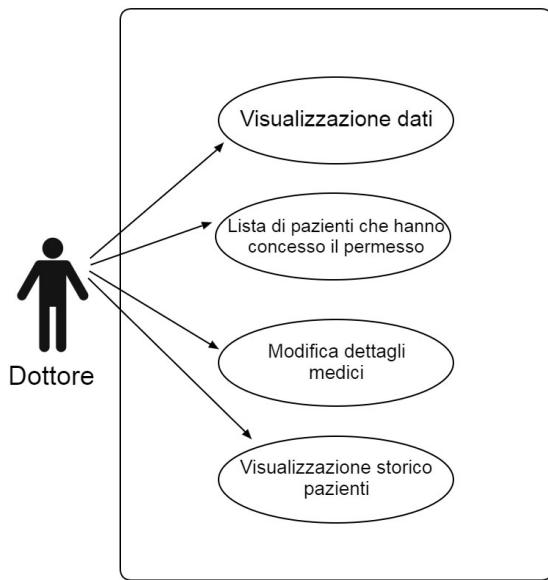


Figura 3.8: Use case diagram dottore

3.3.3 Caso d'uso dottore

Nella figura 3.8 viene mostrata l'interazione del medico con il sistema. Questo può effettuare il login dopo essersi registrato ed effettuare tutte le operazioni consentite. Per apportare modifiche alla cartella clinica di un paziente oppure per visualizzare le informazioni personali, deve essere richiesto l'accesso, che deve essere successivamente accettato dal paziente stesso. Il medico può apportare modifiche e poi salvare i dati sulla blockchain, in una zona di memoria diversa da quella precedente data l'immutabilità delle informazioni sulla catena stessa. I dottori possono comunque modificare i propri dati, visualizzare la lista dei pazienti che gli hanno concesso l'accesso e visualizzare i dati attuali e quelli inseriti precedentemente, oltre a vedere lo storico dei pazienti con rispettiva data di visita.

3.3.4 Creazione profilo utente

Per quanto riguarda la creazione dell'utente invece, questa deve avvenire una volta sola, siccome il dato rimarrà successivamente salvato all'interno del sistema. Il paziente si reca in ospedale per la creazione del proprio account, come illustrato nello schema presente in figura 3.9. Il paziente fornisce i dati al medico in questione, che provvederà ad aggiungere i dati all'interno del sistema. Successivamente i dati saranno inoltrati alla rete Fabric sottostante che provvederà in maniera automatica alla creazione dell'account, alla generazione della cartella clinica, alla fornitura della password per accedere al sistema.

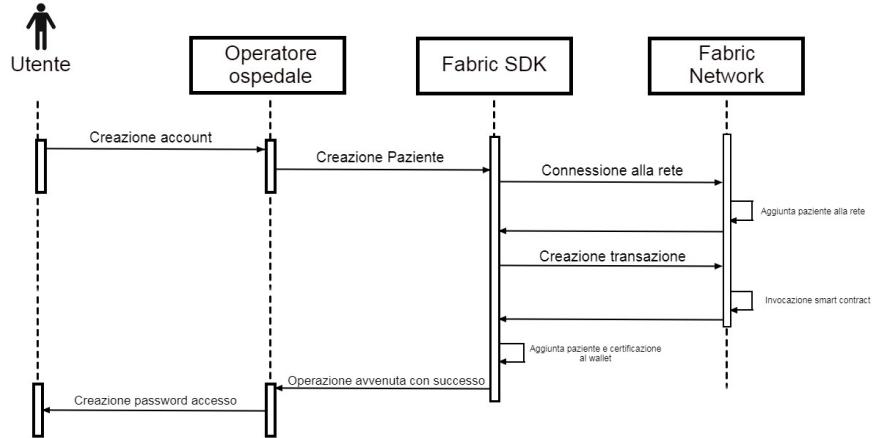


Figura 3.9: Sequence diagram che mostra l'aggiunta di un paziente all'interno del sistema

3.4 Prototipo del sistema

In questo paragrafo viene presentata l'organizzazione del sistema e la struttura delle interfacce dal punto di vista degli attori.

3.4.1 Dashboard paziente

Come anticipato in precedenza, per far sì che avvenga la registrazione di un utente alla piattaforma, questo può recarsi presso un ospedale nel quale verranno registrati i suoi dati. Viene creata quindi una transazione che permette di salvare i dati sul ledger. Quindi è necessario fornire i dati e sottometterli in una schermata come quella in figura 3.10. Una volta creato l'account, il paziente potrà effettuare il login alla piattaforma e cambiare i propri dati di accesso e le credenziali, per ragioni di sicurezza. Infatti, i dati iniziali saranno inseriti da un medico, quindi sarà richiesto il cambio della password temporanea, per rendere il sistema più sicuro. La figura 3.11 descrive questa situazione.

Inoltre il paziente ha la possibilità, oltre a visualizzare i propri dati, di garantire o revocare l'accesso ai medici che lo hanno richiesto. All'interno della piattaforma è presente una sezione apposita, descritta dalla figura 3.12, che permette di effettuare queste operazioni.

