UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA BIOMEDICA

Introduzione alle tecnologie blockchain e ai loro possibili usi in sanità e nelle scienze biomediche.

Relatore: Prof. Giovanni Sparacino

Laureando: Marco Tondello

ANNO ACCADEMICO: 2021/2022

INDICE

SOMMARIO

L'idea di una moneta digitale decentralizzata risale fino gli anni '80, ma è stata realmente concretizzata solo nel 2008. Utilizzando e implementando tecniche e concetti già esistenti del mondo informatico e crittografico, è stato successivamente creato il Bitcoin; la prima vera e propria moneta digitale che sfruttava una nuova tecnologia: la blockchain.

Nonostante siano state impiegate, nei primi anni di vita, per le sole criptovalute, nell'ultimo decennio le tecnologie blockchain hanno rapidamente acquisito importanza e popolarità al punto tale da essere proposte per l'implementazione in settori oltre quello per cui sono state inizialmente pensate.

Il presente elaborato si pone come obiettivo quello di trattare a caratteri generali le tecnologie blockchain per poi esporre, tramite una rassegna bibliografica, suoi potenziali usi in ambito sanitario e biomedicale. Dopo una breve introduzione storica viene fornita una descrizione dei componenti principali e delle funzioni che ne garantiscono le proprietà di immutabilità, condivisione e decentralizzazione; caratteristiche che hanno motivato il successo di tali tecnologie che non si è ancora però concretizzato oltre l'ambito finanziario.

1. BREVE STORIA DELLA TECNOLOGIA BLOCKCHAIN

1.1 Nascita nel contesto economico finanziario

La nascita della tecnologia blockchain è avvenuta, grazie a Satoshi Nakamoto, un gruppo tutt'ora anonimo di uno o più individui, nel 2008, con la sua presentazione nel documento "*Bitcoin: A Peer to Peer Electronic Cash System"*. [1] Questo documento presentava un progetto che sarà la base, con eventuali modifiche e variazioni, di tutte le future tecnologie blockchain. Il Bitcoin fu infatti solo la prima delle sue tante implementazioni.[2]

È necessario perciò riferirsi a Bitcoin, o più in generale al concetto di contante digitale, per poter capire appieno le motivazioni e le problematiche che hanno portato alla nascita delle tecnologie blockchain.

Il concetto di contante digitale non è nuovo, è una volontà e una visione che risale fino agli anni '80. Si può pensare alla carta di credito come soluzione a tale problema, ma in realtà non lo è mai stata, a causa di un problema di anonimato. Col contante fisico non si può rintracciare chi ha speso, quanto ha speso, e in cosa ha speso mentre con una carta di credito si può risalire a tutte queste informazioni, che vengono poi registrate.

Sorse inoltre un particolare problema nella transizione da contante fisico a digitale, quello dell'affidabilità, ovvero del *double-spending*. Era possibile, ottenuta una unità di contante digitale, farne una copia e poterlo spendere più volte, cosa assai più complicata da fare con la controparte fisica.

Il primo a proporre una soluzione ad entrambi i problemi fu il criptologo David Chaum. Nel 1982 pubblicò "*Computer systems established, maintained, and trusted by mutually suspicious groups*" [8], dove proponeva un sistema distribuito che viene stabilito, mantenuto e considerato affidabile da parti non necessariamente affidabili. Inventò la *blind signature*, letteralmente firma cieca, un protocollo criptografico che permette di firmare un documento o un dato senza conoscerne il contenuto, e che, implementato nel sistema proposto da Chaum, garantiva l'anonimato prevenendo anche il double-spending.

Questa è stata la prima seria proposta di contante elettronico. Funzionava, ma richiedeva comunque l'accesso perenne ad un server gestito da un'autorità centrale, generalmente una banca, e che tutti i partecipanti si fidassero di essa. Se il server fosse andato offline, non si sarebbero potute effettuare transazioni.

Un po' di anni dopo, nel 1988, lo stesso Chaum in collaborazione con altri due crittografi, A. Fiat e M. Naor, proposero il contante elettronico offline.

L'idea era semplice ma efficace: non preoccuparsi di prevenire il *double-spending* ma concentrarsi sull'individuarlo. Questa tecnica crittografica prese il nome di *secret sharing*; una

informazione veniva criptata, divisa e consegnata ad un certo numero di partecipanti. Se un numero abbastanza alto di partecipanti avesse partecipato alla decodifica dell'informazione criptata, essa avrebbe potuto essere decriptata.
Quando una moneta veniva assegnata ad una persona, ne veniva codificata anche l'identità in modo tale che solo lei, nemmeno l'entità che rilasciava la moneta, potesse decodificarla. Ogni volta che la moneta veniva spesa, il destinatario chiedeva di decodificare una parte dell'identità, e poi la registrava. Questa decodifica non era abbastanza per poter risalire all' identità, ma se mai la moneta fosse stata spesa una seconda volta, nel momento in cui i due destinatari andavano dall'entità elargente a scambiare le loro monete, quest'ultima poteva risalire all'identità della persona che ha commesso *double-spending*.
Chaum prese queste idee e nel 1989 le commercializzò fondando *"Digicash"*. L'effettivo contante venne chiamato *ecash*, e venne implementato da alcune banche, di cui una in Finlandia e le restanti negli Stati Uniti.
Col tempo Chaum brevettò alcune tecnologie dietro il sistema *Digicash*, tra cui la *blind signature*, impedendo così ad altre persone di sviluppare sistema di contanti elettronici basati sullo stesso protocollo. Questo, più il fatto che *Digicash* permetteva transazioni solo tra utenti e mercanti, e non tra utenti e utenti, portò infine al suo fallimento.
Un importante falla dei sistemi di contante elettronico di quel periodo era che il suo valore era legato ad un bene fisico. Per ottenere 100$ di *ecash*, nel caso di *Digicash*, era necessario pagare 100$ alla banca che lo elargiva. Vennero tentate le più svariate alternative, fino a legare il valore del contante elettronico all'oro contenuto in un vault. Tutti questi metodi però presentavano l'ennesimo problema, erano legati ad un bene materiale. [6]

Per realizzare una moneta propriamente digitale, che abbia un proprio valore slegato da beni materiali, era necessario creare scarsità (come avviene per l'oro e il diamante).
Nel mondo digitale un modo per ottenere scarsità è legare l'ottenimento della moneta alla risoluzione di un problema computazionale, che richiede tempo e risorse per essere risolto.
L'idea che le soluzioni a puzzle computazionali potessero essere oggetti digitali di valore risale a metà degli anni '70; si possono infatti trovare in una forma primitiva nei *puzzle di Merkle*. [4]
L'idea venne poi ripresa da C. Dwork and M. Noar [9], nel 1992, per combattere lo *spam* di e-mail [4]. Ogni volta che una e-mail veniva inviata era necessario risolvere uno di questi puzzle, che richiedeva pochi secondi, e allegarne la soluzione. Se il destinatario non avesse ricevuto anche la soluzione del puzzle, l'e-mail sarebbe stata scartata.
Per l'utente medio che inviava poche e-mail questo non era un problema, ma per lo *spammer* che ne inviava migliaia contemporaneamente, risolvere i puzzle poteva diventare proibitivo.

Questi puzzle dovevano avere però specifiche proprietà per essere utili come deterrenti di *spam*. Doveva essere impossibile, per lo *spammer*, risolvere un puzzle e allegare la sua soluzione a tutte le e-mail in invio: queste dovevano quindi essere indipendenti tra di loro (generalmente si basavano su mittente, destinatario, data e ora).
Inoltre, il destinatario dell'e-mail doveva essere in grado di controllare la validità della soluzione senza dover risolvere il puzzle a sua volta.
Infine, con il miglioramento dell'hardware a disposizione e quindi con la facilitazione nel risolverli, questi necessitavano di un modo per scalare in difficoltà.
Tutto ciò è ottenibile usando delle funzioni crittografiche *hash* per disegnare tali puzzle.
Questa idea venne poi ripresa e riproposta da Adam Back nel 2002 col nome *Hashcash* [5] che però non prese mai piede, anche se l'idea di limitare l'accesso a risorse tramite l'utilizzo di puzzle computazionali è ancora discussa tutt'oggi. [6]

L'idea di creare una struttura dati formata da blocchi concatenati (ovvero *blockchain*) invece, appare già nei primi anni '90.
Nel 1991 H. Stornetta rilasciarono un trattato [10] che introdusse l'idea di registrare il *timestamp* (letteralmente marca temporale, una sequenza di caratteri che rappresenta data e/o ora) di un documento, piuttosto che il contenuto. Lo scopo era di dare un'idea approssimativa di quando un documento è stato creato, oltre che all'ordine in cui sono stati creati. Tutto ciò richiedeva ovviamente che il *timestamp* non venisse modificato dopo la sua creazione.
Il sistema, al momento della certificazione del documento, lo firmava con un puntatore al documento precedente ed il *timestamp*. Il puntatore era di tipo particolare in quanto rimandava ai dati contenuti nel documento, non a una locazione; se quindi le informazioni in questione venivano alterate, esso diventava nullo.
Con questo protocollo, ogni certificato di un documento assicura l'integrità del contenuto del documento che lo precede. Ciò significa che ogni documento certificato, fissa l'intera storia di quelli certificati fino a quel punto. Se un cliente nel sistema tiene traccia di almeno alcuni certificati, quelli dei suoi documenti e quelli successivi e precedenti, tutti i partecipanti del sistema possono assicurare che l'ordine e la storia dei documenti non possano essere modificati.
Un trattato successivo [11] propone un miglioramento all'efficienza; invece di legare i documenti individualmente, essi possono essere collezionati in blocchi e questi collegati tra di loro. In ogni blocco, i documenti sarebbero comunque legati tra di loro ma in una struttura ad albero invece di una catena lineare, meno efficiente. [6]

