Università degli Studi di Padova

Dipartimento di Diritto Pubblico, Internazionale e Comunitario
Dipartimento di Matematica

Corso di Laurea in *Diritto e Tecnologia*
Anno Accademico 2022 / 2023

Titolo tesi:
I PROTOCOLLI DI CONSENSO NELLE DIFFERENTI IMPLEMENTAZIONI DELLA TECNOLOGIA BLOCKCHAIN

Relatore:
Ph.D, Alessandro Brighente

Studente:
Manuele Marsich

Indice

1. Introduzione

Con blockchain (in italiano, "catena di blocchi") ci si riferisce ad un particolare tipo di struttura dati informatica in grado di organizzare in "blocchi" (concatenati in ordine cronologico) delle informazioni, creando in questo modo un registro di dati condiviso [24].

Alcune delle caratteristiche tipiche della blockchain, quali la decentralizzazione, l'immutabilità assoluta, la sicurezza e la tracciabilità, rendono l'utilizzo di tale tecnologia una soluzione promettente in innumerevoli ambiti lavorativi e settori [24], come ad esempio quello della finanza digitale, della sanità pubblica (molti ospedali e database di sistemi sanitari in diversi paesi già sfruttano la tecnologia blockchain per tenere traccia dei dati dei pazienti e dei loro referti [23]) e dell'industria globale in generale. Negli ultimi anni è infatti diventata una parte sempre più importante del mondo accademico, delle società fintech e più in generale di tutte quelle realtà che si occupano di transazioni digitali e che hanno interesse a mantenere una traccia immutabile di informazioni, registrata permanentemente in maniera sicura e legittima, senza la necessità di fiducia pregressa tra i partecipanti alla realtà in questione [25].

Per poter raggiungere tali caratteristiche è necessario che la blockchain sia sorretta da meccanismi in grado di garantire con efficacia il consenso fra i diversi "attori" partecipanti. Esistono diversi tipi di protocolli in grado di farlo, ciascuno con i propri vantaggi e svantaggi, ma l'esistenza stessa di un algoritmo di consenso è necessaria per proteggere la blockchain e garantire l'affidabilità dei dati che si trovano al suo interno: in concreto, un algoritmo di consenso non è altro che la modalità grazie al quale ogni *nodo* della rete blockchain (ossia le persone partecipano alla sua creazione o modifica) si accorda su come inserire nuovi dati all'interno della rete [14].

La blockchain infatti non viene in genere strettamente utilizzata da una singola azienda o individuo, poiché funziona grazie alla cooperazione di più partecipanti. Ognuno di essi prende parte alla sicurezza della rete, garantendo che ogni nuova informazione inserita sia valida, che ogni nuova interazione con i blocchi preesistenti sia possibile, ed in particolare, che nessun partecipante tenti di compiere frodi per proprio tornaconto.

Lo scopo di questa relazione è quindi quello di presentare quali sono le principali tecnologie blockchain al momento esistenti, prestando particolare attenzione alle differenti implementazioni dei protocolli che si occupano di raggiungere e assicurare un consenso distribuito all'interno delle stesse.

Nello specifico, verrà descritta l'origine del concetto di blockchain e del concetto di consenso all'interno delle tecnologie a registro distribuito, prendendo in esame il *Proof of Work* (PoW), il *Proof of Stake* (PoS) con i suoi derivati ed il *Proof of Capacity* (PoC), prima analizzandoli singolarmente ed infine confrontandoli fra loro.

2. La tecnologia blockchain

In questo capitolo verrà illustrata una panoramica generale sulla tecnologia blockchain. Nella sezione 2.1 verrà descritta l'origine dei concetti stessi di "sistema distribuito" e "catena di blocchi". In seguito, nella sezione 2.2 verrà fornita una descrizione specifica sulla composizione dei singoli blocchi che ne fanno parte, fornendone quindi una definizione più tecnica. Infine, nella sezione 2.3 verranno presentati i principi crittografici alla base della tecnologia in questione.

2.1 Origine

L'origine del concetto di *catena di blocchi* può essere fatta risalire già al 1991, quando il fisico e ricercatore scientifico Wakefield Scott Stornetta ed il crittografo Stuart Haber scrissero e pubblicarono un articolo denominato "How to time-stamp a digital document" [2].

All'interno della loro ricerca, Stornetta e Haber si occuparono di teorizzare un meccanismo per garantire l'immutabilità delle informazioni e dei dati conservati su sistemi digitali. La soluzione che proponevano era di utilizzare un *Time Stamping Service* (TSS), il quale, tramite l'utilizzo di funzioni di hash, firme digitali ed il concatenamento di ogni documento a quello precedente, avrebbe potuto verificare e confermare l'originalità di tutti i documenti stessi [2].

Grazie a questa arcaica teorizzazione di un sistema blockchain, nel 1998 l'informatico Nick Szabo iniziò a lavorare sull'idea di una valuta decentralizzata, che chiamò *Bit Gold*. Infatti Szabo scrisse (senza mai pubblicarlo ufficialmente) nel 2005 un *whitepaper* [3] in cui proponeva l'utilizzo di una catena di dati legati algoritmicamente tra loro tramite

hashing, associandovi inoltre la data e l'ora in cui avvenivano gli scambi di denaro. Lo scopo era quello di risolvere il problema del *double spending* nelle transazioni informatiche: tutti gli scambi di valuta, infatti, se passanti attraverso un sistema centrale controllato da una autorità (come una banca, ad esempio) per forza di cose potrebbero essere modificati dall'autorità stessa, e di conseguenza lo stesso denaro potrebbe essere speso più di una volta. L'idea di *Bit Gold*, tuttavia, venne solo teorizzata, senza mai essere realizzata in maniera pratica.

Nel 2000, Stefan Konst propose in un documento di ricerca intitolato "Secure Log Files Based on Cryptographically Concatenated Entries" una ulteriore idea per migliorare le già teorizzate in passato catene di blocchi: il modello prevedeva che le singole voci nella catena venissero ricondotte crittograficamente alla sua genesi (o primo blocco), in modo da dimostrarne l'autenticità [4]. Si tratta dello stesso modello che vediamo oggi implementato in quasi tutte le blockchain, tra cui quella della famosa criptovaluta *Bitcoin*. Satoshi Nakamoto, infatti, nella pubblicazione del relativo *whitepaper* nel 2008 [1], si è avvalso di tutte le idee teorizzate nei decenni precedenti, dando vita alla prima catena di blocchi pubblica della storia.

2.2 Caratteristiche generali

Le tecnologie solitamente utilizzate nei database tradizionali presentano diverse problematiche dal punto di vista della registrazione di transazioni finanziarie. Ad esempio, si consideri la vendita di un immobile. Una volta che il denaro viene scambiato, la proprietà del bene passa all'acquirente: ognuna delle due parti potrà, individualmente, documentare in registri propri lo scambio monetario, tuttavia nessuna delle due fonti verrà considerata veramente attendibile. In una situazione di questo tipo infatti, il compratore

potrebbe essere tentato di affermare di aver effettuato il pagamento, pur non avendo di fatto inviato il denaro al venditore.

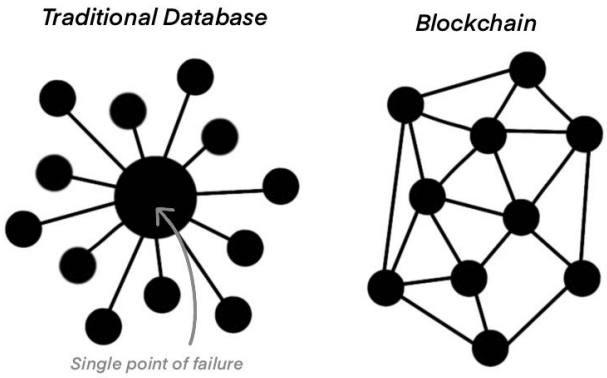


Figura 2.2: Un database tradizionale confrontato con la blockchain

La soluzione che tradizionalmente viene adottata è quella di inserire all'interno del paradigma una autorità centrale, che si occuperà di controllare e validare la transazione. Al tempo stesso, l'introduzione di una autorità terza genererà, per forza di cose, un *single point of failure:* se questa terza parte dovesse venire compromessa o corrotta, la sicurezza ed imparzialità del contratto verrebbe meno.

La tecnologia blockchain, in quanto basata su registro condiviso, distribuito ed immutabile, si predispone di risolvere questi problemi. Si tratta infatti di una alternativa al paradigma centralizzato tradizionale: tutte le transazioni, per poter essere considerate definitive, devono per forza essere approvate dagli utenti (i cosiddetti nodi) facenti parte della blockchain stessa. I registri di ognuno dei partecipanti a questa rete *peer-to-peer* vengono costantemente aggiornati in tempo reale, in modo da risultare sempre identici tra loro.

Essendo necessaria l'approvazione della maggioranza dei nodi (coloro che verificano le transazioni) per effettuare modifiche al registro distribuito, ed essendo tutti i nodi equivalenti fra di loro, viene in questo modo risolto il

problema del *single point of failure.* Nella figura 2.2 è possibile notare la differenza fra i collegamenti dei nodi di una rete blockchain e quelli invece di un database tradizionale.