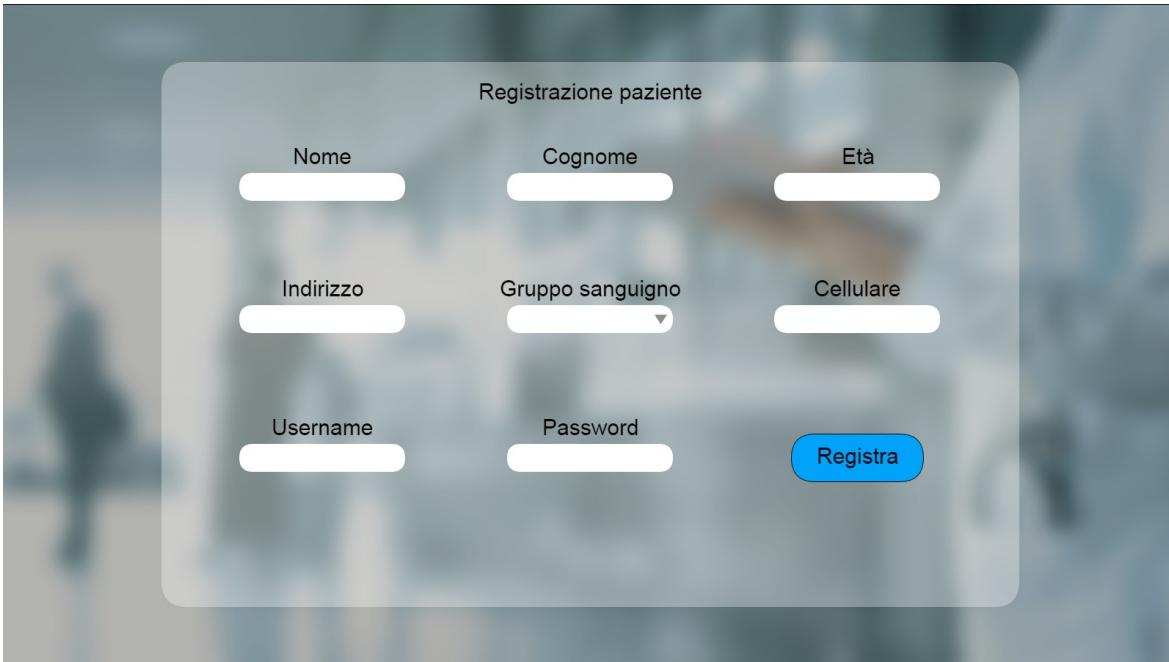


Figura 3.10: Schermata di registrazione di un utente all'interno della rete

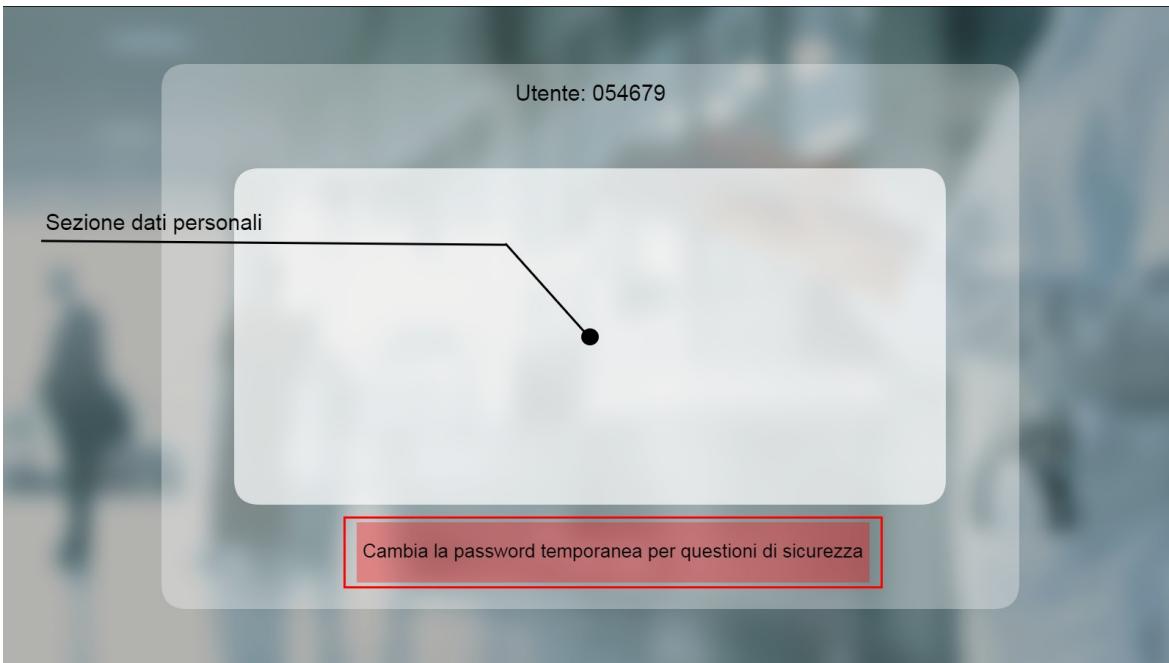


Figura 3.11: Richiesta cambio password in seguito al login da parte del paziente

3.4.2 Dashboard dottore

I medici rappresentano un'altra entità all'interno del sistema e quindi è presente una schermata di registrazione anche per loro. Questi devono fornire i propri dati per essere registrati alla piattaforma e necessitano di credenziali per l'accesso. La figura 3.13 mostra la schermata di registrazione di un medico, in cui sono presenti dati come l'ospedale in cui opera, i dati personali, il campo di specializzazione e le credenziali di accesso.

Lista medici			
Nome	Cognome	ID	Permessi
Mario	Rossi	03421	CONCEDE REVOCA
Alberto	Verdi	22564	CONCEDE REVOCA
Giovanna	Bianchi	07890	CONCEDE REVOCA

Figura 3.12: Schermata di visualizzazione di tutti i medici che hanno richiesto l'accesso ai dati

Dopo essersi registrato, il medico può effettuare il login con le proprie credenziali. Anche il paziente può effettuare il login, e la figura 3.14 rappresenta la pagina per effettuare l'accesso.