Nel 2008 i concetti di contante digitale, creazione di scarsità tramite puzzle computazionali, e dati concatenati tra di loro, di cui abbiamo visto la nascita e lo sviluppo, culminarono nella creazione della tecnologia blockchain. I primi tentativi risalgono però a qualche anno prima. Nel 1998 Wei Dai propose il *b-money*, mentre nel 2005 Nick Sazbo, introdusse *Bitgold*, che però presentavano seri problemi. Se fossero nati disaccordi non ci sarebbe stato nessun sistema per risolverli, anche se a detta degli autori la decisione era lasciata alla maggioranza degli utenti appartenenti alla rete. Questo però poteva portare a grosse falle di sicurezza visto che qualsiasi persona poteva impostare un server appartenente alla rete, o centinaia di essi sotto identità diverse, a meno che non ci fosse un'entità centrale dedita a controllare ogni utente entrante nel sistema. Mancava inoltre un sistema per creare scarsità. Risolvere un puzzle computazionale, la cui difficoltà non aumentava nel tempo, portava al guadagno di una unità di moneta. Con il miglioramento dell'hardware però, risolvere puzzle e quindi ottenere monete risultava sempre più facile, portando perciò a una diminuzione del valore di quest'ultime.
Questi problemi, mai risolti, e assenti in Bitcoin, causarono l'abbandono prematuro delle due proposte, che non uscirono mai dalla fase concettuale. [6]

1.2 Trend attuali

Dalla sua nascita nel 2008 le tecnologie blockchain hanno subito varie evoluzioni; si è passati dalla "versione" 1.0 all'attuale 4.0, transitando per le 2.0 e 3.0.
L'utilizzo della versione 1.0 era limitata alle sole criptovalute, situazione cambiata con l'introduzione degli *smart contracts* (che verranno trattati nel capitolo 3) che diedero inizio all'era 2.0, il cui scopo era quello di espanderne l'utilizzo a qualsiasi business che potesse trarre vantaggio da quest'ultimi tramite l'automatizzazione di complesse transazioni. La principale esponente di questa versione è la tecnologia blockchain *Ethereum*, tutt'oggi coinvolta in numerosi progetti grazie al suo continuo aggiornamento. Lo sviluppo e la crescita delle *DApps* ha dato inizio alla 3.0. Queste non sono altro che applicazioni il cui *backend* è basato su blockchain e *smart contracts*, ma presenta un'interfaccia utente che ne permette il facile utilizzo, creando così la possibilità di sviluppare piattaforme per lo scambio di *NFT* (*non fungible token,* che permettono di identificare univocamente un oggetto digitale), prestiti in criptovalute e altro. L'attuale versione 4.0 promette di rendere la tecnologia blockchain usufruibile da qualsiasi tipo di business o utente, come avvenuto per l'internet e i social. Questo viene definito come *Web 3.0*. [2] [6]
Nel settore sanitario e biomedico, come si vedrà in seguito, molte delle proposte si basano sull'impiego di smart contracts e sull'utilizzo della tecnologia Ethereum. [20] [21]

2. DEFINIZIONI DEI CONCETTI DI BASE

La tecnologia blockchain può risultare complessa; tuttavia, può essere semplificata esaminandone i singoli componenti. Utilizza infatti idee e meccanismi dell'informatica e della crittografia già esistenti (funzioni *hash* criptografiche, chiavi criptografiche asimmetriche) uniti a concetti di database (come registri). Questo capitolo ha lo scopo di introdurre e discutere i singoli concetti: *transazioni*, *funzioni di hash*, *chiavi simmetriche* e *asimmetriche*, *registri*, *blocchi*, *reti peer-to-peer* e infine *blockchain*.

Nel successivo capitolo 3 presenteremo invece come questi componenti danno vita alla tecnologia blockchain e alle sue importanti proprietà.

2.1 Transazioni

Una transazione è un'interazione fra due o più parti, in cui le informazioni scambiate variano in base al sistema dove sono implementate.

Richiede generalmente input e output, anche se sono possibili variazioni.

L'input è in genere una lista di dati da trasferire. Il mittente deve dimostrare che ne è il proprietario firmando, con una chiave privata, la transazione.

L'output è formato usualmente dall'indirizzo dei destinatari con, inoltre, quanti e quali dati dovranno ricevere.

La validità e l'autenticità delle transazioni sono fattori importanti. La validità assicura che la transazione rispetti i protocolli del sistema dove è implementata, mentre l'autenticità che il mittente dei beni digitali ne sia l'effettivo proprietario.[2]

2.2 Funzioni di hash

Le funzioni di *hash*, come *MD5*, *SHA-1*, *SHA-256* (quella utilizzata prevalentemente nelle implementazioni attuali di blockchain), *SHA-3* e *BLAKE2*, sono lo strumento più utilizzato dai crittografi; si trovano nelle firme digitali, codifica a chiave pubblica, verifica di integrità, autenticazione di messaggi, protezione di password e tanti altri protocolli crittografici. Ogni volta che viene mandata una e-mail, un messaggio sul telefono o si effettua la connessione a un sito web si può star certi della presenza di una funziona di *hash* nel percorso.[7]

Una funzione di *hash*, dato in input un testo di lunghezza qualsiasi, calcola un output, chiamato *digest*, di lunghezza definita e relativamente unico. Permette così a due individui di prendere la stessa informazione in input, calcolarne il *digest*, e ottenere lo stesso risultato in output a patto che non siano state apportate modifiche al messaggio iniziale. Anche la più piccola modifica al

messaggio in input (ad esempio un solo bit) provoca infatti l'ottenimento di un risultato completamente diverso.
Le funzioni di hash crittografiche hanno le seguenti importanti proprietà:

1. Sono resistenti alle preimmagini. Una preimmagine di un valore H, a cui è stato applicato una funzione di *hash*, è qualsiasi messaggio M, tale che hash(M) = H. [7] La resistenza alle preimmagini garantisce quindi l'impossibilità di calcolare il corretto input dato un valore di output.
2. Sono resistenti alle seconde preimmagini. Le funzioni di *hash* sono disegnate in modo tale che dato uno specifico input, è computazionalmente infattibile trovare un secondo input che produca lo stesso output (dato x, trovare y tale che hash(x)=hash(y)). L'unico approccio disponibile è quello di cercare tra tutti gli input possibili. Si pensi però ad un messaggio di input a 1024 bit, vorrebbe dire cercare tra 2^{1024} valori possibilmente corretti.
3. Sono resistenti alle collisioni. È impossibile trovare due valori di input che producano lo stesso *digest*. Questa proprietà implica la resistenza alle seconde preimmagini.

Una funzione hash in particolare è usata in molte applicazioni blockchain: la *Secure Hash Algorithm (SHA)*, con una grandezza di output di 256 bit (*SHA-256*). La *SHA-256* ha un output di 32 bytes (8 bit = 1 byte, 256 bit = 32 byte), generalmente mostrato come una stringa esadecimale di 64 caratteri. Questo significa che ci sono 2^{256} possibili digest. Di seguito un esempio di implementazione di *SHA-256* dei caratteri 1, 2, a, b, A e B:

SHA256(1) = 6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
SHA256(2) = 53c234e5e8472b6ac51c1ae1cab3fe06fad053beb8ebfd8977b010655bfdd3c3
SHA256(a) = ca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb
SHA256(b) = 3e23e8160039594a33894f6564e1b1348bbd7a0088d42c4acb73eeaed59c009d
SHA256(A) = 559aead08264d5795d3909718cdd05abd49572e84fe55590eef31a88a08fdffd
SHA256(B) = df7e70e5021544f4834bbee64a9e3789febc4be81470df629cad6ddb03320a5c

Dato che ci sono praticamente una infinità di input, ma un numero finito di *digest*, è possibile, ma altamente improbabile, che avvenga una collisione dove hash(x)=hash(y), con y e x due differenti input. Nel caso dello *SHA-256* ci sono 2^{256} possibili *digest* ma, per esempio, in un messaggio a 1024 bit, 2^{1024} possibili input. Questo significa che se avvenisse una collisione si dovrebbero cercare i due messaggi tra $2^{1024}/2^{256} = 2^{768}$ possibili input. [7]
Molto spesso le funzioni di hash vengono utilizzate in combinazione con i *nonce crittografici*.

Un *nonce* non è altro che un numero arbitrario che viene utilizzato solo una volta, e che viene combinato con il valore di input in una funzione di *hash* per ottenere digest diversi per ogni *nonce*:

hash (valore in input + nonce) = digest [2] [3]

2.3 Crittografia a chiave simmetrica

I sistemi a chiave simmetrica utilizzano una singola chiave per la codifica e decodifica del messaggio. In un sistema utilizzante questa tecnica le informazioni codificate possono essere decodificate tramite la chiave con qui il messaggio è stato codificato; nessuno altro può accedere all'informazione.