Di conseguenza, è possibile sintetizzare le principali caratteristiche di una blockchain nei seguenti punti:

- *Tecnologia a registro distribuito*

  Le informazioni sono distribuite e registrate su più nodi per garantire la sicurezza informatica e la resilienza del sistema. È possibile quindi elaborare le transazioni senza intermediari, ossia senza la presenza di un'autorità centrale fidata.

- *Immutabilità*

  Essendo pensata come una evoluzione del concetto di *libro mastro*, è possibile aggiungere nuovi blocchi di informazioni in qualsiasi momento, ma i blocchi precedentemente inseriti e già verificati dai nodi facenti parte al network non possono essere modificati o cancellati. Ciò significa che le informazioni scritte sulla blockchain lasciano una "impronta" immutabile e sempre verificabile.

- *Tracciabilità*

  È sempre possibile risalire, operando a ritroso, all'origine di ogni transazione (in quanto ognuna di esse è parte di un blocco inscritto per sempre nella catena stessa).

- *Sicurezza*

  La tecnologia blockchain produce una struttura dati con qualità di sicurezza intrinseche nella stessa. Come osservato, viene fatto largo uso di principi di crittografia, decentralizzazione e consenso: tutti questi

elementi garantiscono che vi sia fiducia nella affidabilità e immutabilità delle transazioni.

- Crittografia → ogni nuovo blocco si collega crittograficamente a tutti i blocchi precedenti, formando una catena impossibile da manomettere.
- Consenso → tutte le transazioni all'interno dei blocchi sono convalidate e concordate da un meccanismo di consenso, il quale si occupa di verificare che ogni transazione venga effettuata in maniera corretta e possa essere considerata definitiva.

2.3 Com'è fatta una blockchain

Una blockchain è una struttura dati (che può essere sia pubblica che privata) distribuita che viene replicata e condivisa tra i membri di una rete. Come visto, è stata implementata per la prima volta con la distribuzione del *whitepaper* di Bitcoin da parte di Satoshi Nakamoto [1], il cui intento principale era quello di risolvere il problema della doppia spesa nei sistemi digitali [6].

I nodi della rete Bitcoin hanno il compito di appendere alla catena di blocchi solamente delle transazioni che sono effettivamente state validate e concordate dai membri della rete stessa, e per questo motivo, la decentralizzazione diventa l'autorità: chi possiede cosa è determinato solo dalla blockchain stessa.

Una blockchain è quindi un registro digitale distribuito, in grado di inscrivere permanentemente dei dati in modo sicuro e verificabile. Questi dati infatti non sono modificabili in alcun modo una volta inseriti all'interno della blockchain, tranne nel caso in cui si voglia modificare anche tutti i blocchi successivi: ciò necessiterebbe del consenso della maggioranza della rete.

Le unità più piccole all'interno di una blockchain sono le singole transazioni, le quali non sono altro che un insieme di dati contententi le informazioni sullo scambio monetario [11].

Le singole transazioni vengono poi raggruppate e inserite in una struttura dati specifica, il blocco, che contiene:

1. La dimensione del blocco: variabile nelle differenti implementazioni blockchain.
2. Un *header*: ossia l'intestazione del blocco, consiste in diversi campi, tra cui il *timestamp* (o marca temporale) e l'*hash* del blocco precedente.
3. Un *transaction counter*: cioè un contatore delle transazioni del blocco.
4. Le transazioni stesse: la parte fondamentale del blocco, costituito da un insieme di transazioni.

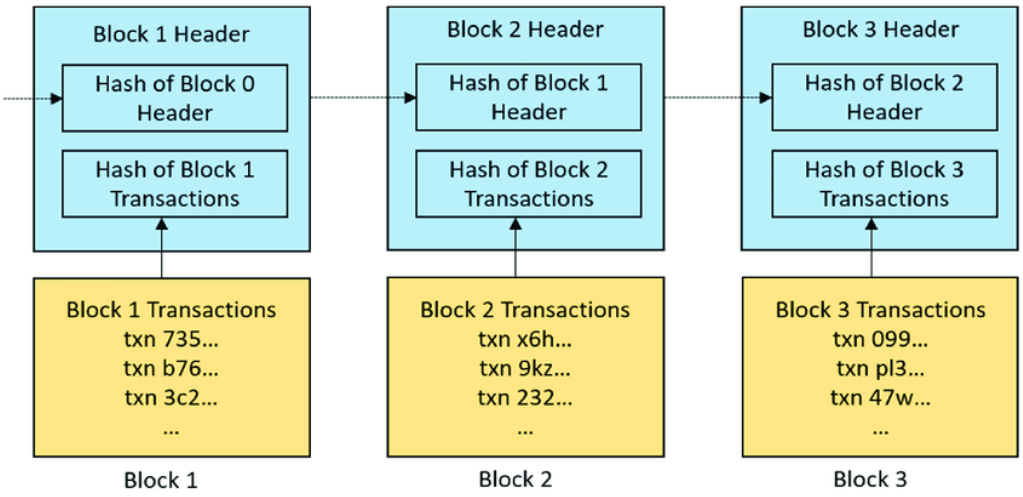


Figura 2.3: Un esempio di come i blocchi sono concatenati tra loro per formare la blockchain

Il primo blocco di una blockchain è convenzionalmente denominato *genesis block*, ossia blocco genesi: si tratta dell'unico blocco dove l'*header* non presenta riferimenti a blocchi precedenti [11].

Di conseguenza, la blockchain può essere vista come un elenco di blocchi protetti dalla crittografia: un blocco può avere una o più transazioni associate ed ognuno di essi (tranne quello *genesis*) contiene anche un puntatore all'*hash* del blocco precedente, oltre che un riferimento temporale riguardo il momento in cui è stato inscritto nella catena [7]. La figura 2.3 mostra una versione semplificata della catena di blocchi che si crea una volta che questi ultimi vengono concatenati uno con l'altro.

2.4 La crittografia nella blockchain

All'interno di una blockchain, la sicurezza dei dati e delle informazioni inserite viene garantita da tecniche di crittografia già esistenti nel panorama informatico, quali l'utilizzo di funzioni di hash e di una coppia di chiavi (pubblica e privata), cioè di crittografia asimmetrica.

In generale, viene fatto largamente uso di crittografia nei protocolli utilizzati dalle criptovalute, infatti essa garantisce anche che le transazioni siano anonime, sicure e *trust-less* (cioè senza fiducia necessaria nei confronti dell'interlocutore) [8].

Tutte le transazioni registrate nella blockchain sono infatti criptate e possono essere decifrate solo dai loro destinatari. In questo modo, la blockchain rende impossibile la decrittazione da parte di persone non autorizzate, quindi non è necessario implementare ulteriori sistemi di sicurezza speciali (oltre all'utilizzo della coppia di chiavi e delle funzioni di hash) per "proteggere" tali dati.

2.4.1 Funzioni di hash

Con *hashing* si intende la conversione di dati (input) attraverso una funzione matematica in un *hash* (output) che ha sempre la stessa lunghezza finale, indipendentemente dall'input. Se un piccolo dettaglio dell'input cambia, l'output cambia completamente: il fatto che due input diversi non possano mai dare lo stesso output (cioè che non ci possa essere collisione) è un forte fattore di sicurezza. Le funzioni hash sono progettate per essere *one-way*, cioè unidirezionali. L'unico modo per riprodurre l'input dall'output di una funzione hash è quello di procedere per tentativi, esaminando tutte le combinazioni possibili. Una procedura di questo tipo richiede miliardi di tentativi di *try and error* ("prova ed errore"), ed è praticamente impossibile da attuare anche utilizzando i computer più moderni e potenti.

Gli algoritmi di hash trovano il loro utilizzo in molteplici applicazioni informatiche, in particolare vengono utilizzati per analizzare l'integrità di dati importanti, oltre che per la creazione e verifica di firme digitali, la verifica di credenziali di login oppure il reset sicuro di password per l'accesso a database o siti internet [10].

2.4.2 Crittografia assimetrica

La crittografia asimmetrica viene solitamente utilizzata per trasmettere in maniera sicura un messaggio tra un mittente ed un destinatario. Ognuna delle due parti possiede sia una chiave privata, la quale è nota solo al proprietario stesso, che una chiave pubblica, che è invece di pubblico dominio.

I dati cifrati con una chiave privata possono essere decifrati solo con la relativa chiave pubblica e viceversa. Di conseguenza, tutti sono in grado di cifrare un messaggio utilizzando la chiave pubblica del destinatario previsto

(essendo essa di pubblico dominio e quindi consultabile da chiunque), ma solamente quest'ultimo sarà in grado di decifrarlo [9].

La crittografia asimmetrica è inoltre fondamentale anche nel verificare l'identità del mittente del messaggio. Per fare ciò si utilizza infatti la chiave pubblica del mittente stesso: solamente se il messaggio è stato scritto con la sua chiave privata sarà possibile decifrarlo.