Registrazione dottore

Nome	Cognome	Età
<input type="text"/>	<input type="text"/>	<input type="text"/>
Ospedale	Specializzazione	Identificativo
<input type="text"/>	<input type="text"/>	<input type="text"/>
Username	Password	Registra
<input type="text"/>	<input type="text"/>	Registra

Figura 3.13: Schermata di registrazione di un medico all'interno della rete

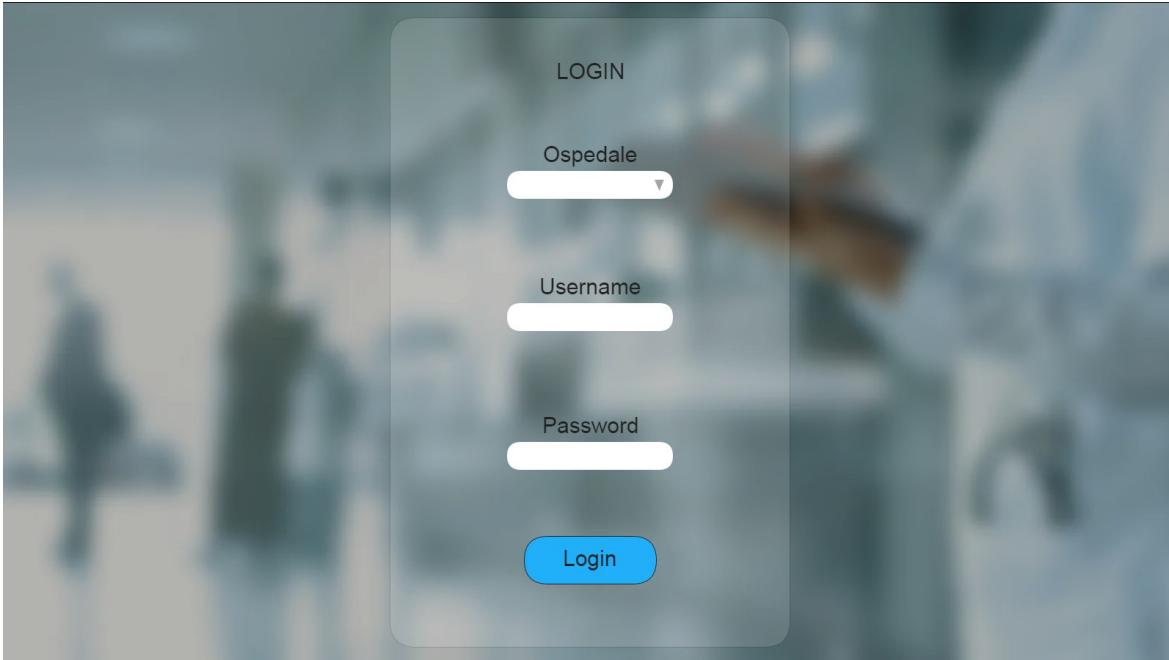


Figura 3.14: Schermata di accesso al sistema dopo aver effettuato la registrazione

3.5 Ledger

Il ledger in Hyperledger Fabric presenta una struttura come quella in figura 3.15. Si tratta di un libro mastro distribuito che comprende due parti: la blockchain vera e propria e il world state, un database contenente i valori correnti che caratterizzano lo stato attuale del ledger.

3.5.1 Blockchain e World State

Il world state permette di rendere le operazioni più semplici siccome permette di accedere direttamente al valore di uno stato senza dapprima doverlo calcolare. La blockchain invece rappresenta il registro delle transazioni e rappresenta tutti i cambiamenti che hanno portato all'ottenimento di quel preciso world state. Le transazioni sono sempre racchiuse in blocchi e distribuite sulla rete secondo il modello previsto da Fabric, descritto nel capitolo precedente. Anche qui l'header di ogni blocco include un hash delle transazioni del blocco e un hash del blocco precedente. In questo modo, tutte le transazioni sul libro mastro sono sequenziate e crittograficamente collegate tra loro. Questo hashing e collegamento rende i dati del libro mastro molto sicuri. Anche se un nodo che ospita il libro mastro fosse manomesso, non sarebbe in grado di convincere tutti gli altri nodi che ha la blockchain "corretta" perché il libro mastro è distribuito attraverso una rete di nodi indipendenti.

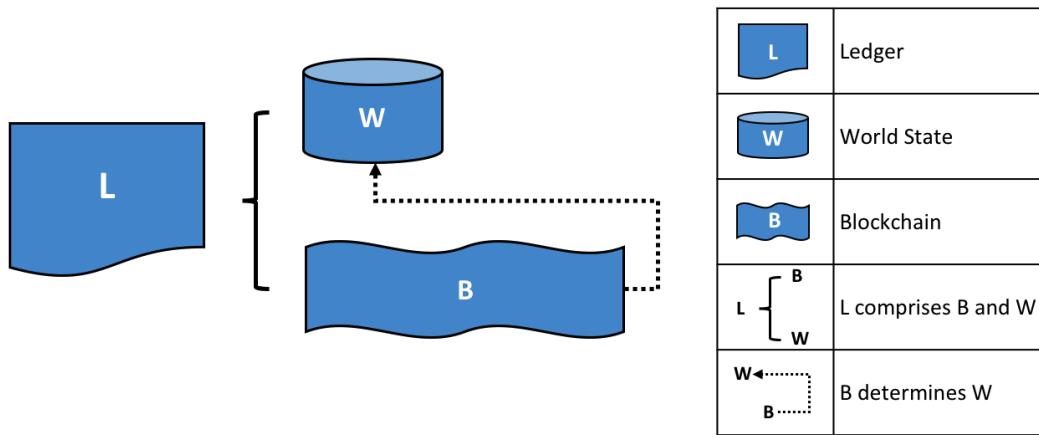


Figura 3.15: Struttura del ledger, composto da un world state e dalla blockchain

3.6 Implementazione

La piattaforma da realizzare utilizza le già note proprietà e i vantaggi della blockchain, affiancate dalla modularità e dalla privacy aggiunte da Fabric. Come in Ethereum, c’è la possibilità di inserire della logica di business all’interno dei nodi attraverso i chaincode, meccanismo chiave di Hyperledger Fabric. I dati dei pazienti sono di fondamentale importanza all’interno del sistema, e saranno i componenti delle transazioni, in modo tale da essere salvati in maniera permanente sul ledger. L’applicazione si basa sull’utilizzo di Hyperledger Fabric, con il sostegno di Docker come piattaforma per l’utilizzo di container, CouchDB come database di stato e codice in Python, Java e JavaScript. L’interfaccia utente viene scritta utilizzando Angular e la connessione tra il front end e il back end viene effettuata tramite chiamate REST.