Questo metodo presente dei problemi; la chiave deve essere scambiata tramite un canale sicuro, implicando un rapporto di fiducia preesistente tra le due parti, ed inoltre è necessario crearne una nuova per ogni rete di scambio. [7]

2.4 Crittografia a chiave asimmetrica

I sistemi a chiave asimmetrica utilizzano due chiavi: una pubblica, che viene appunto resa visibile senza il rischio di compromettere la sicurezza del processo, e una chiave privata che, al contrario, rimane segreta per mantenere la crittografia. Entrambe sono legate tra di loro matematicamente; un messaggio codificato con una chiave è infatti decodificabile solo con l'impiego dell'altra chiave, ma non è possibile risalire alla chiave privata a partire da quella pubblica.

Questo sistema permette quindi la creazione di un rapporto di fiducia tra utenti che non sono affidabili o semplicemente non sanno se fidarsi.

I documenti, o le transazioni, vengono criptati utilizzando la chiave pubblica del destinatario, in modo tale che solo lui possa decriptarli avendo la relativa chiave privata. [2]

Questo sistema presenta però uno svantaggio: le funzioni matematiche che generano il codice cifrato e quelle inverse, che lo decifrano, sono molto lente, più di quelle impiegate nella crittografia simmetrica.

Per ovviare a questo problema si utilizza un sistema detto 'a crittografia mista': il messaggio viene criptato con una chiave simmetrica, che viene a sua volta codifica con la chiave pubblica del destinatario e spedita insieme al messaggio, velocizzando di molto il processo. [7]

2.5 Registri

Un registro è una collezione di transazioni. Generalmente, essi vengono tenuti in grandi database digitali di proprietà o gestiti da una terza parte fidata per conto della comunità. Questi registri possono essere implementati in maniera centralizzata su un solo, o più server.

2.6 Blocchi

Un blocco è una selezione di transazione raggruppata insieme e organizzata logicamente, la cui dimensione varia in base al design della blockchain in cui è implementato.
È formato generalmente da dati, da un'intestazione e da un numero, secondo un sistema simile al seguente:

- *Intestazione del blocco*
    - Il *digest* dell'intestazione del blocco precedente;
    - Il *digest* dei dati contenuti nel blocco;
    - Un *timestamp*;
    - Un valore di *nonce*.
- *I dati del blocco*
    - Lista di transazioni più eventuali eventi del registro;
    - Altri dati.
- *Numero del blocco*, conosciuto come altezza del blocco.[2]

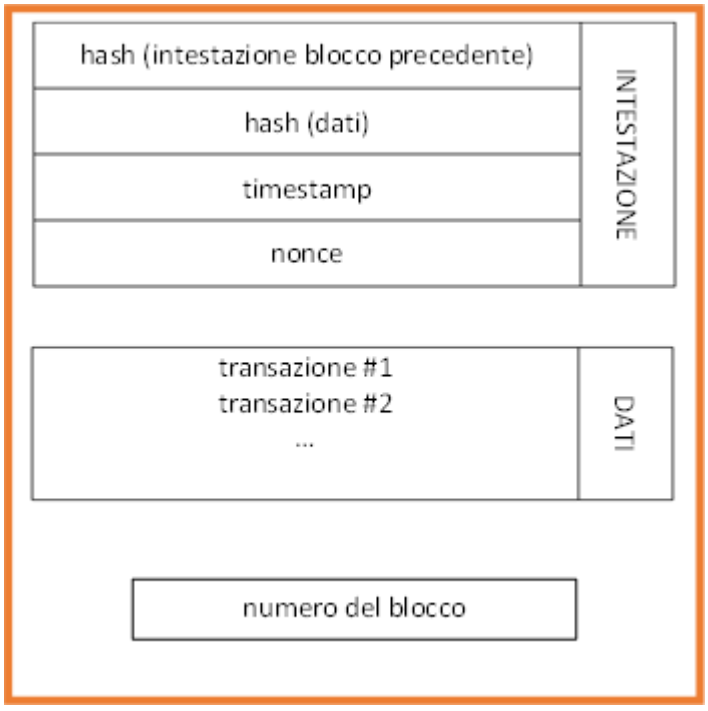

Fig 1. Generico blocco di una rete blockchain [2]

Un tipo particolare di blocco è quello di genesi; esso non presenta il *digest* dell'intestazione del blocco precedente in quanto è il primo della blockchain e il momento in cui è stata creata.

2.7 Rete peer-to-peer (P2P)

Una normale rete può essere vista come un insieme di nodi, solitamente computer su cui si interfacciano degli utenti, collegati ad un nodo centrale, generalmente un server, gestito da un'autorità.

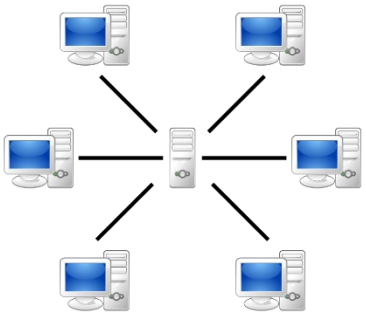

Fig 2. Rete centralizzata

Una rete peer-to-peer invece è un insieme di nodi collegati direttamente a tutti gli altri nodi, senza necessità di un'autorità centrale.

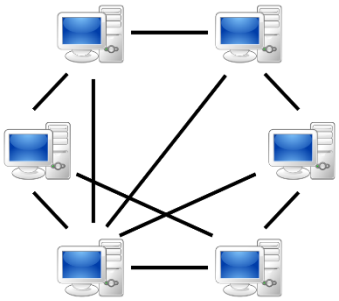

Fig 3. Rete peer-to-peer

2.8 Blockchain

La blockchain è un tipo particolare di registro formata da blocchi collegati tra di loro, in maniera cronologica e sequenziale, da funzioni *hash*.

La struttura generica di una blockchain può essere visualizzata con l'aiuto del seguente grafico:

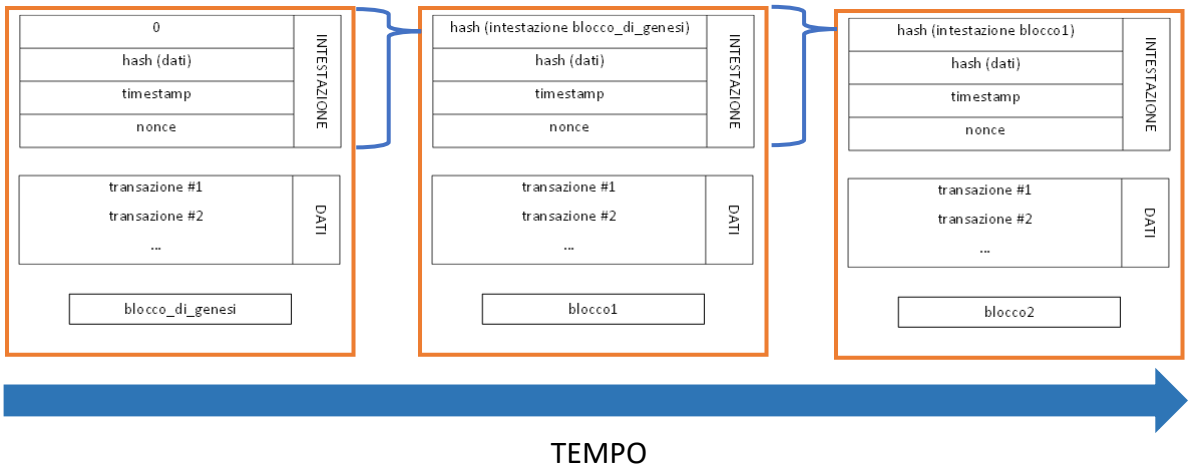

Fig 4. Struttura di una blockchain [7]

Ogni blocco è collegato al precedente tramite un puntatore che contiene il *digest* dell'intestazione di quello che lo precede, formando così' una catena di blocchi (*blockchain*). La particolarità di questa struttura dati è che nell'intestazione di ogni blocco è presente il *digest* dei dati contenuti in esso. Se quindi un blocco già pubblicato dovesse subire la più piccola modifica ai suoi dati, il *digest* di questi cambierebbe, e di conseguenza anche quello dell'intestazione. Ciò scatenerebbe una reazione a catena che causerebbe il cambiamento dei *digest* di ogni singolo blocco che segue. [2]

3. DESCRIZIONE DEL PRINCIPIO DI FUNZIONAMENTO DELLE TECNOLOGIE BLOCKCHAIN

3.1 Tecnologie blockchain

Una generica tecnologia blockchain consiste in una struttura blockchain distribuita e condivisa in una rete peer-to-peer, a solo inserimento, immutabile e aggiornabile solo tramite consenso dei partecipanti della rete.