Per quanto riguarda la blockchain, il sistema della coppia di chiavi viene utilizzato dalle reti di criptovalute, le quali assegnano ad ogni singolo utente una chiave privata, dalla quale viene generata tramite crittografia una chiave pubblica collegata. Quest'ultima è l'unica informazione necessaria a ricevere una transazione, e può essere condivisa con chiunque. Viceversa, la chiave privata deve rimanere tale, in quanto permette di spendere i fondi disponibili nel *wallet*, ossia il portafoglio digitale [8].

3. Il concetto di "consenso"

In informatica quando si parla di consenso ci si riferisce al raggiungimento di un accordo tra più parti riguardo dei singoli dati, spesso necessari in operazioni di computazione. Infatti, uno dei problemi principali dei sistemi di calcolo distribuiti (o multi-agente) è il raggiungimento di una affidabilità complessiva del sistema (e quindi del consenso) in presenza di un gran numero di processi difettosi oppure di partecipanti malevoli ed inaffidabili [14].

Nelle blockchain, il consenso è necessario in quanto permette di definire quale blocco concatenare come successivo alla catena. Questo processo viene denominato “validazione” dei nuovi blocchi, e viene assegnato ad un nodo specifico, scelto grazie ad un protocollo di consenso distribuito (che varia nelle differenti implementazioni della blockchain) [13]. Il gestore di questo nodo, che riceve una ricompensa per il lavoro di validazione effettato, è la parte più importante dell’intero paradigma: le sue decisioni riguardo quali blocchi inserire ed in quale ordine farlo impattano direttamente la sicurezza, l’affidabilità e la coerenza dei dati (e quindi delle transazioni) che si trovano all’interno del registro distribuito. La presenza stessa di una ricompensa è infatti un incentivo per tutti i nodi partecipanti a mantenere un comportamento benevolo e corretto [13].

Il raggiungimento automatico di un accordo fra i vari nodi è chiaramente fondamentale in ogni blockchain pubblica, in quanto come visto permette a quest’ultima di essere *self-auditing*, tuttavia i casi in cui un accordo tra diversi nodi indipendenti risulta necessario si estendono ben oltre la tecnologia blockchain e risalgono a decenni prima dell’invenzione di Bitcoin [12].

In un sistema distribuito, infatti, i nodi sono ripartirti sulla rete. Alcuni di questi nodi potrebbero smettere di funzionare del tutto (è la situazione del crash fault) o iniziare a comportarsi in maniera anomala o fraudolenta (Byzantine Fault, o "problema dei generali bizantini"). Più nello specifico: ci troviamo di fronte ad *n* processi, dei quali *m* potrebbero essere fallaci [19].

In uno scenario del genere diventa ovviamente non banale arrivare ad una decisione comune. Le conseguenze prodotte da una situazione di questo tipo sono facilmente osservabili in un qualsiasi sistema distribuito, come ad esempio una banale rete locale di computer. In questi casi, il guasto (ma anche la manomissione intenzionale) di un componente spesso porta ad uno stato conflittuale dei dati salvati nelle diverse parti della rete, cioè ad una situazione di incoerenza dello stato generale delle macchine interconnesse: quale di questi stati è da considerarsi valido?

Di conseguenza, è facile comprendere come il problema del consenso si potrebbe verificare in un qualunque sistema digitale e non si tratta di una peculiarità delle strutture blockchain.

Il compito dei protocolli di consenso è proprio quello di rendere i processi funzionanti concordi tra loro, anche in presenza di processi fallaci.

3.1 Il problema dei generali bizantini

Con "problema dei generali bizantini" ci si riferisce ad un problema informatico, teorizzato per la prima volta in un omonimo articolo dai matematici Leslie Lamport, Marshall Pease e Robert Shostak nel 1982, riguardante il raggiungimento del consenso nei sistemi distribuiti [15].

Gli autori formularono il problema esponendolo sotto forma di metafora: si presupponga la presenza di diversi generali bizantini (da qui il nome del problema), collocati in differenti aree strategiche ed in procinto di assediare una città nemica. [16]

Un attacco coordinato da tutte le parti in gioco comporterebbe sicuramente la conquista della città. Viceversa, un attacco non coordinato porterebbe certamente alla sconfitta.

I generali hanno la possibilità di comunicare tra loro per organizzare e decidere come sferrare l'attacco finale solo mediante l'uso di messaggeri, in quanto le truppe a loro sottoposte si trovano sparse nell'area geografica adiacente.

È evidente come fra questi messaggeri vi potrebbero essere dei traditori, i quali sicuramente avrebbero la possibilità di trasmettere agli altri generali una informazione sbagliata o che contraddice persino l'ordine che gli è stato dato.

Il problema risiede, dunque, nella facoltà di portare avanti l'attacco in modo efficace nonostante il rischio di tradimento [17].

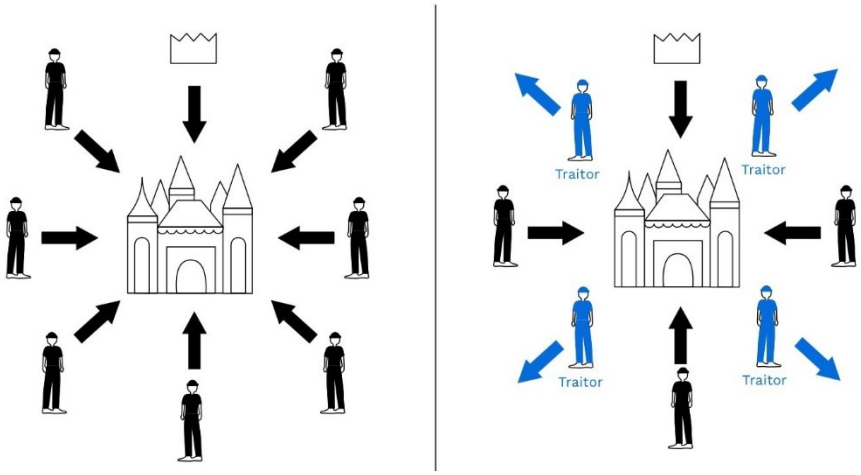


Figura 3.1: una esemplificazione grafica del problema

Inoltre è necessario considerare che, anche quando non ci si trovi in presenza di traditori, non per forza tutti i generali avranno la stessa opinione su come eseguire l'attacco.

Solo nella migliore delle ipotesi infatti i generali saranno concordi nell'ordine da impartire: attaccare o ritirarsi. Nella maggioranza dei casi, purtroppo, il messaggio non sarà così coordinato [18]. Nella figura 3.1 è possibile osservare dal punto di vista grafico come potrebbero comportarsi i generali.

È di conseguenza necessario cercare una soluzione che:

- Sia in grado di rendere l'intero insieme dei generali concorde sul medesimo piano di azione da seguire (attaccare o ritirarsi).
- Assicuri che l'esecuzione del piano non possa venire messa in discussione o impedita in presenza di messaggeri o generali traditori.

Dal punto di vista matematico è facile capire come occorrano almeno $3 * n + 1$ generali onesti per avere la meglio su n generali traditori. Nell'articolo originale di Lamport, Shostak e Pease viene infatti dimostrato che non esiste soluzione al problema se più del 33% dei generali sono traditori [15].

Il problema dei generali bizantini fa parte di un più generale insieme di problemi, composto da tutte quelle situazioni e casistiche nelle quali vi è la necessità di trovare un accordo o decisione distribuita in un sistema digitale-informatico [17].

Si tratta infatti di un problema di consenso: quando una rete di computer è in grado di raggiungere quest'ultimo anche in presenza di guasti e nodi traditori, essa può essere definita come "Byzantine Fault Tolerant", cioè tollerante ai guasti di tipo bizantino.

Quello dei generali bizantini è un dilemma che è stato ampiamente studiato nel tempo, a partire dagli anni '90 con l'introduzione dell'algoritmo di consenso *Practical Byzantine Fault Tolerance* (pBFT), il quale è stato in seguito ottimizzato con una serie diversificata di soluzioni. Tuttavia solamente grazie a Satoshi Nakamoto e all'introduzione nel *whitepaper* di *Bitcoin* del *Proof of Work (PoW)*, il preminente algoritmo di consenso distribuito utilizzato per le criptovalute, è stato possibile considerare il problema definitivamente risolto.

Come già visto, l'obiettivo degli algoritmi di consenso è proprio quello di assicurare la stabilità e l'affidabilità del sistema blockchain, facendo in modo di trovare e validare una decisione unanime riguardo i dati che verranno inseriti all'interno della rete [17].

3.1.1 Practical Byzantine Fault Tolerance (pBFT)

Il *Practical Byzantine Fault Tolerance* è stato uno dei primi tentativi di risoluzione del problema dei generali bizantini. Si tratta di un algoritmo di consenso che venne introdotto e descritto per la prima volta da Miguel Castro e Barbara Liskov in un articolo accademico pubblicato nel 1999 [20].

Lo scopo del *Practical Byzantine Fault Tolerance* era quello di raggruppare in un meccanismo consolidato le singole soluzioni (preesistenti all'epoca) volte a raggiungere il consenso distribuito. Una forma migliorata dello stesso protocollo di consenso è stata implementata anche in diversi moderni sistemi informatici distribuiti, comprese alcune popolari piattaforme blockchain come quella di *Hyperledger* o *Zilliqa*.