3.6.1 Prerequisiti

Per l’esecuzione del sistema deve essere installato docker e docker-compose. I prerequisiti per l’installazione del progetto sono visibili sulla pagina dei documenti di Hyperledger Fabric [26]. Bisogna aver installato git e l’ultima versione di cURL (comando utilizzato per svolgere una serie di operazioni come effettuare richieste HTTP).

3.6.2 Fabric-samples

Per la creazione della rete, sono stati usati i samples messi a disposizione da Fabric per iniziare a lavorare con questo framework. Questi esempi permettono di esplorare importanti funzionalità della rete e di imparare a interagire con le blockchain attraverso l'utilizzo degli SDK di Fabric. Per l'utilizzo dei samples devono essere dapprima scaricate le immagini di docker e i file binari. Nella cartella degli esempi è presente una rete di test basata su docker costituita da due peer dell'organizzazione e un nodo del servizio di ordering. In figura 3.16 possiamo vedere quali cartelle compongono i samples e la presenza della test network.

```
> asset-transfer-basic
> asset-transfer-events
> asset-transfer-ledger-queries
> asset-transfer-private-data
> asset-transfer-sbe
> asset-transfer-secured-agreement
> auction
> bin
> chaincode
> ci
> client
> commercial-paper
> config
> fabcar
> first-network
> high-throughput
> interest_rate_swaps
> off_chain_data
> patient-asset-transfer
> private-collections
> scripts
> server
> test-application
> test-network
> token-erc-20
> token-utxo
```

Figura 3.16: Struttura delle cartelle nei Fabric samples

La prima serie di cartelle (asset-transfer) si occupa di fornire una serie di smart contract per mostrare come trasferire asset utilizzando la rete di Fabric. Ci sono diversi esempi a riguardo e ognuno di essi mostra una diversa funzionalità e un diverso modo di agire attraverso la rete.

- Basic: Questo sample è consigliato per i nuovi utenti e consente di creare e trasferire un asset inserendo i dati nel libro mastro e recuperandoli.
- Events: Viene mostrato come i contratti possono generare eventi che interagiscono con l'applicazione.

- Transfer Ledger Queries: Esempi di interrogazioni al libro mastro e aggiornamenti effettuati su di esso.
- Secured Agreement: Vengono mostrate le politiche di approvazione e le possibili configurazioni per la sovrascrizione di queste politiche.

Il resto delle cartelle mostra la possibilità di creazione di un token attraverso uno smart contract, la possibilità di creare uno storage off-chain per la gestione dei dati e viene fornita la rete di test.

3.6.3 CouchDB

Per il funzionamento della rete viene utilizzato CouchDB come database di stato per Fabric. Questa piattaforma supporta due tipi di database di stato: LevelDB (quello predefinito) archivia i dati dei chaincode come coppie chiave-valore; CouchDB è un database facoltativo che consente di modellare i dati sul libro mastro come JSON ed eseguire query complesse sui valori dei dati anziché sulle chiavi. Il supporto di CouchDB ti consente inoltre di distribuire indici con il tuo chaincode per rendere le query più efficienti, in modo tale da essere eseguite su set di dati di grandi dimensioni.

3.6.4 Rete ad alto throughput

Il throughput non è altro che la frequenza con la quale vengono trasmessi i dati. Controllare e misurare questo fattore all'interno di una rete risulta essere di fondamentale importanza in quanto fornisce informazioni sulle prestazioni della stessa. L'obiettivo è quello di fornire un sistema con un alto throughput per la gestione di migliaia di transazioni al secondo che possono aggiornare tutte lo stesso asset all'interno del libro mastro. Quando molte transazioni arrivano contemporaneamente, nel periodo in cui una transazione viene simulata sul peer ed è pronta per essere impegnata nel libro mastro, un'altra potrebbe già aver modificato quel valore. Per risolvere questo problema il valore viene aggiornato di frequente e non viene letta una singola riga, ma viene considerato un insieme di righe. All'interno della cartella high-throughput nel progetto si trova il file high-throughput.go che mostra proprio come gestire una grande quantità di dati in presenza di molte transazioni. In particolare vengono fornite quattro funzioni. In figura 3.17 è mostrata la funzione di update presente all'interno dello smart contract che permette di aggiornare il libro mastro per includere un nuovo valore delta per una particolare variabile. Se viene aggiunta per la prima volta, assume un valore pari a zero. In figura 3.18 è mostrata la funzione di get, che ha l'obiettivo di recuperare i valori all'interno del libro mastro. La funzione prune in figura 3.19 invece è una funzione che taglia tutte le righe di una variabile considerata mentre ne calcola il valore finale. Una volta che tutte le righe sono state elaborate ed eliminate, viene aggiunta una singola nuova riga che definisce un delta contenente il valore finale della variabile. In figura 3.20 la funzione di delete elimina dal libro mastro tutte le righe associate a una variabile aggregata.

Queste funzioni vengono poi richiamate dalla funzione invoke che ha il compito di indirizzare le invocazioni alla funzione appropriata attraverso una serie di statement if-else.

```

94 func (s *SmartContract) update(APIstub shim.ChaincodeStubInterface, args []string) pb.Response {
95     // Check we have a valid number of args
96     if len(args) != 3 {
97         return shim.Error("Incorrect number of arguments, expecting 3")
98     }
99
100    // Extract the args
101    name := args[0]
102    op := args[2]
103    _, err := strconv.ParseFloat(args[1], 64)
104    if err != nil {
105        return shim.Error("Provided value was not a number")
106    }
107
108    // Make sure a valid operator is provided
109    if op != "+" && op != "-" {
110        return shim.Error(fmt.Sprintf("Operator %s is unrecognized", op))
111    }
112
113    // Retrieve info needed for the update procedure
114    txid := APIstub.GetTxID()
115    compositeIndexName := "varName~op~value~txID"
116
117    // Create the composite key that will allow us to query for all deltas on a particular variable
118    compositeKey, compositeErr := APIstub.CreateCompositeKey(compositeIndexName, []string{name, op, args[1], txid})
119    if compositeErr != nil {
120        return shim.Error(fmt.Sprintf("Could not create a composite key for %s: %s", name, compositeErr.Error()))
121    }
122
123    // Save the composite key index
124    compositePutErr := APIstub.PutState(compositeKey, []byte{0x00})
125    if compositePutErr != nil {
126        return shim.Error(fmt.Sprintf("Could not put operation for %s in the ledger: %s", name, compositePutErr.Error()))
127    }
128
129    return shim.Success([]byte(fmt.Sprintf("Successfully added %s% to %s", op, args[1], name)))
130 }