Viene spesso fatta confusione tra blockchain e tecnologie blockchain, ma queste sono diverse tra loro. La seconda infatti è un insieme di regole e protocolli su come una blockchain va creata e mantenuta più altre tecnologie, tra cui la blockchain stessa. Differenti tecnologie blockchain hanno differenti regole di partecipazione, specifiche sulla creazione di transazioni, metodi per l'archiviazione dei dati e meccanismi di consenso. Quando una rete che addotta una tecnologia blockchain (da qui in avanti rete blockchain) è creata, l'effettiva blockchain è vuota a parte il blocco genesi. [12]

3.1.1 Distribuita e condivisa

La condivisione e distribuzione in una rete peer-to-peer indica che la blockchain non è posseduta da una singola entità, ma ogni utente ne possiede una copia a livello locale e partecipa alla sua costruzione. [3]

3.1.2 Immutabile

I blocchi di una blockchain possono essere solo inseriti in ordine temporale; ciascuno sarà preceduto da uno più recente e succederà a uno più vecchio, implicando l'impossibilità di cambiarlo una volta aggiunto, e rendendo di conseguenza la catena di blocchi immutabile. [3]

3.1.3 Aggiornabile solo tramite consenso

Ogni modifica alla blockchain viene portata a compimento solo se validata da stringenti criteri, in base all'implementazione utilizzata, e solo se è stato raggiunto un consenso tra tutti i partecipanti della rete. L'ottenimento di quest'ultimo avviene tramite particolari meccanismi, analizzata nel capitolo 3.2, che consistono nell'elezione di un singolo nodo che apporterà le modifiche, che verranno poi replicate da tutti gli altri nodi.

Questo è l'attributo più importante e critico delle tecnologie blockchain in quanto ne consente la decentralizzazione. [3]

3.1.5 Tipologie di reti blockchain

Le reti blockchain possono essere divise in due categorie in base al modello di autorizzazione che adottano, che determina chi può pubblicare nuovi blocchi: *permissioned* (reti blockchain private) e *permissionless* (reti blockchain pubbliche).

Le reti *permissionless* sono piattaforme decentralizzate la cui interazione con la blockchain è aperta a tutti i nodi, senza necessità di avere un'autorizzazione. La natura aperta di questo tipo di rete la rende però esposta a potenziali attacchi da parte di nodi malintenzionati che mirano a modificare le transazioni o blocchi. Per prevenire tali situazioni, questa tipologia di rete spesso utilizza dei particolari modelli di consenso, che prevedono il dispendio di risorse da parte dei nodi per poter avere la possibilità di pubblicare un nuovo blocco.

Nelle reti *permissioned* invece è necessaria un'autorizzazione da parte di un'autorità per poter interagire con la blockchain. Quest'ultima può impedirne la lettura ad alcuni nodi, come anche negare l'abilità di inviare transazioni. Vengono utilizzati anche qui dei meccanismi di consenso per l'interazione con la blockchain, ma molto spesso sono meno esosi in termini di risorse dei rispettivi modelli utilizzati nelle reti private. Prima dell'accesso alla rete viene infatti stabilita l'identità del partecipante, assicurando così che i nodi abilitati a modificare la blockchain siano affidabili. L'abilitazione può essere revocata in casi di malcondotta. [2]

3.1.6 Layout di una rete blockchain

Le tecnologie blockchain possono essere pensate come uno strato di una rete blockchain che opera su una rete peer-to-peer che ha come base internet. Un diagramma utile visualizzare il layout è il seguente:

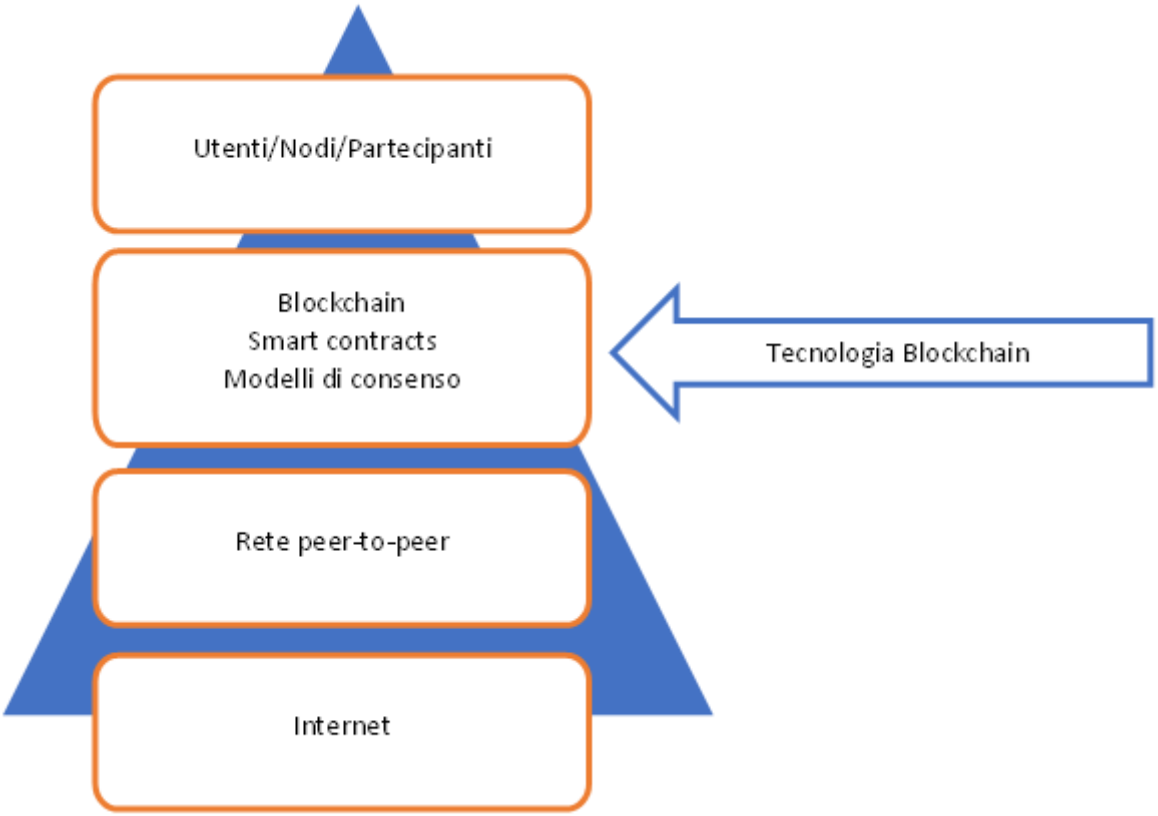

Fig 5. Diagramma di una rete blockchain [3]

Nel primo strato del diagramma c'è internet, che provvede un livello base di comunicazione per ogni rete. Una rete peer-to-peer lo sfrutta come base e fornisce protocolli di propagazione di informazioni, come il protocollo Gossip.

Una tecnologia blockchain si basa quindi su quest'ultima. È rappresentata da un unico blocco, composto da blockchain, meccanismi di consenso, e *smart contracts*.

Nell'ultimo livello sono presenti i nodi, che si connettono alla rete blockchain ed eseguono varie operazioni come fornire il consenso, verificare transazioni ed elaborazione. Questi possono essere generalmente, *miners* (minatori), che creano nuovi blocchi, o *validators*, che convalidano e firmano digitalmente le transazioni. Tutti i nodi a priori sono *validators*, ma non tutti sono necessariamente *miners*, anche se vengono incentivati ad esserlo.[3]

3.1.7 Inserimento di transazioni nella blockchain

Le tecnologie blockchain sono innumerevoli (basti pensare anche solo alla quantità di criptovalute presenti nel mercato, ognuna con una sua tecnologia) e risulta impossibile, per gli scopi di questo elaborato, elencare in dettaglio un processo che va dalla creazione di una transazione al suo inserimento nella blockchain, che le coinvolga tutte.

I passaggi elencati di seguito rappresentano il procedimento che è stato utilizzato nella prima tecnologia blockchain creata [1], ovvero quella del *Bitcoin*, e che tutte quelle a venire hanno

usato come base. A questi passaggi sono stati implementate nozioni di carattere generale per fornirne una visione che non si limiti al solo *Bitcoin*.
Dalla nascita di una transazione, alla sua inclusione in un blocco, e all'eventuale pubblicazione nella blockchain, avvengono i seguenti passaggi:

1. La transazione è prima creata e poi firmata digitalmente da un nodo con la sua chiave privata. Durante la creazione le viene assegnato, generalmente in automatico ed in base alla rete blockchain utilizzata, un valore chiamato tassa (con le criptomonete consiste in una percentuale della moneta trasferita), che può essere visto come un valore di priorità; più alto è questo valore più sono incentivati i *miners* ad aggiungerla ad un blocco. Alcune implementazioni non utilizzano le tasse, ma si basano su un sistema temporale; le transazioni più vecchie sono quelle più prioritarie.
2. La transazione è propagata, tramite protocolli di trasmissione dati, a tutti gli utenti, che la convalidano e la verificano utilizzando la chiave pubblica dell'utente che la ha creata. Viene poi aggiunta alla loro memoria locale, chiamata *mempool*, che contiene tutte le transazioni da confermare.
3. Una volta che la transazione giunge ai *miners* viene raggruppata insieme ad altre in un blocco. Le transazioni da inserire nel blocco vengono scelte in base ai criteri di priorità utilizzati.
4. Inizia il processo di *mining*, ovvero la ricerca di un nuovo blocco. Durante questo passo i *miners* fanno a gara per finalizzare il blocco appena creato.
5. Il primo *miner* che adempie ai criteri del meccanismo di consenso implementato nella rete blockchain "trova il blocco", che viene quindi trasmesso a tutti i nodi della rete.
6. Viene verificata la validità del nuovo blocco, controllando che tutte le transazioni in esso siano valide (ogni rete blockchain ha propri criteri di validità per le transazioni).
7. Se il blocco ha raggiunto il consenso viene aggiunto alla blockchain. Le transazioni contenuti in esso vengono confermate, eseguite e rimosse dalla *mempool*. In questo passo vengono lanciati anche eventuali *smart contracts*.
8. Il nodo *miner* che lo ha aggiunto guadagna eventualmente il valore delle tasse delle transazioni che ha incluso nel blocco e un incentivo (generalmente un numero variabile di monete nel caso di una criptovaluta). [1] [3]

Una decisione critica che ogni rete blockchain deve affrontare è decidere quale sarà il nodo che pubblicherà il prossimo blocco. Questa viene presa grazie ad un meccanismo di consenso.[3]

3.2 Meccanismi di consenso

Il consenso è un processo di accordo tra nodi non affidabili sullo stato finale di un dato. Tra due entità (sistema server e client per esempio) questo è facile ottenerlo, ma tra molte risulta più complesso.