L'algoritmo di consenso pBFT è progettato per funzionare al meglio in sistemi asincroni ed ha le seguenti caratteristiche:

- Tutti i nodi sono ordinati in sequenza. Uno di essi prende il nome di *leader* ed è considerato come il nodo principale, gli altri hanno invece funzione di backup e di validazione.
- Ogni nodo all'interno della rete comunica con gli altri nodi in maniera intensiva: lo scopo è quello di raggiungere un accordo riguardo lo stato della rete grazie all'opinione della maggioranza degli stessi.
- Il protocollo pBFT richiede che ogni nodo non *leader* verifichi sia che i messaggi provengano effettivamente dagli altri nodi all'interno della rete, ma anche la loro integrità ed affidabilità. [21]

Lo svantaggio principale è che il *Practical Byzantine Fault Tolerance*, abbracciando la dimostrazione di Lamport, parte dal presupposto che più del 66% dei nodi della rete siano onesti [22]. Se più del 33% dei nodi della rete dovessero invece non essere tali, il pBFT sarebbe vulnerabile a diversi tipi di attacchi. Un esempio è quello del *sybil attack*: si tratta della situazione in cui un singolo attore, a causa del fatto che non esiste modo di verificare a chi appartengono i differenti nodi di una rete sorretta dal pBFT, riesce ad ottenere il controllo della stessa grazie al possedimento di più del 33% dei nodi [21].

Un altro svantaggio del *Practical Byzantine Fault Tolerance* è rappresentato dal fatto che la scalabilità dei network che lo utilizzano non è molto elevata: il tempo necessario a raggiungere il consenso, rispetto a quanto avviene utilizzando protocolli più recenti, aumenta in maniera consistente all'aumentare dei nodi presenti.

4. I principali protocolli di consenso nelle differenti blockchain

Verranno di seguito descritti singolarmente i protocolli di consenso ad oggi più utilizzati nelle diverse implementazioni della blockchain. In particolare, nella sezione 4.1 verrà analizzato nel dettaglio il *Proof of Work* (*PoW*), che sorregge la rete di Bitcoin e di Litecoin, nella sezione 4.2 il *Proof of Stake* (*PoS*), che è stato invece adottato a partire da Settembre 2022 dalla Ethereum Foundation per l'omonima rete [26] (in precedenza anch'essa si basava sul *Proof of Work)*, ed infine nella sezione 4.3 il *Proof of Capacity* (*PoC*), un algoritmo di consenso di nuova generazione che viene invece utilizzato all'interno della blockchain di Chia Network.

4.1 Proof of Work (PoW)

In informatica, con *Proof of Work* ci si riferisce ad uno specifico tipo di prova crittografica composta da due parti: la prima è tenuta a dimostrare che un certo quantitativo di potere computazionale è stato speso nell'esecuzione di una operazione in un sistema digitale, l'altra che il potere sia stato utilizzato nel modo più efficiente possibile [27].

Il concetto alla base del protocollo venne teorizzato formalmente nel 1992 da parte di Cynthia Dwork e Moni Naor [28]. Inizialmente si trattava di una strategia volta a porre fine al problema delle email indesiderate (le cosiddette email spam, che ebbero la loro massima diffusione proprio all'inizio degli anni '90). Nel loro lavoro individuarono e presentarono diversi metodi per impedire agli spammer di inviare enormi quantità di email: l'idea era proprio quella di includere un costo computazionale per poter utilizzare i servizi di posta elettronica [29].

Tuttavia, il termine *Proof of Work* appare per la prima volta nel titolo di un articolo di Markus Jakobsson ed Ari Juels pubblicato nel 1999 [30], nel quale venne descritto l'utilizzo della strategia formalizzata da Dwork e Naor nel contesto dei protocolli crittografici.

Nelle blockchain che fanno uso di *Proof of Work*, i nodi che partecipano alla risoluzione dei problemi computazionali all'interno del meccanismo di consenso competono tra loro nel diventare *leader* di quel round, ossia nel riuscire ad ottenere la facoltà di aggiungere il blocco successivo alla catena.

Colui che riesce per primo a risolvere i puzzle crittografici ed a trovare quindi la soluzione viene ricompensato per il suo lavoro grazie al rilascio di un premio in criptovaluta. Questa procedura viene comunemente denominata *mining*.

I nodi non vincenti che hanno partecipato al consenso avranno invece a questo punto il compito di validare che la soluzione trovata sia corretta: quest'ultima è infatti computazionalmente difficile da trovare, ma facilmente verificabile.

Come già visto in precedenza, nel processo di *mining* e validazione dei nuovi blocchi ogni nodo è continuamente impegnato nel risolvere complesse funzioni di hash. Più nello specifico, la soluzione del problema del *Proof of Work* si ottiene trovando una x tale che $H(x) \leq T$ [1], dove con H si intende una funzione hash crittografica e con T una soglia (detta *target*) fissata dalla rete stessa [38]. L'operazione di generazione di un nuovo blocco sarà tanto più difficile e costosa quanto più piccola è tale soglia.

All'interno del *genesis block* è stato preventivamente fissato un valore T come *target*, tuttavia ogni 2016 blocchi validati quest'ultimo può venire modificato nei seguenti modi [38]:

- Nel caso in cui gli ultimi 2016 blocchi siano in media stati validati in meno di 600 secondi, viene diminuito il valore di *T*.
- Nel caso in cui gli ultimi 2016 blocchi siano in media stati validati in più di 600 secondi, viene invece aumentato il valore di *T*.

In una blockchain di *n* nodi la probabilità *P* di un singolo nodo *a* di diventare *leader* e quindi di riuscire a minare un nuovo blocco è $Pa = \frac{c_a}{\sum_{a=1}^{n} c_a}$ [34].

Nello specifico, la procedura che i minatori seguono è la seguente:

1. Viene aumentato (incrementandolo ogni volta di 1) un numero casuale presente nell'*header* (o intestazone) del blocco che vogliono aggiungere, detto anche *nonce.*
2. Si esegue un doppio hash SHA256 dell'intero *header* del blocco che si vuole aggiungere, ossia $SHA^2(block\ header)$. Quest'ultimo, come già visto, comprende oltre che il *nonce* anche altri valori tra cui un *timestamp* e l'hash del blocco precedente.
3. Si verifica che $H \leq T$, cioè che l'hash calcolato sia minore di *T*:
   - nel caso ciò sia vero, il nodo ha risolto il problema (e di conseguenza anche minato il blocco, che verrà aggiunto alla blockchain e validato da tutti gli altri nodi).
   - in caso contrario, si riparte dal punto 1.

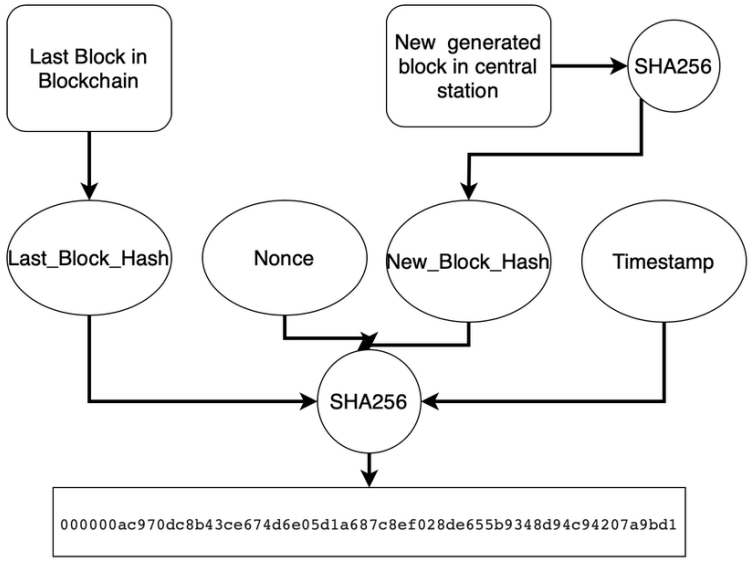


Figura 4.1: funzionamento del Proof of Work

Nella figura 4.1 viene mostrato in maniera grafica il meccanismo appena descritto. Di seguito verranno riassunte brevemente le più importanti caratteristiche delle blockchain che fanno uso del protocollo *Proof of Work* come algoritmo di consenso.

4.1.1 Bitcoin

Nel 2008 Satoshi Nakamoto, pseduonimo di una identità rimasta anonima, individuò nell'utilizzo del *Proof of Work* l'elemento chiave per risolvere il problema della doppia spesa [1], utilizzando quindi delle basi crittografiche già conosciute (funzioni di hash, *Proof of Work*, *Byzantine Fault Tolerance*) per dare vita alla prima blockchain. Nel riepilogo del white paper di Bitcoin si legge infatti quanto segue [1]:

"We propose a solution to the double-spending problem using a peer-to-peer network.

The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work."

Il client ufficiale rilasciato da Satoshi Nakamoto, conosciuto inizialmente come *Bitcoin-Qt (*in seguito come *Bitcoin Core)* è un software libero che deriva direttamente dal codice scritto da egli stesso per implementare la rete *peer-to-peer* che ne risulta, ossia la blockchain [40].