```

Figura 3.17: Funzione di aggiornamento all'interno del file high-throughput.go

```

143 func (s *SmartContract) get(APIstub shim.ChaincodeStubInterface, args []string) pb.Response {
144     // Check we have a valid number of args
145     if len(args) != 1 {
146         return shim.Error("Incorrect number of arguments, expecting 1")
147     }
148     name := args[0]
149     // Get all deltas for the variable
150     deltaResultsIterator, deltaErr := APIstub.GetStateByPartialCompositeKey("varName~op~value~txID", []string{name})
151     if deltaErr != nil {
152         return shim.Error(fmt.Sprintf("Could not retrieve value for %s: %s", name, deltaErr.Error()))
153     }
154     defer deltaResultsIterator.Close()
155     // Check the variable existed
156     if !deltaResultsIterator.HasNext() {
157         return shim.Error(fmt.Sprintf("No variable by the name %s exists", name))
158     }
159     // Iterate through result set and compute final value
160     var finalVal float64
161     var i int
162     for i = 0; deltaResultsIterator.HasNext(); i++ {
163         // Get the next row
164         responseRange, nextErr := deltaResultsIterator.Next()
165         if nextErr != nil {
166             return shim.Error(nextErr.Error())
167         }
168         // Split the composite key into its component parts
169         _, keyParts, splitKeyErr := APIstub.SplitCompositeKey(responseRange.Key)
170         if splitKeyErr != nil {
171             return shim.Error(splitKeyErr.Error())
172         }
173         // Retrieve the delta value and operation
174         operation := keyParts[1]
175         valueStr := keyParts[2]
176         // Convert the value string and perform the operation
177         value, convErr := strconv.ParseFloat(valueStr, 64)
178         if convErr != nil {
179             return shim.Error(convErr.Error())
180         }
181
182         switch operation {
183         case "+":
184             finalVal += value
185         case "-":
186             finalVal -= value
187         default:
188             return shim.Error(fmt.Sprintf("Unrecognized operation %s", operation))
189     }
190
191     return shim.Success([]byte(strconv.FormatFloat(finalVal, 'f', -1, 64)))
192 }
```

Figura 3.18: Funzione di prelievo all'interno del file high-throughput.go

```

205 func (s *SmartContract) prune(APIstub shim.ChaincodeStubInterface, args []string) pb.Response {
206     // Check we have a valid number of args
207     if len(args) != 1 {
208         return shim.Error("Incorrect number of arguments, expecting 1")
209     }
210     // Retrieve the name of the variable to prune
211     name := args[0]
212     // Get all delta rows for the variable
213     deltaResultsIterator, deltaErr := APIstub.GetStateByPartialCompositeKey("varName~op~value~txID", []string{name})
214     if deltaErr != nil {
215         return shim.Error(fmt.Sprintf("Could not retrieve value for %s: %s", name, deltaErr.Error()))
216     }
217     defer deltaResultsIterator.Close()
218     // Check the variable existed
219     if !deltaResultsIterator.HasNext() {
220         return shim.Error(fmt.Sprintf("No variable by the name %s exists", name))
221     }
222     // Iterate through result set computing final value while iterating and deleting each key
223     var finalVal float64
224     var i int
225     for i = 0; deltaResultsIterator.HasNext(); i++ {
226         // Get the next row
227         responseRange, nextErr := deltaResultsIterator.Next()
228         if nextErr != nil {
229             return shim.Error(nextErr.Error())
230         }
231         // Split the key into its composite parts
232         _, keyParts, splitKeyErr := APIstub.SplitCompositeKey(responseRange.Key)
233         if splitKeyErr != nil {
234             return shim.Error(splitKeyErr.Error())
235         }
236         // Retrieve the operation and value
237         operation := keyParts[1]
238         valueStr := keyParts[2]
239         // Convert the value to a float
240         value, convErr := strconv.ParseFloat(valueStr, 64)
241         if convErr != nil {
242             return shim.Error(convErr.Error())
243         }
244         // Delete the row from the ledger
245         deltaRowDelErr := APIstub.Delstate(responseRange.Key)
246         if deltaRowDelErr != nil {
247             return shim.Error(fmt.Sprintf("Could not delete delta row: %s", deltaRowDelErr.Error()))
248         }
249         // Add the value of the deleted row to the final aggregate
250         switch operation {
251             case "+":
252                 finalVal += value
253             case "-":
254                 finalVal -= value
255             default:
256                 return shim.Error(fmt.Sprintf("Unrecognized operation %s", operation))
257         }
258     }
259     // Update the ledger with the final value
260     updateResp := s.update(APIstub, []string{name, strconv.FormatFloat(finalVal, 'f', -1, 64), "+"})
261     if updateResp.Status == ERROR {
262         return shim.Error(fmt.Sprintf("Could not update the final value of the variable after pruning: %s", updateResp.Message))
263     }
264     return shim.Success([]byte(fmt.Sprintf("Successfully pruned variable %s, final value is %f, %d rows pruned", args[0], finalval, i)))
265 }
```