I meccanismi di consenso sono procedimenti che portano al raggiungimento del consenso tra i nodi, permettendo così una decentralizzazione della rete. Una delle loro caratteristiche fondamentali è che lo garantiscono anche in presenza di nodi fallati.

Le falle di un nodo sono principalmente di due tipi:

- *Fail-stop*; occorrono quando un nodo è semplicemente crashato, o non ha risposto. Tra le due tipologie sono le più facili da gestire.
- *Bizantine*; occorre quando un nodo esibisce intenti maliziosi o assume atteggiamenti inconsistenti. È più difficile da gestire delle altre falle dato che portano alla creazione di confusione a causa delle informazioni ingannevoli. Possono essere il diretto risultato di un attacco esterno, di un bug nel software o di una corruzione dei dati.[3] [13]

In una rete blockchain il modello di consenso viene utilizzato per decidere quale nodo proporrà il nuovo blocco da aggiungere alla blockchain. I più importati e utilizzati sono i seguenti:

- *Proof of Work (PoW):* dove il nodo che pubblicherà il prossimo blocco sarà il primo che riuscirà a risolvere un puzzle computazionale. La soluzione a questo puzzle è la "*proof*" (prova) che hanno speso risorse (corrente elettrica sottoforma di potenza di calcolo). Il puzzle è fatto in modo tale che una volta trovata la soluzione questa sia facile da verificare per tutti gli altri nodi partecipanti alla rete, permettendo così la verifica e convalida del blocco.

  Un puzzle comune è quello di richiedere che il *digest* dell'intestazione del nuovo blocco sia minore di un certo valore. I nodi *miners* quindi, cambiando il *nonce* del blocco, cercheranno di trovare un *digest* che soddisfi i requisiti. Ovviamente per ogni tentativo dovranno ricalcolarlo, diventando a lungo andare, un processo intensivo.

  Il valore del *digest* che deve essere raggiunto può essere modificato nel tempo per aumentare la difficoltà o ridurla; influenzando così il tempo di pubblicazione di ogni blocco, e impedendo che una sola entità prenda possesso del processo.

  Consideriamo un puzzle dove, usando l'algoritmo *SHA-256*, un computer deve trovare il *digest* rispettando i seguenti criteri:

  SHA256("blockchain" + nonce) = digest che inizia con "000000"

La stringa “blockchain” è inserita insieme ad un valore di *nonce*, solitamente è un valore numerico. Le soluzioni a questo puzzle sono facili da calcolare:

SHA256("blockchain0")=0xbd4824d8ee……..938 (non risolto)
SHA256("blockchain1") = 0xdb0b9c1cb5……..a10 (non risolto)
SHA256("blockchain10730895") = 0x000000ca14……..587 (risolto)

Per la risoluzione sono stati necessari 10730896 tentativi, completati in circa 54 secondi con un computer relativamente vecchio. [2]
In questo esempio per aumentare la difficolta basta aggiungere uno zero alla stringa. Cambiando la stringa in “0000000”, lo stesso hardware ha impiegato 934224175 tentativi, risolvendo il puzzle in 1 ora, 18 minuti e 12 secondi, con la seguente soluzione:

SHA256("blockchain934224174") = 0x0000000e2a……..a81 [2]

Non è attualmente conosciuta nessuna scorciatoia a questo processo; un *miner* che pubblica un nodo deve spendere potenza di calcolo per trovare il corretto valore di *nonce*. Una volta trovato un valore che soddisfa i requisiti, viene allegato al blocco e condiviso con tutti i nodi della rete blockchain. Verificare la validità del blocco e la correttezza del *nonce* è facile, in quanto basta aggiungerlo all’intestazione del blocco e calcolare il *digest*, richiedendo il calcolo di un solo *hash*. [2]

- *Proof of Stake (PoS):* basato sull’idea che più risorse un utente ha investito nella rete, ovvero lo “*stake*” (investimento), più questo vorrà che il sistema abbia successo invece di fallire. Lo *stake* corrisponde generalmente a criptovalute che l’utente possiede nella rete.
  Con questo modello non c’è bisogno di effettuare calcoli intensivi come nel *proof of work*. Dato che non è necessario un impiego massiccio di tempo, elettricità o potenza, molte reti blockchain hanno deciso di abbandonare l’idea di ricompensare la creazione di un nuovo blocco; in questi sistemi i blocchi *miner* fanno a gare per guadagnare le tasse delle transazioni.
  I principali metodi con la quale una rete blockchain impiega lo *stake* sono: selezione random, votazione a più round e in base all’età. Gli utenti con più *stake* hanno comunque più probabilità di pubblicare un nuovo blocco, nonostante le diverse metodiche.

Con la selezione random, la rete analizza lo *stake* di ogni utente e sceglie in base alla ratio tra questo e la quantità totale. Se un utente possiede il 42% dello *stake* totale, ha la possibilità di essere scelto il 42% delle volte.

Tramite il sistema di votazione a più round c'è un grado di difficoltà aggiunto. La rete sceglierà diversi utenti con *stake* a cui saranno proposti dei blocchi da votare. Il blocco scelto verrà quindi pubblicato. Possono essere necessari più round per sceglierlo definitivamente.

Il metodo basato sull'età prevede l'aggiunta della omonima proprietà. Quando uno *stake* viene tenuto da un utente per un certo intervallo di tempo (per esempio dieci giorni), quell'utente ha diritto di partecipare alla votazione del prossimo blocco. Una volta che la votazione è avvenuta, l'età dello *stake* utilizzata per la selezione viene resettata, e altro tempo deve passare prima che questo sia in grado di esprimere un voto. Questo metodo permette agli utenti che posseggono più *stake* di effettuare più voti ma senza dominare il sistema. Questo metodo viene generalmente impiegato in reti blockchain di criptovalute, e lo *stake* rappresenta una moneta.[2]

3.3 Forking

I cambiamenti ai protocolli e alla struttura dati di una rete blockchain si chiamano *forks* (bivi); questi possono portare ad una divisione di quest'ultima, creando più versione della stessa.

Il termine *fork* viene anche utilizzato da alcune reti blockchain per descrivere conflitti nella blockchain temporanei e che non derivano da cambiamenti a livello di software. Nel caso in cui due *miner* pubblichino due differenti versioni del successivo blocco simultaneamente, alcuni nodi potrebbero ricevere uno prima dell'altro. In questo caso considerano valido il primo che hanno ricevuto, ma salvano la blockchain con il blocco "scartato" in caso diventi la più lunga. Il prossimo nuovo blocco si collegherà a solo uno dei due precedenti; viene quindi considerata valida la blockchain più lunga e scartata quella più corta.

Esistono due categorie di *fork*: *soft fork* (bivi leggeri) e *hard forks* (bivi forti).

I *soft fork* sono cambiamenti all'implementazione della tecnologia blockchain retrocompatibili. I nodi non aggiornati possono continuare ad effettuare transazione con quelli aggiornati. Se nessuno nodo (o solo pochi) effettua l'aggiornamento, questo non verrà protratto. Un esempio di *soft fork* può essere quello che avviene nel caso una rete blockchain decidesse di ridurre le dimensioni dei blocchi da 1MB a 0.5MB. I nodi aggiornati aggiusterebbero la dimensione dei

blocchi e continuerebbero le transazioni normalmente; i nodi non aggiornati vedrebbero questi blocchi come validi dato che non violano le loro leggi (la dimensione del blocco è comunque inferiore a 1MB). Tuttavia, se un nodo non aggiornato creasse un blocco con una dimensione maggiore di 0.5MB, questo verrebbe rifiutato e classificato come invalido dai nodi aggiornati.
Gli *hard fork* sono invece cambiamenti che non sono retrocompatibili. Ad un certo punto, solitamente dopo uno specifico blocco, tutti i nodi pubblicanti devono aggiornarsi in modo che i nuovi blocchi non vengano rigettati. I nodi non aggiornati non possono continuare ad effettuare transazioni nella blockchain in quanto sono programmati per rifiutare ogni blocco non conforme alle loro (ormai obsolete) specifiche. Questi pubblicheranno blocchi nel vecchio formato, che saranno rifiutati dai nodi aggiornati e accettati solo da quelli obsoleti. Questo però porta all'esistenza contemporanea di due versioni della stessa blockchain, dove gli utenti di una non possono interagire con quelli dell'altra.
La maggior parte degli *hard fork* sono intenzionali (se per esempio si scoprisse che la funzione di *hash* utilizzata ha una falla, risulterebbe necessario crearne uno per implementarne una nuova), ma possono comunque essere causati da errori di software.[2]