Bitcoin è storicamente il primo esempio di una implementazione completa del concetto di "criptovaluta", descritto per la prima volta nel 1996 da Wei Dai all'interno di una mailing list [39].

4.1.2 Litecoin

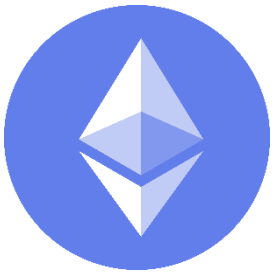
Litecoin è stato il primo *hard fork* di Bitcoin, avvenuto nel 2011 [31]. Entrambe le blockchain utilizzano lo stesso algoritmo *Proof of Work* per raggiungere il consenso. Le differenze principali fra Bitcoin e Litecoin si esauriscono in due caratteristiche:

- La supply totale, cioè il numero massimo di singole unità BTC/LTC esistenti in circolazione (4 volte maggiore per Litecoin).
- La velocità di generazione ed approvazione dei nuovi blocchi (4 volte maggiore per Litecoin).

In un primo momento diverse persone ritenevano che Litecoin, proprio grazie alla maggior velocità di generazione dei nuovi blocchi ed alle commissioni di transazione mediamente più basse rispetto a Bitcoin [35], avrebbe preso il posto di quest'ultima come criptovaluta con la capitalizzazione di mercato più alta [36].

Tuttavia, il sorpasso non è mai avvenuto e nel corso degli anni l'interesse per Litecoin è progressivamente sceso (in particolare perché la sua blockchain è stata spesso utilizzata solo come piattaforma di beta-testing per modifiche che dovevano in seguito essere implementate nella blockchain di Bitcoin) [37].

4.1.3 Ethereum 1.0



Nel 2015 venne rilasciata da Vitalik Buterin la prima versione di Ethereum, una criptovaluta che si prefissava di integrare alla sua blockchain aspetti innovativi quali gli *smart contracts* e la *decentralized finance* (DeFI) [32].

Inizialmente la blockchain utilizzava come meccanismo di consenso una variante del *Proof of Work*, conosciuta nello specifico come *Ethash*.

Le transazioni di Ethereum vengono elaborate in blocchi, come nel caso di Bitcoin. In Ethereum 1.0, ogni blocco conteneva [33]:

- La difficoltà del blocco, ad esempio: 3.324.092.183.262.715.
- Un mixHash, ad esempio: 0x44bca881b07a6a09f83b130798072441705d9a665c5ac8bdf2f39a3cdf3bee29
- Un nonce, ad esempio: 0xd3ee432b4fb3d26b

Questi dati del blocco erano direttamente correlati al protocollo Ethash, che non viene al giorno d'oggi più utilizzato nella blockchain di Ethereum. Infatti, nel 2022 la Ethereum Foundation ha disattivato il *Proof of Work* ed ha iniziato, invece, ad utilizzare il *Proof of Stake* come algoritmo di consenso [26].

4.2 Proof of Stake (PoS) e derivati

Il *Proof of Stake* (*PoS*) è un algoritmo di consenso alternativo al *Proof of Work*, proposto per la prima volta a Luglio del 2011 da QuantumMechanic, un utente sul forum di "Bitcointalk" [41], come soluzione ai problemi intrinsechi derivanti dalla grande quantità di elettricità ed energia necessaria per l'utilizzo del *Proof of Work*.

Nel corso degli anni sono nati diversi protocolli di consenso direttamente derivanti dal *Proof of Stake* originale, come per esempio il *Delegated Proof of Stake* (*DPoS*) ed il *Pure Proof of Stake* (*PPoS*), che verranno descritti in seguito.

4.2.1 Proof of Stake (PoS) originale

La prima formalizzazione teorica del protocollo *Proof of Stake* è avvenuta nel 2012 da parte dei fondatori della blockchain di *Peercoin* (conosciuta anche come *PPCoin*) [42]. Sunny King e Scott Nadal pubblicarono infatti un *whitepaper* relativo alla criptovaluta in questione che abbracciava proprio il protocollo di consenso proposto nell'anno precedente dall'utente del forum di "Bitcointalk": il *Proof of Stake*.

Analogamente a come succede con il *Proof of Work*, nelle blockchain che fanno uso dell'algoritmo *Proof of Stake* il consenso viene raggiunto selezionando in maniera pseudo-randomica il nodo *leader* che avrà il compito di aggiungere e validare il blocco successivo nella catena. La differenza principale fra i due sistemi è che nel *Proof of Work* la scelta viene fatta sulla base della potenza computazionale dei nodi partecipanti al round di consenso, nel *Proof of Stake* invece viene valutato lo *stake* di questi ultimi, ossia la quantità di criptovalute impegnate come garanzia all'interno della blockchain [43].

Lo *stake* funziona come una garanzia posta dal nodo per il corretto svolgimento del proprio lavoro di validatore. Qualora un nodo dovesse compiere errori o attività fraudolente, perderebbe tutti i suoi token messi in *staking* [44].

La probabilità di venire scelti è proporzionale alla quantità di monete che i nodi stessi possiedono: più monete hanno, più alta sarà la probabilità di riuscire a minare il blocco seguente [44]. Di conseguenza, in una blockchain utilizzante il *Proof of Stake* di *n* nodi la probabilità *P* di un singolo nodo *a* di diventare *leader* e quindi di riuscire a minare un nuovo blocco è $Pa = \frac{S_a}{\sum_{a=1}^{n} S_a}$ [34], dove S_a è lo *stake* del nodo *a*.

Per garantire la casualità del sistema ed evitare che i nodi più ricchi vengano continuamente selezionati come *leader* dal meccanismo di consenso, oltre allo *stake*, vengono spesso utilizzati anche altri criteri (spesso diversi fra le differenti implementazioni dell'algoritmo *Proof of Stake*) [45].

In particolare il più famoso è conosciuto come *Coin Age Based Selection*: l'importanza dello *stake* (ai fini della formula sopramenzionata) di ciascun nodo aumenta progressivamente in base a quanto tempo è passato dall'ultima modifica effettuata allo stesso (di solito i primi vantaggi cominciano ad essere assegnati quando le criptovalute rimangono in *stake* per più di 30 giorni, aumentando esponenzialmente col passare del tempo). Il valore così ottenuto viene definito *coin age*, ed è direttamente collegato alla probabilità del nodo di venire scelto come leader (quando ciò accade, tuttavia, il *coin age* ritorna a 0) [45].

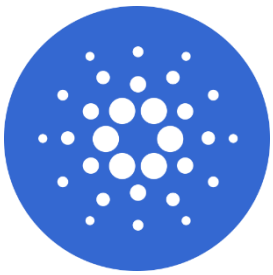
4.2.1.1 Ethereum 2.0

Come già visto nei paragrafi precedenti, durante l'estate del 2022 Ethereum è stato aggiornato alla versione 2.0 attraverso il *Merge*, procedura che ha apportato una serie di novità e miglioramenti sostanziali alla blockchain ed al suo funzionamento [26].

La novità più importante è ovviamente stata il passaggio al *Proof of Stake*, tuttavia anche altri aspetti del progetto sono stati significativamente modificati. Ad esempio, la *tokenomics*, ossia il modello che descrive le caratteristiche economiche di un token e contiene tutte le informazioni relative all'emissione e alla distribuzione di nuovi token all'interno di una blockchain (in questo caso, di nuovi ETH all'interno della blockchain di Ethereum). Il passaggio al nuovo meccanismo di consenso ed allo *staking* ha determinato infatti una emissione di token ETH minore all'interno della rete, cosa che è andata ad influire anche sull'inflazione della criptovaluta che, nel periodo precedente al *Merge*, era del 4% circa. Viceversa, in alcuni momenti immediatamente successivi all'aggiornamento 2.0, Ethereum ha evidenziato una tendenza deflazionaria (sono stati distrutti più Ether di quanti ne venissero emessi) [54].

In generale, grazie al passaggio alla versione 2.0, si prevede che la rete di Ethereum consumerà il 99,5% di energia in meno rispetto a quando utilizzava il *Proof of Work* [53].

4.2.1.2 Cardano



Cardano è la piattaforma all'interno della quale è nato il token ADA, criptovaluta che come molte altre può essere innanzitutto utilizzata per inviare e ricevere fondi. Più nello specifico, si tratta di una piattaforma che avrà nel futuro il suo focus sugli *smart contracts*. Le modalità di implementazione ed il loro funzionamento saranno simili a quelli di Ethereum, offrendo tuttavia nuovi livelli di sicurezza e scalabilità grazie a una architettura *multilayer* (sono previste infatti diverse *sidechain* complementari alla blockchain principale).

Gli sviluppatori dietro al progetto di Cardano hanno creato un'approfondita *roadmap* futura dello sviluppo, composta da cinque fasi principali [50] [51]:

- **Fase 1 (*Byron*)**: La rete viene lanciata con le funzionalità di base (trasferimento di token ADA).
- **Fase 2 (*Shelley*)**: Vengono intrapresi passaggi verso la decentralizzazione con nodi gestiti dalla comunità.
- **Fase 3 (*Goguen*)**: gli *smart contracts* sono abilitati sulla rete.
- **Fase 4 (*Basho*)**: Vengono introdotte delle *sidechain* di appoggio alla blockchain principale, migliorando la scalabilità e l'interoperabilità.
- **Fase 5 (*Voltaire*)**: la *governance* (o gestione) decentralizzata rende ADA completamente indipendente dal mondo esterno.