Figura 3.19: Funzione di taglio all'interno del file high-throughput.go

```

277 func (s *SmartContract) delete(APIstub shim.ChaincodeStubInterface, args []string) pb.Response {
278     // Check there are a correct number of arguments
279     if len(args) != 1 {
280         return shim.Error("Incorrect number of arguments, expecting 1")
281     }
282
283     // Retrieve the variable name
284     name := args[0]
285
286     // Delete all delta rows
287     deltaResultsIterator, deltaErr := APIstub.GetStateByPartialCompositeKey("varName~op~value~txID", []string{name})
288     if deltaErr != nil {
289         return shim.Error(fmt.Sprintf("Could not retrieve delta rows for %s: %s", name, deltaErr.Error()))
290     }
291     defer deltaResultsIterator.Close()
292
293     // Ensure the variable exists
294     if !deltaResultsIterator.HasNext() {
295         return shim.Error(fmt.Sprintf("No variable by the name %s exists", name))
296     }
297
298     // Iterate through result set and delete all indices
299     var i int
300     for i = 0; deltaResultsIterator.HasNext(); i++ {
301         responseRange, nextErr := deltaResultsIterator.Next()
302         if nextErr != nil {
303             return shim.Error(fmt.Sprintf("Could not retrieve next delta row: %s", nextErr.Error()))
304         }
305
306         deltaRowDelErr := APIstub.DelState(responseRange.Key)
307         if deltaRowDelErr != nil {
308             return shim.Error(fmt.Sprintf("Could not delete delta row: %s", deltaRowDelErr.Error()))
309         }
310     }
311
312     return shim.Success([]byte(fmt.Sprintf("Deleted %s, %d rows removed", name, i)))
313 }
314

```

Figura 3.20: Funzione di cancellazione all'interno del file high-throughput.go

3.6.5 Creazione dei vari componenti all'interno del sistema

Ospedali

Per la creazione delle due organizzazioni come ospedali all'interno della rete si fa riferimento alla figura 3.21

```

13 // @description Crea un profilo di connessione e restituisce la configurazione di rete all'ospedale 1
14
15 exports.buildCCPHosp1 = () => {
16   // load the common connection configuration file
17   const ccpPath = path.resolve(__dirname, '..', '..', 'first-network',
18     | 'organizations', 'peerOrganizations', 'hosp1.lithium.com', 'connection-hosp1.json');
19   const fileExists = fs.existsSync(ccpPath);
20   if (!fileExists) {
21     | throw new Error(`no such file or directory: ${ccpPath}`);
22   }
23   const contents = fs.readFileSync(ccpPath, 'utf8');

24
25   // build a JSON object from the file contents
26   const ccp = JSON.parse(contents);

27
28   console.log(`Loaded the network configuration located at ${ccpPath}`);
29   return ccp;
30 };
31
32 // @description Crea un profilo di connessione e restituisce la configurazione di rete all'ospedale 2
33
34 exports.buildCCPHosp2 = () => {
35   // load the common connection configuration file
36   const ccpPath = path.resolve(__dirname, '..', '..', 'first-network',
37     | 'organizations', 'peerOrganizations', 'hosp2.lithium.com', 'connection-hosp2.json');
38   const fileExists = fs.existsSync(ccpPath);
39   if (!fileExists) {
40     | throw new Error(`no such file or directory: ${ccpPath}`);
41   }
42   const contents = fs.readFileSync(ccpPath, 'utf8');

43
44   // build a JSON object from the file contents
45   const ccp = JSON.parse(contents);

46
47   console.log(`Loaded the network configuration located at ${ccpPath}`);
48   return ccp;
49 };

```

Figura 3.21: Creazione dei profili di connessione della rete per gli ospedali

3.6.6 Pazienti

In questa sezione viene mostrata la creazione di un utente e più in particolare in figura 3.22 e 3.23 viene raffigurata la metodologia per la creazione dell'utente, con conseguente iscrizione all'organizzazione e aggiunta al wallet.

```

75 exports.registerAndEnrollUser = async (caClient, wallet, orgMspId, userId, adminUserId, attributes, affiliation) => {
76   try {
77     // Controllo registrazione utente già fatta
78     const userIdentity = await wallet.get(userId);
79     if (userIdentity) {
80       console.log(`An identity for the user ${userId} already exists in the wallet`);
81       throw new Error(`An identity for the user ${userId} already exists in the wallet`);
82     }
83
84     //Registrazione utente
85     const adminIdentity = await wallet.get(adminUserId);
86     if (!adminIdentity) {
87       console.log(`An identity for the admin user ${adminUserId} does not exist in the wallet`);
88       throw new Error(`An identity for the admin user ${adminUserId} does not exist in the wallet`);
89     }
90
91     const provider = wallet.getProviderRegistry().getProvider(adminIdentity.type);
92     const adminUser = await provider.getUserContext(adminIdentity, adminUserId);
93
94     attributes = JSON.parse(attributes);
95     const firstName = attributes.firstName;
96     const lastName = attributes.lastName;
97     const role = attributes.role;
98     const speciality = (role === 'doctor') ? attributes.speciality : '';
99     const secret = await caClient.register({
100       affiliation: affiliation,
101       enrollmentID: userId,
102     });

```

Figura 3.22: Creazione dei profili degli utenti

```

103 // Deve accedere il cliente, gli altri ruoli non sono ammessi
104   role: 'client',
105   attrs: [
106     {
107       name: 'firstName',
108       value: firstName,
109       ecert: true,
110     },
111     {
112       name: 'lastName',
113       value: lastName,
114       ecert: true,
115     },
116     {
117       name: 'role',
118       value: role,
119       ecert: true,
120     },
121     {
122       name: 'speciality',
123       value: speciality,
124       ecert: true,
125     },
126   }, adminUser);
127 const enrollment = await caClient.enroll({
128   enrollmentID: userId,
129   enrollmentSecret: secret,
130   attrs: [
131     {
132       name: 'firstName',
133       value: firstName,
134       ecert: true,
135     },
136     {
137       name: 'lastName',
138       value: lastName,
139       ecert: true,
140     },
141     {
142       name: 'role',
143       value: role,
144       ecert: true,
145     },
146     {
147       name: 'speciality',
148       value: speciality,
149       ecert: true,
150     },
151   },
152   credentials: {
153     certificate: enrollment.certificate,
154     privateKey: enrollment.key.toBytes(),
155   },
156   mspId: orgMspId,
157   type: 'X.509',
158 };
159 await wallet.put(userId, x509Identity);
160 console.log(`Successfully registered and enrolled user ${userId} and imported it into the wallet`);
161 } catch (error) {
162   console.error(`Failed to register user ${userId} : ${error}`);
163   throw new Error(`Failed to register user ${userId}`);
164 }