3.4 Smart contracts

Nella loro prima apparizione gli *smart contracts* (contratti intelligenti) vennero definiti come "un protocollo di transazione computerizzato che esegue i termini di un contratto". L'uso suggerito era quello di tradurre clausole contrattuali in codice ed unirlo ad una entità (hardware o software) che possa applicarlo in modo automatico e autonomo [14], minimizzando così il bisogno di intermediari affidabili tra le entità della transazione, e prevenendo l'occorrenza di errori di natura maliziosa o accidentale.
Nel contesto di una tecnologia blockchain, gli *smart contracts* sono *script* scritti in linguaggio di programmazione, che varia da tecnologia a tecnologia, inseriti nella blockchain con un loro indirizzo. L'indirizzo viene calcolato con vari metodi, ma solitamente è il risultato di un'operazione di *hash* sull'indirizzo del creatore a cui viene aggiunto un *nonce*.
Uno *smart contract* viene eseguito quando una transazione viene indirizzata ad esso.
Eccellono quando sono programmati per gestire transazioni di dati [15] tra nodi della rete.
Un esempio utile a capire le meccaniche di uno *smart contract* è quello di un distributore di snack infallibile (lo snack non può incastrarsi)[14]:

soldi + snack selezionato = snack garantito

In una rete blockchain il medesimo esempio è paragonabile ad un nodo che rilascia uno smart contract chiamato “Distributore” al cui interno sono definite due funzioni:

- “ricarica”: che permette di ripristinare la quantità di snack
- “acquista”: che rilascia una unità di snack per ogni moneta scambiata

Una possibile reale implementazione tramite codice è la seguente:

```
pragma solidity 0.8.7;

contract Distributore
{
    // Dichiara le variabili del contratto
    address public owner;   //indirizzo del proprietario
    mapping (address => uint) public snack_disponibili;

    // Quando il contratto "Distributore" è rilasciato:
    // 1. imposta l'indirizzo del proprietario del contratto
    //    ponendolo uguale all'indirizzo del nodo che lo rilascia
    // 2. imposta a 100 la quantità di snack disponibili nel distributore
    constructor()
    {
        owner = msg.sender;
        snack_disponibili[address(this)] = 100;
    }

    // Permette solo al proprietario di ricaricare gli snack
    function ricarica(uint amount) public
    {
        require(msg.sender == owner,
                "Solo il proprietario può ricaricare gli snack.");
        snack_disponibili[address(this)] += amount;
    }

    // Permette a chiunque di acquistare snack
    function acquista(uint amount) public payable
    {
        require(msg.value >= amount * 1 moneta,
                "Devi pagare almeno una moneta per snack");
        require(snack_disponibili[address(this)] >= amount,
                "Non ho abbastanza snack per completare la transazione");
        snack_disponibili[address(this)] -= amount;
        snack_disponibili[msg.sender] += amount;
    }
}
```

Fig 6. Esempio di implementazione di smart contract tramite codice (https://ethereum.org)

Il codice riportato in figura 6 è stato scritto utilizzando *Solidity*, un linguaggio di programmazione per *smart contracts* della tecnologia blockchain *Ethereum*.

La funzione ricarica è programmata in modo da essere invocabile solamente dal nodo che ha rilasciato il contratto, ovvero il proprietario.
La funzione acquista invece è invocabile da tutti.
Le transazioni che chiamano entrambe le funzioni vengono trasmesse a tutti i nodi, come comuni transazioni, per essere aggiunte alla blockchain.
Va osservato che:

- Lo *smart contract* ha un suo account nella blockchain, se questa supporta un modello *account-based*. [16] In questo esempio può essere considerato un nodo che tiene gli *assets* "snack".
- Il contratto permette di esprimere logica finanziaria in codice; "trasferisci 1 unità di snack per ogni unità di moneta ricevuta".
- Uno *smart contract* propriamente scritto dovrebbe descrivere tutti i possibili risultati del contratto. Nell'esempio sopra la funzione "acquista" potrebbe essere scritta in modo da rifiutare tutte transazioni con unità di moneta non intere.
- La relazione che il nodo proprietario intende stabilire con gli altri nodi della rete deve coinvolgere una trasmissione di dati. [15]
- Deve essere attivato da messaggi o transazioni dirette al suo indirizzo.
- Deve essere deterministico: lo stesso input produce lo stesso output. Se un contratto viene scritto in modo non deterministico quando verrà invocato ed eseguito da ogni nodo, questi non avranno lo stesso risultato, e non si otterrà quindi il consenso. Solitamente, in base alla tecnologia blockchain, scrivere un contratto non deterministico è impossibile; verrà imposto un linguaggio di programmazione che non contiene costrutti non-deterministici [18], oppure risulterà impossibile rilasciarlo. [17]
- Lo *smart contract* risiede nella blockchain, e quindi il suo codice può essere ispezionato da ogni partecipante della rete.
- Dato che ogni interazione con un contratto occorre via transazioni firmate, tutti i partecipanti hanno una traccia crittograficamente verificabile delle operazioni effettuate.

Le reti blockchain che supportano gli *smart contracts* permettono quindi transazioni molto più complesse. Gli utenti possono ispezionare il codice e verificare il risultato prima di decidere di ingaggiare il contratto, hanno la certezza dell'esecuzione del codice dato che è rilasciato sulla blockchain, e hanno una garanzia verificabile del processo dato che ogni transazione è digitalmente firmata. La possibilità di una disputa è dunque eliminata.[17]

Un costrutto particolare è quello degli oracoli, i quali permettono l'acquisizione di dati esterni alla rete blockchain. Questi dati possono essere di qualsiasi tipo: prezzi, vincitori di una maratona, meteo…

Sono utilizzati nelle transazioni che coinvolgono informazioni esterne alla rete come, utilizzando l'esempio di figura 6, "comprare due snack se entro domani piove". Uno *smart contract*, non potendo comunicare con l'esterno per motivi di sicurezza, interpella un oracolo che inserisce la risposta nella blockchain, in modo che poi la transazione sia verificabile da tutti gli utenti.

Un problema degli oracoli è il rischio di nullificare gli effetti di decentralizzazione, nel caso le informazioni siano ottenute da una sola fonte. Questo viene risolto creando oracoli decentralizzati che raccolgono informazioni da più fonti diverse; se una fonte è fallata, lo *smart contract* continuerà lo stesso il suo corretto funzionamento. [19]

4. ALCUNE APPLICAZIONI DELLE TECNOLOGIE BLOCKCHAIN NELL'AMBITO SANITARIO

Attraverso l'analisi di recenti studi [20] [21], effettuati su numerosi articoli, è stato possibile constatare come le tecnologie blockchain abbiano trovato impiego in diversi campi della sanità grazie alle innate caratteristiche di decentralizzazione, immutabilità e condivisione. In questo capitolo verranno proposti alcuni esempi di implementazione.

4.1 Fascicolo Sanitario Elettronico

L'organizzazione in maniera frammentata dei fascicoli sanitari dei pazienti è uno dei più grandi problemi nella sanità moderna. [21] Con l'integrazione di una tecnologia blockchain, i professionisti possono avere accesso, a discrezione del paziente, a tutti i dati loro necessari per svolgere al meglio il loro ruolo. Inoltre, quest'ultimi si interesserebbero di più alla loro salute in quanto sono coinvolti direttamente nel mantenimento del loro fascicolo sanitario. [20]

Una prima proposta, che considera il rapporto medico-paziente come una serie di transazioni, è quella di Gordon e Catalini. [22] Uno *smart contract* può essere progettato per creare un ecosistema sanitario che permetta l'accesso ai dati sensibili al paziente, e a chi da lui scelto. Un medico può quindi scrivere note, inserire ricette e proporre test, che verranno elaborati come una transazione. Allo stesso modo la farmacia può registrare la transazione nella blockchain quando rilascia il farmaco. Questi ecosistemi possono essere integrati con altri sviluppatori di *smart contract* per la creazione di routine di allenamento o diete, in base alle esigenze del singolo paziente. [21]

Azaria et al. [23] hanno presentato *MedRec*, un progetto del *MIT (Massachusetts Institute of Technology)* in collaborazione con l'ospedale *Beth Israel Deaconess*, che utilizza una piattaforma blockchain per dare ai pazienti l'accesso ai loro dati. I pazienti, quindi, possono decidere di fornire l'accesso al loro fascicolo sanitario ad una terza parte, riducendo così la burocrazia cartacea, dato che solitamente i primi devono portarsi appresso tutto lo storico di esami mentre sono alla ricerca di un consulto medico. [20]
Un'integrazione per la protezione della privacy nella condivisione di dati è quella avanzata da Liu et al. [24] che utilizza la piattaforma *Ethereum* per ridurre il rischio di una fuga di dati medici sensibili.