Al momento ci troviamo durante lo sviluppo della fase *Goguen* ed i contratti intelligenti sono infatti in fase di testing [50].

4.2.2 Delegated Proof of Stake (DPoS)

Il meccanismo di consenso *Delegated Proof of Stake (DPoS)* è una espansione del meccanismo *Proof of Stake*, è stato creato nel 2014 da una proposta di Daniel Larimer [54], sviluppatore e creatore di diverse blockchain (tra cui Steem, BitShares ed EOS). La prima implementazione del *Delegated Proof of Stake* è avvenuta l'anno successivo ed al giorno d'oggi viene considerata una delle varianti del *Proof of Stake* più sicure e semplici da utilizzare.

Gli utenti di una blockchain che utilizzata questo meccanismo di consenso sono tenuti a selezionare un numero sufficiente di nodi delegati, chiamati anche "testimoni", che avranno il compito di assicurare la decentralizzazione della rete. I delegati eletti verificheranno le transazioni e genereranno i blocchi. Normalmente, se un delegato riceve una ricompensa grazie alla creazione di un nuovo blocco, la condivide con quei portafogli che hanno votato per lui durante la scelta iniziale e che solitamente hanno anche una partecipazione sostanziale nella rete (cioè diversi token in *staking*). Nella figura 4.2.2 vengono riassunte le modalità di funzionamento del *Delegated Proof of Stake.*

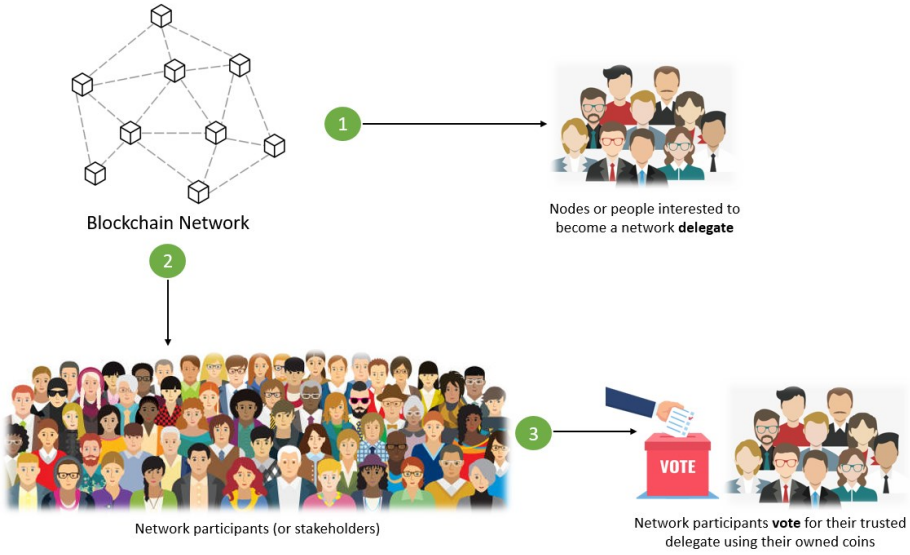


Figura 4.2.2: funzionamento del Delegated Proof of Stake

Se un delegato non riesce a convalidare una transazione o subisce una interruzione della sua connessione di rete, questo meccanismo di consenso consente agli altri delegati di rilevare rapidamente la discrepanza e di sostituire rapidamente i "testimoni" creatori di blocchi che non soddisfano i requisiti richiesti al raggiungimento del consenso.

Uno degli aspetti più innovativi del *Delegated Proof of Stake* è la possibilità di modificare i parametri di rete in qualsiasi momento senza passare per un *fork*, utilizzando un processo di voto da parte dei "testimoni": grazie a questa possibilità del meccanismo di consenso è possibile rendere la blockchain molto flessibile.

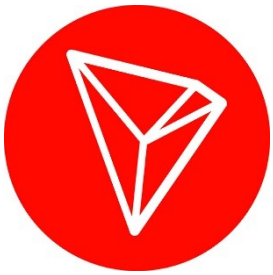
4.2.2.1 EOS

Un primo esempio di criptovaluta utilizzante il *Delegated Proof of Stake* è EOS. Nata da una idea di Dan Lariner, il *whitepaper* è stato pubblicato nel 2017 [59]. EOS consente agli utenti della rete di sviluppare, ospitare ed eseguire app decentralizzate, oltre ad essere molto versatile nell'hosting di *smart contracts*. Pur essendo ancora in fase di sviluppo, EOS è considerata una delle infrastrutture più potenti e meglio gestita fra quelle operanti nell'ambito delle *Decentralized Applications (dApps)* [60].

Grazie all'utilizzo dell'algoritmo di consenso *Delegated Proof of Stake*, la scalabilità della blockchain è molto alta e le operazioni sui blocchi di EOS (come il lancio di *dApps*, ma anche semplici transazioni monetarie) avvengono in maniera molto più rapida, ad esempio, rispetto ad Ethereum. Tuttavia, essendo un progetto molto centralizzato sulle idee della fondazione

EOS.IO, la decentralizzazione effettivamente raggiunta dallo stesso non è quella ideale [61].

4.2.2.2 TRON



Un altro degli esempi più famosi di piattaforme che utilizzano il *Delegated Proof of Stake*, TRON è una progetto blockchain che nasce da una idea di Justin Sun nel 2014, il quale ha poi anche fondato nel 2018 la TRON Foundation [57]. Si tratta (similarmente ad EOS) di una piattaforma "all in one" che oltre allo scambio di criptovalute, conosciute come TRX, dà agli utenti l'accesso ad applicazioni di intrattenimento (gioco), video e materiali grafici; così come la possibilità di comunicare tra loro [57] [58].

La piattaforma di TRON è infatti suddivisa in 3 *layer* principali [58]:

- ***Storage Layer****:* consiste in un'archiviazione distribuita a blocchi e a stati, con una scheda dati grafica per permettere un'elaborazione veloce.
- ***Application Layer:*** è adatto agli sviluppatori per creare e distribuire *dApps*, personalizzarle ed emettere i propri token.
- ***Central Layer****:* contiene vari moduli che si occupano delle funzioni blockchain più classiche di TRON, come gli *smart contracts*, la gestione dell'algoritmo di consenso e delle transazioni di TRX.

Tutti i livelli sono collegati con un protocollo compatibile con diversi linguaggi di programmazione, tra cui Java, Python, C++, Scala e Go [58].

4.2.3 Pure Proof of Stake (PPoS)

Viste le criticità dei meccanismi di consenso già esistenti, Silvio Micali, un matematico e informatico italiano, ha proposto nel 2017 l'utilizzo di un nuovo algoritmo, conosciuto con il nome di *Pure Proof of Stake (PPoS)*. Ricordiamo che ci troviamo nell'ambito dei sistemi distribuiti, in cui quindi vi sono più nodi distribuiti e dislocati in comunicazione tra di loro [62], di conseguenza l'obiettivo principale è sempre quello di ottenere un consenso distribuito riguardo le decisioni da prendere sul futuro della catena di blocchi. Il *Pure Proof of Stake* è considerato un meccanismo di consenso "bizantino", in quanto prevede l'elezione di un *leader* e si pone come presupposto che almeno 2/3 dei nodi che partecipano saranno onesti.

L'algoritmo di consenso *Pure Proof of Stake* procede per round ed ognuno di questi porterà alla creazione di un nuovo blocco da appendere alla blockchain. Questo avviene tramite l'estrazione di un numero *n* di utenti che avranno diritto a formare un "comitato" che deciderà su quali blocchi rigettare e quali invece approvare. La scelta riguardo che utenti parteciperanno al consenso avviene tramite *Verifiable Random Functions*, cioè delle funzioni che prendono in input un valore ed una chiave e producono in output un numero pseudo-randomico (assieme ad una *proof* riguardo le modalità di creazione dell'output, utilizzabile da chiunque per verificare la correttezza della funzione stessa) [62]. Il funzionamento è chiaramente riconducibile a quello di una lotteria, infatti ogni utente che esegue la VRF può essere considerato come un biglietto per parteciparne.

Il vantaggio principale rispetto al *Proof of Stake* originale è di facile comprensione: con il *Pure Proof of Stake* non c'è la necessità di mettere in *stake* parte del proprio patrimonio [62].

4.2.3.1 Algorand

Nel 2017, come conseguenza alla proposta dell'utilizzo del *Pure Proof of Stake*, Silvio Micali fonda Algorand, una piattaforma digitale di pagamenti e valuta digitale [63]. Pur non ricorrendo ad una *Initial Coin Offering* (cioè donazioni da parte di utenti *retail*), nei primi 4 mesi dopo il lancio il progetto è riuscito a raccogliere oltre 4 milioni di euro basandosi solamente su donazioni di investitori accreditati [65].