```

Figura 3.23: Registrazione dei dati dei clienti (esclusivamente il ruolo di paziente)

Capitolo 4

Conclusioni

Il costo dell’assistenza sanitaria aumenta ogni anno a causa delle elevate esigenze di prevenzione e diagnosi sanitaria, esigenze insoddisfatte dei pazienti, aumento della popolazione anziana, costi di sviluppo di nuovi farmaci, aumento delle malattie croniche e altri fattori. Le industrie farmaceutiche e sanitarie sono complesse. Queste industrie hanno bisogno della collaborazione di più parti interessate per conformarsi ad ambienti altamente regolamentati che cercano in ultima analisi di garantire la sicurezza dei pazienti. Tutti gli aspetti della produzione di nuovi farmaci, inclusi, tra l’altro, studi clinici, produzione e distribuzione, hanno rigide linee guida normative ed etiche a cui entrambe le industrie devono attenersi. La complessità e la mancanza di trasparenza, coordinamento e fiducia sono grandi sfide in questo contesto. È qui che la blockchain potrebbe fornire soluzioni riducendo i costi dell’assistenza sanitaria, i processi di sviluppo dei farmaci e i tempi di consegna dei farmaci ai pazienti con bisogni insoddisfatti.

4.1 Innovazione apportata dalla blockchain

Sono sempre più necessari dati decentralizzati in grado di tenere traccia della storia clinica del paziente, di monitorare i suoi parametri vitali in tempo reale, di trasmettere le informazioni in modo sicuro ed evitare manipolazioni. Queste sono caratteristiche capaci di riformare l’assistenza sanitaria nel nostro paese e in tutto il resto del mondo. La blockchain è una tecnologia in grado di garantire tutti questi aspetti permettendo di trarne il meglio. Sarà possibile riformare vari aspetti del settore sanitario e non solo, grazie all’utilizzo di sistemi sempre più innovativi e al passo con i tempi, in grado di soddisfare le esigenze richieste.

Nel corso di questa tesi sono stati trattati vari argomenti riguardanti la tecnologia blockchain, passando dalla sua struttura, dalle sue caratteristiche fino ad arrivare alle possibili vulnerabilità. Sono state presentate le transazioni e i blocchi e ne è stato illustrato il funzionamento. È stata scelta una soluzione permissioned per i vantaggi apportati da tale meccanismo, compresa la sicurezza e la privacy garantite. È stato analizzato il framework per lo sviluppo di blockchain Hyperledger Fabric, perfetto per le aziende e per i contesti in cui è necessario un certo grado di riservatezza. Questo ha consentito di lavorare in un ambiente chiuso, caratterizzato dalla forte modularità e personalizzazione. La struttura pensata, caratterizzata da uno storage off-chain, permette di alleggerire le operazioni per la modifica dei dati, rendendo

i processi più veloci. La soluzione presentata, permissioned e caratterizzata da un'amministrazione dei dati in uno storage gestito dallo stato, presenta alcuni inconvenienti. Infatti, realizzando questa piattaforma, si rinuncia in parte alla decentralizzazione caratteristica della blockchain, facendo prevalere le proprietà di performance e facilità di accesso. Una soluzione permissionless infatti permetterebbe una decentralizzazione assoluta, di fondamentale importanza per la distribuzione dei dati, e come abbiamo visto garantirebbe anche un adeguato grado di sicurezza, fornito dai membri stessi della rete. Un altro vantaggio del sistema, sostenuto da Fabric, consiste nel non utilizzare la proof of work come meccanismo di consenso. Questo permette di non sprecare risorse ed energie nel calcolo dei valori per la validazione dei blocchi, rendendo la rete più efficiente. Il progetto Pharmaledger è stato uno dei primi a porre le basi di innovazione e trasformazione nel settore sanitario attraverso l'introduzione della blockchain. Il progetto proposto in questa tesi rispetto al prototipo di Pharmaledger si avvale della forte modularità garantita da Hyperledger Fabric, caratteristica che consente una consistente modifica dei parametri in gioco e un'ampia personalizzazione. Inoltre il progetto proposto si fa forte della privacy e della riservatezza garantite dall'utilizzo di una soluzione permissioned invece che una permissionless, che sembra essere ideale in un campo in cui i dati sono di fondamentale importanza e necessitano del massimo della riservatezza.

4.2 Trend futuri

L'innovazione rappresentata dalla blockchain si muove sempre di più in questo senso, proponendo di ampliare e migliorare settori che presentano problemi di fondamentale importanza. All'interno del settore sanitario la tecnologia rappresenta un'innovazione capace di arrecare molti benefici, ma ci sono tantissime possibili applicazioni. Sono sempre di più le società e le organizzazioni che investono in tale proposito, cercando di creare una rete interconnessa di nodi che mantengono la fiducia tra di essi nello scambio di informazioni spesso sensibili. Ci sarà quindi sempre un maggiore investimento nella tecnologia e aumenteranno sempre di più le applicazioni, sperando di arrivare a una condizione caratterizzata dalla massima efficienza possibile, in termini di tempo, di spazio e di sicurezza.