Sono stati proposti, nell'ambito del fascicolo sanitario elettronico, molti sistemi basati su tecnologie blockchain, atti a semplificarne la consultazione e migliorare la protezione dei dati; Fan et al. con *MedBlock* [25], *FHIRChain* di Zhang et al. [26] che incapsula lo "*Health Level Seven Fast Healthcare Interoperability Resources*", in breve *FHIR*, uno standard per la condivisione di dati clinici, ancora, Li et al [27] con un sistema di preservazione di dati medici basati su *Ethereum*.[20]

Come si può notare, tutte queste soluzioni sono causate dalla necessità di fronteggiare la deriva centralizzata dei sistemi sanitari, che sposta sempre di più il possesso dei dati dal paziente a terze parti, non sempre direttamente coinvolte. [21]

4.2 Assicurazione Sanitaria

Nonostante le caratteristiche della tecnologia blockchain possano aiutare nel processo di reclami assicurativi solo un articolo, avanzato da Zhou et al. [28], suggerisce una sua implementazione. Questa avviene con la proposta di *MIStore*, un sistema di archiviazione di assicurazioni sanitarie basato sulla tecnologia *Ethereum*. I dati dell'assicurazione sanitaria di un paziente possono essere criptati e immagazzinati immutabilmente nella blockchain, che ne migliora la affidabilità ed elimina la necessità di terze parti nella sua gestione.[20]

4.3 Analisi dei dati sanitari

La tecnologia blockchain, in collaborazione con altre emergenti tecnologie come il *deep-learning*, sono state usate per fornire analisi predittive di dati sanitari [20].

Kotsiuba et al. [29] afferma che queste possono dare un'unica opportunità nell'affrontare problemi riguardanti le analisi e la sicurezza dei dati medici. È stato presentato un ecosistema decentralizzato di dati sanitari, la cui confidenzialità veniva protetta realizzando un'effettiva infrastruttura di dati condivisa, aumentando quindi la base di supporto per la raccolta di dati clinici.

4.4 Contraffazione di farmaci

I farmaci contraffatti sono un grave problema del mondo moderno e mietono ogni anno da centoventimila al milione di vittime [21]. È accentuato dalla scarsa reperibilità di alcuni farmaci, che ne aumenta il costo e ne incoraggia la contraffazione. Queste tecnologie possono aiutarci a sviluppare una soluzione. [30] Impiegate con intelligenza artificiale e analisi dei dati avanzata, posso eliminare i prodotti contraffatti dalla fornitura insieme ad altri benefici: ogni farmaco

verrebbe tracciato dalla blockchain, e quindi il produttore, magazzino e anche le compagnie che si occupano della spedizione, verrebbero ricollegate all'ospedale o farmacia. Il cliente sarebbe in grado di verificarne l'autenticità e provenienza con una semplice applicazione. L'intelligenza artificiale e l'analisi dei dati renderebbero inoltre i sistemi di spedizione più efficienti, prevenendo anche la distorsione del prezzo attraverso un'efficiente distribuzione. [21]

4.5 Mercato genomico

Il mercato dei dati genomici è nuovo ed emergente. Ci si aspetta però che le possibilità nel sequenziamento personale del genoma porti alla creazione di un mercato dati del valore di miliardi di dollari. Le reti blockchain possono essere costruite per permettere alle persone di scambiare in sicurezza questi dati.[31] Queste favoriscono infatti la protezione dei dati genomici, permettendo ai compratori di procurarli in modo più efficace e sicuro, e affrontano il problema legato alla mole di dati. [21]

4.6 Fornitura farmaceutica

La fornitura di medicinali può essere monitorata in sicurezza e in modo trasparente con una tecnologia blockchain, riducendo ritardi e problemi legati all'errore umano. Può essere anche utilizzata in ogni step del processo per tracciarne prezzi, costo e anche inquinamento prodotto. Bocek et al. hanno presentato una reale dimostrazione di ciò utilizzando sensori interfacciati con una tecnologia blockchain che registrava le temperature a cui venivano tenuti e trasportati vari farmaci; queste misure erano quindi tenute immutabilmente in una blockchain pubblica per una libera e trasparente consultazione.[32] Dato che la legge Europea impone la dimostrazione che i medicinali non siano stati esposti a condizioni specifiche, in particolar modo certe temperature, che ne comprometterebbero la qualità, una tale tecnologia potrebbe portare ad una sensibile riduzione dei costi, automatizzando il processo di controllo.[20] [21]

4.7 Ricerca clinica

La tecnologia blockchain ha trovato vasto impiego anche nell'ambito della ricerca clinica.

Nugent et al. [33] propongono una soluzione basata su *smart contracts* per prevenire la falsificazione di dati e la mancata dichiarazione di particolari risultati clinici, aumentando così la fiducia nei test clinici.

Angeletti et al. [34] hanno avanzato un modello concettuale per la tracciabilità del consenso in test clinici. Viene effettuato il *timestamp* del consenso, e successivamente incluso in una

blockchain basata su *Ethereum*, rendendolo così pubblico. Tutti i piani, protocolli, consensi e possibili risultati possono essere inseriti nella blockchain prima ancora dell'effettivo test clinico prevenendo così episodi di corruzione o risultati indesiderati.[20]

Quello di Kleinakei et al. [35] è invece un servizio di notarizzazione basato su blockchain che impiega degli *smart contract* per certificare *query* di database biomedici e rispettivi risultato, garantendone trasparenza.

Maslove et al. [36] invece hanno proposto *BlockTrial*, un interfaccia web che permette agli utenti di eseguire *smart contract* relativi a test clinici sulla piattaforma *Ethereum*. Tramite l'utilizzo di oracoli i pazienti forniscono l'accesso ai loro dati, che risiedono al di fuori della blockchain, permettendo ai ricercatori la possibilità di interrogarli con *query*. Tutte queste transazioni vengono impresse nella blockchain aumentando l'affidabilità dei dati raccolti, e permettendo ai pazienti una partecipazione attiva nella ricerca.

4.8 Monitoraggio remoto del paziente

Il monitoraggio da remoto di un paziente consiste generalmente nella raccolta di dati biomedicali dal corpo o da dispositivi medici portatili per il monitoraggio dello stato al di fuori di luoghi addetti a tale scopo, come la propria abitazione.

Liang et al. [37], tramite un'implementazione basata *sull'Hyperledger* (che considera le tecnologie blockchain come dei moduli *plug and play*) nel settore *mHealth* (sanità mobile), propongono un metodo che permette la raccolta di dati e la loro condivisione con chi di interesse, assicurando sia la trasparenza che l'accessibilità. Saravan et al. [38] invece hanno presentato un sistema *end-to-end*, che attraverso degli *smart contract* migliora la sicurezza crittografica e formalizza l'accesso ai dati su cui effettuare monitoraggi. L'autore afferma che una tecnologia blockchain è stata impiegata in una realizzazione di un dispositivo portatile di assistenza per pazienti malati di diabete.

Ichikawa et al. [39] hanno avanzato a loro volta un sistema che raccoglie dati del paziente, tramite un dispositivo mobile, per poi inserirli in una rete blockchain basata su *Hyperledger*, che ne garantisce la sicurezza.

Cichozs et al. [40] presentano invece un sistema con contratti a firma multipla, che permette a pazienti malati di diabete di trasferire i parametri vitali, raccolti attraverso un sensore, a una piattaforma blockchain, dove vengono collezionati, immagazzinati e analizzati. In casi di emergenza i pazienti, o chi ne fa le veci, viene allertato tramite social.[20]

4.9 Soccorso in situazioni di emergenza

La tecnologia blockchain può permetterci di enfatizzare la collaborazione durante una situazione di emergenza. [41] Informazioni false o poco chiare da tutte le fonti disponibili oggigiorno possono compromettere una forte e decisiva risposta nel fornire aiuto. [42]
Una rete blockchain può significativamente ridurre la complessità, e rendere più semplice ed efficaci le collaborazioni tra più fonti.[21]

4.10 Gestione credenziali dello staff medico

Il processo di certificazione dei professionisti in ambito medico può richiedere uno spreco di tempo e soprattutto di risorse, attraverso il passaggio di chiamate, fax ed e-mail.
Tramite l'impiego di uno *smart contract* come un database "vivente" le credenziali possono essere continuamente modificate dal continuo inserimento o da aggiornamenti delle stesse. Non sarebbe più necessario cercarle, generalmente in un periodo di tempo abbastanza dilatato, in diversi archivi ognuno con una locazione diversa. Sarebbero tutte contenute nello *smart contract* dell'utente, a patto che questo lo tenga aggiornato, e disponibili ad una semplice ed efficace consultazione. Un approccio di questo tipo risparmierebbe indubbiamente tempo e risorse. [21]

4.11 Integrazione con intelligenza artificiale

Una tecnologia blockchain, integrata con un'intelligenza artificiale, può creare un potente strumento di diagnosi, migliorando anche la qualità del trattamento. Il lavoro da diretto diventa predittivo, ottenendo così dei servizi ottimali costruiti su misura caso per caso.
Prendiamo un case study basato sul COVID-19.
Le pratiche cliniche del COVID-19 utilizzano tecnologia blockchain. Con la combinazione tra questa e l'intelligenza artificiale, i pazienti affetti da covid verrebbero diagnosticati e curati in fretta, aiutando così la costruzione di procedure terapeutiche raccomandabili in futuri casi (virus paragonabili al COVID-19). Una rete blockchain permetterebbe la condivisione di dati da laboratori clinici a ospedali, o ad altri professionisti, mantenendo sempre la privacy e sicurezza mentre un'intelligenza artificiale verrebbe impiegata per l'analisi dei dati, aiutando lo sviluppo di nuovi farmaci e terapie adatte.[21]

5. CONCLUSIONI

Per poter trattare gli usi delle tecnologie blockchain in campo biomedico e sanitario è stato necessario fornire prima una breve introduzione storica, per contestualizzarne la nascita, e poi un'esposizione, a carattere generale, dei suoi componenti e delle sue funzioni. Si è potuto quindi analizzare, nel capitolo 3, come la natura stessa della tecnologia garantisca l'immutabilità e la condivisione dei dati contenuti, mentre i modelli di consenso impiegati permettano la decentralizzazione di questi, rendendo il loro sabotaggio o alterazione proibitivi dal punto di vista del costo in risorse. A tutto ciò si aggiungono gli *smart contracts*; veri e propri *script* la cui esecuzione permette l'attuazione di transazioni complesse mantenendo le proprietà sopracitate.