Proprio attraverso l'utilizzo del *Pure Proof of Stake*, Algorand ha la caratteristica unica di assicurare la possibilità di far parte della *governance* della blockchain a tutti i partecipanti alla rete, indifferentemente dalla quantità di ALGO posseduta (la criptovaluta nativa del progetto) [64].

Algorand ha una fornitura limitata di 10 miliardi di ALGO, che sono stati coniati al momento del lancio della blockchain, tuttavia al momento ne sono presenti in circolazione solo meno di un terzo [64].

4.3 Proof of Capacity (PoC)

Un più recente algoritmo di consenso è il *Proof of Capacity (PoC)*, si tratta di un meccanismo utilizzato in alcune blockchain che permette ai nodi all'interno del network di utilizzare lo spazio libero su piattaforme di archiviazione di massa (ad esempio, gli hard disk) come mezzo per validare le transazioni e decidere con che priorità inserire nella catena i nuovi blocchi che vengono proposti dai diversi nodi partecipanti. Il raggiungimento del consenso distribuito attraverso questo procedimento è una valida alternativa al raggiungimento dello stesso tramite l'utilizzo della potenza computazionale (come avviene nel *Proof of Work*) oppure lo sfruttamento

della quantità di criptovalute messe in *stake* (come avviene nel *Proof of Stake*).

L'algoritmo *Proof of Capacity* infatti è nato inizialmente allo scopo di tentare di contenere l'alto spreco di risorse energetiche che avveniva nelle blockchain facenti uso del *Proof of Work*, ma allo stesso tempo anche con il fine di evitare di dare troppa importanza agli individui più abbienti, e che quindi potrebbero abusare dell'algoritmo *Proof of Stake* acquistando una enorme quantità di criptovalute all'unico scopo di metterle in *staking* [46].

4.3.1 Chia Network

Chia Network (il cui token nativo è XCH) è una criptovaluta nata nel 2017 dall'idea di Bram Cohen, già autore di *BitTorrent*, il noto protocollo *peer to peer* di condivisione file. La blockchain di Chia si basa sul soprammenzionato algoritmo di consenso, il *Proof of Space*. Il focus del progetto è infatti proprio quello di arginare i danni ambientali causati dal *mining* delle criptovalute già esistenti (Bitcoin, Ethereum...) [47].

La soluzione proposta è quella di sostituire il *mining* con il *farming* attraverso il *Proof of Capacity*: esso può essere visto come ad un modo per dimostrare che si sta riservando in maniera esclusiva parte dello spazio libero sul proprio disco rigido (HDD) o disco solido (SDD) al processo di formazione delle nuove criptovalute [48]. I differenti nodi della blockchain "seminano" lo spazio nei loro dischi rigidi installando un client apposito fornito da Chia Network (che in questo caso funziona in maniera analoga al Bitcoin Core). Tale client viene utilizzato per generare una serie di numeri crittografici

(*hash functions*), che nella blockchain di Chia vengono denominati *plots*. Ne deriva che più spazio sul disco si dedica al client di Chia, più *plot* si riusciranno a "farmare". I singoli *plot* di ogni nodo verranno infine confrontati con una funzione di hash generata dalla blockchain, il primo che riuscirà a farle combaciare avrà diritto a creare il nuovo blocco da appendere alla catena e riceverà 64 nuovi XCH come compenso [49].

5. Confronto e “Trilemma della Blockchain”

Nei capitoli precedenti sono stati descritti nel dettaglio i principali algoritmi di consenso presenti nel panorama delle tecnologie blockchain, analizzandone in maniera estesa i principali pregi e difetti. In queste pagine finali verrà introdotto il “Trilemma della Blockchain”, grazie al quale sarà possibile confrontare le 3 più importanti caratteristiche del *Proof of Work*, del *Proof of Stake* e del *Proof of Capacity*.

Il “Trilemma della Blockchain” (termine coniato da Vitalik Buterin, co-fondatore di Ethereum) afferma che risulta impossibile che in un meccanismo di consenso distribuito possono essere garantite contemporaneamente tutte le 3 seguenti proprietà, ma solamente 2 al massimo:

- **Decentralizzazione**: invece di essere gestito da pochi nodi, le blockchain dovrebbero distribuire il controllo sulla rete in modo equo fra tutti i partecipanti.
- **Sicurezza**: le blockchain, idealmente, dovrebbero essere in grado di impedire con certezza gli attacchi da parte di potenziali terzi che vogliono prenderne il controllo.
- **Scalabilità**: maggiore è il numero di transazioni che le blockchain possono processare in un determinato tempo, maggiore risulta essere la scalabilità [66].

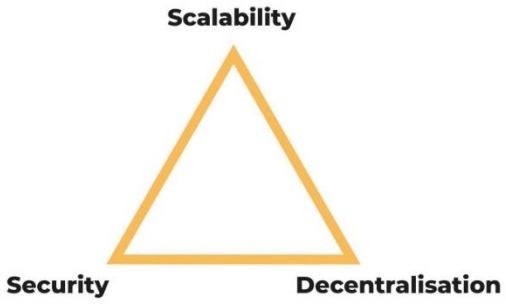


Figura 5: il "Trilemma della Blockchain" disegnato graficamente sotto forma di triangolo

Per risolvere il problema, una soluzione semplice potrebbe essere quella di ridurre il numero di partecipanti che confermano e aggiungono dati alla rete, in modo da migliorarne la scalabilità e la velocità. Tuttavia, questa scelta potrebbe comportare un indebolimento della decentralizzazione, con il controllo affidato a pochi partecipanti, e una diminuzione della sicurezza, a causa della maggiore probabilità di attacchi.

È questa caratteristica che genera il trilemma: la diretta connessione tra le proprietà desiderate di decentralizzazione, sicurezza e scalabilità della blockchain rende la gestione dell'algoritmo che si occupa del consenso della stessa molto complessa, in quanto l'aumento di una proprietà comporta inevitabilmente un indebolimento delle altre [67].

Nella seguente tabella è possibile osservare il livello di scalabilità, decentralizzazione, sicurezza, tolleranza ai guasti e consumo energetico effettivamente raggiunto dai protocolli di consenso analizzati:

	Proof of Work	***Proof of Stake***	***Proof of Capacity***
Decentralizzazione	Alta	Bassa	Media
Sicurezza	Alta	Media	Bassa
Scalabilità	Bassa	Alta	Alta
Consumi energetici	Alti	Medi	Bassi
Tolleranza ai guasti	Alta	Media	Bassa

6. Considerazioni finali

Proprio grazie alle sue modalità di funzionamento (*mining*, costo computazionale elevato ecc…), è facile intuire come l'algoritmo *Proof of Work* possieda delle caratteristiche intrinseche in grado di garantire un maggior grado di decentralizzazione (e sicurezza, a discapito della scalabilità), mentre il *Proof of Stake,* da parte sua, offre una maggiore scalabilità (e quindi minori costi energetici, a discapito della decentralizzazione).

Da questo si può dedurre che il *Proof of Work* risulta l'algoritmo più adatto per tutte quelle criptovalute la cui principale funzione è l'essere una "riserva di valore", come ad esempio Bitcoin, mentre il *Proof of Stake* è da preferirsi in tutte quelle blockchain che ospitano degli *smart contracts* ed applicazioni "vive" di finanza decentralizzata [68].

Il *Proof of Capacity*, infine, pur essendo in grado di garantire sia un buon grado di decentralizzazione che di scalabilità, è molto carente dal punto di vista della sicurezza: risulta infatti molto semplice ottenere una grande percentuale dell'*hash rate* totale, a causa dell'utilizzo al momento limitato delle poche blockchain che lo adottano.

In conclusione, non esiste un algoritmo di consenso superiore in tutti gli aspetti ad un altro, in quanto ognuno di essi presenta specifici vantaggi e svantaggi.