Bibliografia

- [1] Roberto Garavaglia. *Conoscere la Blockchain*. HOEPLI EDITORE, 2021.
- [2] *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008. URL: <https://bitcoin.org/bitcoin.pdf>.
- [3] *La storia del Bitcoin. Come nasce il Bitcoin*. 2006. URL: <https://cryptonomist.ch/2022/05/15/storia-bitcoin-2/>.
- [4] *Various provisions relating to the use of blockchain technology*. 2017. URL: <https://www.leg.state.nv.us/App/NELIS/REL/79th2017/Bill/5463/Overview>.
- [5] Ozgur Dedeayir e Martin Steinert. “The hype cycle model: A review and future directions”. In: *Technological Forecasting and Social Change* 108 (2016), pp. 28–41.
- [6] Usman W Chohan. “Crypto Winters”. In: *Available at SSRN 4142885* (2022).
- [7] *Tecnologie basate su registri distribuiti e smart contract. Pubblicata la legge*. 14 Feb. 2019. URL: <https://poloarchivistico.regione.emilia-romagna.it/news-in-evidenza/tecnologie-basate-su-registri-distribuiti-e-smart-contract-pubblicata-la-legge>.
- [8] Maria Letizia Perugini. *Distributed ledger technologies e sistemi di Blockchain: digital currency, smart contract e altre applicazioni*. Vol. 3. Key Editore, 2018.
- [9] Scarano Vittorio. *Programmazione con Oggetti Distribuiti: Java RMI*. 2010.
- [10] *Un vademecum sulle tecnologie blockchain: quando, quale e come*. Vol. 21. DOI: 10.1109/COMST.2019.2928178.
- [11] Christophe Schinckus. “Proof-of-work based blockchain technology and Anthropocene: An undermined situation?” In: *Renewable and Sustainable Energy Reviews* 152 (2021), p. 111682.
- [12] Wenting Li et al. “Securing proof-of-stake blockchain protocols”. In: *Data privacy management, cryptocurrencies and blockchain technology*. Springer, 2017, pp. 297–315.
- [13] Kyle Croman et al. “On scaling decentralized blockchains”. In: *International conference on financial cryptography and data security*. Springer. 2016, pp. 106–125.
- [14] Amritraj Singh et al. “Sidechain technologies in blockchain networks: An examination and state-of-the-art review”. In: *Journal of Network and Computer Applications* 149 (2020), p. 102471.

- [15] Davide Aliffi e Camilla Valmorra. “FUNZIONI HASH E SICUREZZA CRITTOGRAFICA”. In: () .
- [16] Gabriele D’angelo, Stefano Ferretti e Luca Serena. “SIMULAZIONE DI ATTACCHI DI SICUREZZA SU TECNOLOGIE BLOCKCHAIN”. In: () .
- [17] Congcong Ye et al. “Analysis of security in blockchain: Case study in 51%-attack detecting”. In: *2018 5th International conference on dependable systems and their applications (DSA)*. IEEE. 2018, pp. 15–24.
- [18] Marcella Atzori. “Blockchain technology and decentralized governance: Is the state still necessary?” In: *Available at SSRN 2709713* (2015).
- [19] *Personale del Servizio sanitario nazionale, i dati 2017*. URL: https://www.salute.gov.it/portale/news/p3_2_1_1_1.jsp?lingua=italiano&menu=notizie&p=dalministero&id=3845.
- [20] *Ricorso a procedure negoziate senza previa pubblicazione di un bando nel caso di forniture e servizi ritenuti infungibili*. URL: <https://www.anticorruzione.it/-/delibera%C2%A0numero-950-del-13/09/2017-1>.
- [21] *Rapporto Euro health Consumer Index, 2018*. URL: <https://healthpowerhouse.com/media/EHCI-2018/EHCI-2018-report.pdf>.
- [22] *Conseguiti i 45 traguardi e obiettivi PNRR per il primo semestre*. URL: https://www.mef.gov.it/ufficio-stampa/comunicati/2022/documenti/comunicato_0126.pdf.
- [23] *How blockchain will transform business and the economy*. URL: <https://www.pwc.com/gx/en/industries/technology/publications/blockchain-report-transform-business-economy.html>.
- [24] Elli Androulaki et al. “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains”. In: *CoRR* abs/1801.10228 (2018). arXiv: 1801 . 10228. URL: <http://arxiv.org/abs/1801.10228>.
- [25] *Hyperledger Fabric documents. applications and peers*. URL: <https://hyperledger-fabric.readthedocs.io/en/release-1.2/peers/peers.html?highlight=network%20policy>.
- [26] *Hyperledger Fabric prerequisites installation*. URL: <https://hyperledger-fabric.readthedocs.io/en/release-2.2/prereqs.html#prerequisites>.

Ringraziamenti

Voglio innanzitutto ringraziare per il raggiungimento di questo traguardo la professoressa Genoveffa Tortora, che mi ha permesso di approfondire un tema così interessante e stimolante. Ringrazio il dottor Attilio della Greca per avermi supportato durante il lavoro svolto e per la disponibilità mostrata nei miei confronti.

Ringrazio infinitamente la mia famiglia e le persone a me care, in particolar modo mia madre e mio padre, mi sono stati sempre accanto e mi hanno supportato economicamente e moralmente durante tutto il mio percorso, incoraggiandomi sempre e credendo in me. Questa tesi è il frutto del vostro amore e della vostra dedizione, e spero che vi renda orgogliosi. Vi voglio bene, sempre. Ringrazio mio fratello Fabrizio per la spensieratezza e la gioia che mi ha trasmesso nei momenti di difficoltà, per avermi incoraggiato e per aver creduto sempre in me e mio fratello Pasquale per avermi trasmesso la curiosità, la voglia di conoscere e superare i miei limiti. Ringrazio anche tutte le persone che mi hanno mostrato affetto e si sono interessate al mio percorso.

Ringrazio i miei compagni di corso, insieme abbiamo trascorso un'avventura indimenticabile, fatta di tanti momenti divertenti e anche impegnativi. Abbiamo condiviso lo studio, le gioie e i timori fino ad arrivare al termine di questo meraviglioso percorso. Il nostro legame si è rafforzato sempre di più in questi anni di università rendendoci un gruppo unito, sono davvero molto felice di questo, vi voglio bene.

Ringrazio la mia ragazza, Anastasia, che mi è stata sempre vicino in tutti i momenti. Ho sempre potuto contare sul tuo appoggio, e questo mi ha dato la forza per andare avanti e non mollare mai. Ti ringrazio per la fiducia che hai riposto in me e per l'incoraggiamento che mi hai dato, mi ha aiutato moltissimo.

Voi tutti, in qualche modo, avete contribuito a rendere questo percorso unico e stupendo e per questo ve ne sono riconoscente, grazie.