Nel settore biomedico e sanitario tutto ciò può trovare un vasto impiego come si può evincere dai numerosi articoli riportarti in questo elaborato. Tali studi dimostrano come queste caratteristiche possano essere sfruttate in possibili implementazioni per rendere alcune tipologie di informazioni più trasparenti, proteggere la privacy del paziente e fornire un sistema di gestione dati sanitari decentralizzato, dove ogni utente è il proprietario dei suoi dati. [20] [21]

Tuttavia, la maggior parte degli studi non viene applicata oltre lo stadio concettuale facendo pertanto supporre un'immaturità della tecnologia per realizzazioni concrete in ambito sanitario e biomedicale. [21]

Le tecnologie blockchain, date le loro grandi potenzialità e possibili applicazioni, rimangono un importante settore di studio e ricerca per gli anni a venire, ma il loro attuale utilizzo rimane circoscritto al settore finanziario. [2] [3] [21]

BIBLIOGRAFIA

[1] S. Nakamoto “Bitcoin: A Peer to Peer Electronic Cash System”, 2008.

[2] D. Yaga, P. Mell, N. Roby, K. Scarfone “Blockchain Technology Overview”, 2018.

[3] I. Bashir “Mastering Blockchain”, 3rd Edition, 2020 Packt Publishing.

[4] A. T. Sherman, F. Javani, H. Zhang and E. Golaszewski, “On the Origins and Variations of Blockchain Technologies” in *IEEE Security & Privacy*, vol. 17, pp. 72-77, Jan.-Feb. 2019.

[5] A. Back, “Hashcash: A denial of service counter-measure”, Aug. 2002.

[6] A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder “Bitcoin and Cryptocurrency Technologies”, 2006, Princeton University Press.

[7] J. Aumasson “Serious Cryptography: A Practical Introduction to Modern Encryption”, 2017, No Starch Press.

[8] D. Chaum “Computer systems established, maintained, and trusted by mutually suspicious groups”, Jun. 1982.

[9] C. Dork, M. Naor “Pricing via Processing or Combatting Junk Mail” in *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, Aug. 1982.

[10] S. Haber, W. S. Scornetta “How to Time-Stamp a Digital Document”, in *J. Cryptology* 3, 99–111, 1991.

[11] D. Bayer, S. Haber, W. S. Scornetta “Improving the Efficiency and Reliability of Digital Time-Stamping”, 1993.

[12] D. Drescher, “Basics of Bitcoin”, 2017, Apress.

[13] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei and C. Qijun, "A review on consensus algorithm of blockchain" in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 2567-2572.

[14] N. Szabo “The Idea of Smart Contracts”, 1997.

[15] V. Buterin “Thoughts on UTXOs” 2016.

[16] C. Cachin, S. Schubert, M. Vukoli¢ “Non-determinism in Byzantine fault-tolerant replication”, disponibile: http://arxiv.org/abs/1603.07351

[17] K. Christidis, M. Devetsikiotis “Blockchains and Smart Contracts for the Internet of Things”, in *IEEE Access*, vol. 4, pp. 2292-2303, 2016.

[18] Documentazione Solidity, disponibile: https://docs.soliditylang.org/

[19] Documentazione blockchain Etherium, disponibile: https://ethereum.org/

[20] D. Elangovan et Al "The Use of Blockchain Technology in the Health Care Sector: Systematic Review", *JMIR Med Inform* 2022; 10(1): e17278.

[21] Q. Mamun "Blockchain Technology in the Future of Healthcare" *Smart Health 23*, 2022, 100223.

[22] W. J. Gordon C. Catalini "Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability" in *Computational and Structural Biotechnology Journal*, vol. 16, pp. 224-230, 2018.

[23] A. Azaria, A. Ekblaw, T.Vieira, A. Lippman. "MedRec: Using blockchain for medical data access and permission management" in *Proceedings of the 2nd International Conference on Open and Big Data*, 2016.

[24] J. Liu, X. Li, L. Ye, H. Zhang, X. Du, M. Guizani "BPDS: A blockchain based privacy-preserving data sharing for electronic medical records" in *Proceedings of the 2018 IEEE Global Communications Conference*, 2018.

[25] K. Fan, S. Wang, Y. Ren, H. Li, Y. Yang "MedBlock: Efficient and secure medical data sharing via blockchain", *J Med Syst* 2018; 42(8): 141.

[26] P. Zhang, J. White, D. C. Schmidt, G. Lenz, S. Rosenbloom "FHIRChain: Applying blockchain to securely and scalably share clinical data", *Computational and Structural Biotechnology Journal* Volume 16, 2018, Pages 267-278.

[27] H. Li, L. Zhu, M. Shen, F. Gao, X. Tao, S. Liu "Blockchain-based data preservation system for medical data", *J Med Syst* 2018; 42(8): 141.

[28] L. Zhou, L. Wang, Y. Sun "MIStore: A blockchain-based medical insurance storage system", *J Med Syst* 2018; 42(8): 149.

[29] I. Kotsiuba, A. Velvkzhanin, Y. Yanovich, I. Bandurova, Y. Dyachenko, V. Zhygulin "Decentralized e-Health architecture for boosting healthcare analytics" in *Proceedings of the 2nd World Conference on Smart Trends in Systems, Security and Sustainability,* 2019.

[30] R. Kumar, R. Tripathi "Traceability of counterfeit medicine supply chain through Blockchain", *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*, 2019, pp. 568-570.

[31] E. De Cristofaro "Genomic privacy and the rise of a new research community", *IEEE security & privacy*, 2014, Vol. 12, pp. 80–83.

[32] T. Bocek, B. Rodrigues, T. Strasser, B. Stiller "Blockchains everywhere - A use-case of blockchains in the pharma supply-chain" in *Proceedings of the IFIP/IEEE Symposium on Integrated Network and Service Management*, 2017.

[33] T. Nugent, D. Upton, M. Cimpoesu "Improving data transparency in clinical trials using blockchain smart contracts", *F1000Research*, 2016; 5: 2541.

[34] F. Angeletti, I. Chatzigiannakis, A. Vitaletti “The role of blockchain and IoT in recruiting participants for digital clinical trials” in *Proceedings of the 25th International Conference on Software, Telecommunications and Computer Networks*, 2017.

[35] A. Kleinaki, P. Mytis-Gkometh, G. Drosatos, P. S. Efraimidis, E. Kaldoudi “A blockchain-based notarization service for biomedical knowledge retrieval”, *Computational and Structural Biotechnology Journal*, Volume 16, 2018, Pages 288-297.

[36] D. M. Maslove, J, Klein, K. Brohman, P. Martin “Using blockchain technology to manage clinical trials data: A proof-of-concept study”, *JMIR Med Inform* 2018; 6(4): e11949.

[37] X. Liang, J. Zhao, S. Shetty, J. Liu, D. Li “Integrating blockchain for data sharing and collaboration in mobile healthcare applications” in *Proceedings of the 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications*, 2017.

[38] M. Saravanan, R. Shubha, A. Mary “SMEAD: A secured mobile enabled assisting device for diabetics monitoring” in *Proceedings of the 11th IEEE International Conference on Advanced Networks and Telecommunications Systems*, 2018.

[39] D. Ichikawa, M. Kashiyama, T. Ueno “Tamper-resistant mobile health using blockchain technology”, *JMIR Mhealth Uhealth*, 2017, Jul 26; 5(7): e111.

[40] S. L. Cichosz, M. N. Stausholm, T. Kronborg, P. Vestergaard, O. Hejlesen “How to use blockchain for diabetes health care data and access management: An operational concept”, *Journal of Diabetes Science and Technology*, 2019, pp. 248-253.

[41] D. A. Aranda, L. M. M. Fernandez, V. Stantchev “Integration of Internet of Things (IoT) and Blockchain to increase humanitarian aid supply chains performance” in *2019 5th international conference on transportation information and safety (ICTIS)*, pp. 140-145.

[42] N. A. M. Labonte "Challenges in humanitarian information management and exchange: Evidence from Haiti", *Disasters* vol. 3, pp. S50-S72, 2014.