Bibliografia

[1] Satoshi Nakamoto “Bitcoin: A peer-to-peer electronic cash system” (2008)

http://bitcoin.org/bitcoin.pdf

[2] S. Haber, W. S. Stornetta “How to Time-Stamp a Digital Document” (1991)

http://link.springer.com/content/pdf/10.1007/BF00196791.pdf

[3] Nick Szabo “Bit Gold: Towards Trust-Independent Digital Money” (consultato il 16/09/2022)

https://web.archive.org/web/20140406003811/http://szabo.best.vwh.net/bitgold.html

[4] Stefan Konst “Secure Log Files Based on Cryptographically Concatenated Entries” (2000)

http://konst.de/stefan/seclog.pdf

[5] Amazon AWS “What is blockchain” (consultato il 18/09/2022)

https://aws.amazon.com/it/what-is/blockchain/

[6] Bitcoin Wiki “Double Spending” (consultato il 18/09/2022)

https://en.bitcoin.it/wiki/Double-spending

[7] Marco Iansiti, Karim R. Lakhani “The Truth About Blockchain” (2017)

https://hbr.org/2017/01/the-truth-about-blockchain

[8] Coinbase Wiki “What is cryptography” (consultato il 20/09/2022)

https://www.coinbase.com/it/learn/crypto-basics/what-is-cryptography

[9] William Stallings “Cryptography and Network Security: Principles and Practice”, Capitolo 9 “Public-key cryptography and RSA” (edizione 2017, 978-1-292-15858-7)

[10] John P. Conley “Encryption, Hashing, PPK, and Blockchain: A Simple Introduction” (2019)

http://www.accessecon.com/pubs/VUECON/VUECON-19-00013.pdf

[11] Andreas M. Antonopoulos "Mastering Bitcoin", Capitolo 7 "The Blockchain" (edizione 2014, 978-1-449-37404-4)

[12] Benjamin Sirb, Xiaojing Ye “Consensus Optimization with Delayed and Stochastic Gradients on Decentralized Networks” (2016)

https://math.gsu.edu/bsirb1/ddgd_bigdata.pdf

[13] Natalia Chaudhry, Muhammad Murtaza Yousaf “Consensus Algorithms in Blockchain: Comparative Analysis, Challenges, and Opportunities” (2018)

https://www.researchgate.net/publication/330880555_Consensus_Algorithms_in_Blockchain_Comparative_Analysis_Challenges_and_Opportunities

[14] Paul Krzyzanowski "Consensus - Reaching agreement" (consultato il 10/10/2022)

https://people.cs.rutgers.edu/~pxk/417/notes/content/consensus.html

[15] Leslie Lamport, Robert Shostak, Marshall Pease “The Byzantine Generals Problem” (1982)

https://lamport.azurewebsites.net/pubs/byz.pdf

[16] Stefano Lovati "Algoritmi di consenso: Proof of Work, Proof of Stake o dBFT?" (consultato il 29/10/2022)

https://it.emcelettronica.com/algoritmi-di-consenso-proof-of-work-proof-of-stake-o-dbft

[17] Matteo Gatti "The Byzantine Generals problem and Bitcoin’s solution" (consultato il 26/10/2022)

https://en.cryptonomist.ch/2019/08/04/byzantine-generals-bitcoin-solution/

[18] Steffy Díaz “¿Qué es la Tolerancia a Fallas Bizantinas? Guía rápida” (consultato il 04/11/2022)

https://blog.bitnovo.com/que-es-la-tolerancia-a-fallas-bizantinas-guia-rapida/

[19] Geeks for Geeks "Consensus Problem of Distributed Systems" (consultato il 02/11/2022)

https://www.geeksforgeeks.org/consensus-problem-of-distributed-systems/

[20] Miguel Castro, Barbara Liskov "Practical Byzantine Fault Tolerance" (1999)

https://pmg.csail.mit.edu/papers/osdi99.pdf

[21] Brian Curran "What is Practical Byzantine Fault Tolerance?" (consultato il 05/11/2022)

https://blockonomi.com/practical-byzantine-fault-tolerance/

[22] Bitcoin Wiki “Practical Byzantine Fault Tolerance” (consultato il 06/11/2022)

https://en.bitcoinwiki.org/wiki/PBFT

[23] Sam Daley “Blockchain in Healthcare: 17 Examples to Know” (consultato il 07/11/2022)

https://builtin.com/blockchain/blockchain-healthcare-applications-companies

[24] IBM “What is Blockchain” (consultato il 07/11/2022)

https://www.ibm.com/topics/what-is-blockchain

[25] Stephan Leible, Steffen Schlager, Moritz Schubotz, Bela Gipp “A Review on Blockchain Technology and Blockchain Projects Fostering Open Science” (2019)

https://www.frontiersin.org/articles/10.3389/fbloc.2019.00016/full

[26] Ethereum Blog "The Merge" (consultato il 26/11/2022)

https://ethereum.org/en/upgrades/merge/

[27] Nada Lachtar, Abdulrahman Abu Elkhail, Anys Bacha, Hafiz Malik "A Cross-Stack Approach Towards Defending Against Cryptojacking" (2020)

https://ieeexplore.ieee.org/document/9170774

[28] Cynthia Dwork, Moni Naor "Pricing via Processing or Combatting Junk Mail" (1992)

https://link.springer.com/content/pdf/10.1007/3-540-48071-4_10.pdf

[29] BTC Sentinel, "Proof of Work come nasce e come funziona con le criptovalute" (consultato il 01/01/2022)

https://www.btcsentinel.com/blog/come-nasce-e-cosa-e-la-proof-of-work-prova-di-lavoro-guida

[30] Markus Jakobsson, Ari Juels "Proofs of work and bread pudding protocols" (1999)

https://link.springer.com/content/pdf/10.1007/978-0-387-35568-9_18.pdf

[31] Jeff Reed "Litecoin: An Introduction to Litecoin Cryptocurrency and Litecoin Mining" (edizione 2017, 978-1974692941)

[32] Vitalik Buterin "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform" (2014)

https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf

[33] Ethereum Blog "Proof of Work (PoW)" (consultato il 01/12/2022)

https://ethereum.org/it/developers/docs/consensus-mechanisms/pow/

[34] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, E. Dutkiewicz "Proof of stake consensus mechanisms for future blockchain networks: Fundamentals applications and opportunities" (2019)

https://ieeexplore.ieee.org/document/8746079

[35] Sean Williams “It's just a matter of time before Litecoin overtakes Bitcoin” (consultato il 02/12/2022)

https://www.fool.com/investing/2018/03/06/its-just-a-matter-of-time-before-litecoin-overtake.aspx

[36] Lisa Froelings “Could Litecoin become a better investment than Bitcoin soon?” (consultato il 02/12/2022)

https://cointelegraph.com/news/could-litecoin-become-better-investment-than-bitcoin-soon

[37] Alyssa Hertig “Litecoin is Giving New Life to Bitcoin's Most Experimental Tech” (consultato il 03/12/2022)

https://www.coindesk.com/markets/2017/05/08/litecoin-is-giving-new-life-to-bitcoins-most-experimental-tech/

[38] Greg Walker “What is the Target in Bitcoin” (consultato il 03/12/2022)

https://learnmeabitcoin.com/technical/target

[39] Wei Dai "B-money" (consultato il 04/12/2022)

https://web.archive.org/web/20120825232101/http://www.weidai.com/bmoney.txt

[40] Bitcoin Wiki "Bitcoin Core" (consultato il 04/12/2022)

https://en.bitcoin.it/wiki/Bitcoin_Core

[41] Post sul forum “Bitcointalk” (consultato il 27/12/2022)

https://bitcointalk.org/index.php?topic=27787.0

[42] Sunny King, Scott Nadal “PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake” (2012)

https://decred.org/research/king2012.pdf

[43] Giulia Spinoglio “Proof of Stake, cos’è e perché sta soppiantando il Proof of Work nella tecnologia blockchain” (consultato il 28/12/2022)

https://www.blockchain4innovation.it/esperti/proof-of-stake-cose-perche-sta-soppiantando-il-proof-of-work

[44] Young Platform “Proof-of-Stake” (consultato il 28/12/2022)

https://youngplatform.com/glossary/proofofstake

[45] Techskill Brew “Proof of Stake in Blockchain” (consultato il 28/12/2022)

https://medium.com/techskill-brew/proof-of-stake-or-pos-in-blockchain-part-8-blockchain-basics-32d461232e1c

[46] Adam Hayes “Proof of Capacity (Cryptocurrencies)” (consultato il 15/01/2023)

https://www.investopedia.com/terms/p/proof-capacity-cryptocurrency.asp

[47] Giuseppe Travia “Comprare Chia Coin: Scopriamo l’alternativa a Bitcoin per il futuro green delle criptovalute” (consultato il 19/01/2023)

https://www.finaria.it/criptovalute/comprare-chiacoin

[48] Wikipedia “Chia (criptovaluta)” (consultato il 21/01/2023)

https://it.wikipedia.org/wiki/Chia_(criptovaluta)

[49] Elena Perez “How to farm Chia: A guide to XCH token farming using a hard drive” (consultato il 21/01/2023)

https://cointelegraph.com/news/how-to-farm-chia-a-guide-to-xch-token-farming-using-a-hard-drive

[50] Kriptomat “Cosa sono i Cardano” (consultato il 12/02/2023)

https://kriptomat.io/it/criptovalute/cardano/cosa-sono-i-cardano

[51] Cardano Wiki “Roadmap” (consultato il 12/02/2023)

https://roadmap.cardano.org

[52] Maria Teresa Della Mura “Cardano (ADA): cos’è, come funziona, prezzi e grafici, applicazioni oggi” (consultato il 10/02/2023)

https://www.blockchain4innovation.it/criptovalute/andamento/cose-e-come-funziona-cardano-ada/

[53] Mark Hooson “Ethereum 2.0: il Merge di Ethereum è avvenuto” (consultato il 08/02/2023)

https://www.forbes.com/advisor/it/investire/criptovalute/ethereum-2-0-il-merge-di-ethereum-e-avvenuto/

[54] Daniel Larimer “Delegated Proof-of-Stake (DPOS)” (2014)

https://web.archive.org/web/20140408144653/http://107.170.30.182/security/delegated-proof-of-stake.php

[55] Wikipedia “Delegated Proof of Stake” (consultato il 09/02/2023)

https://it.wikipedia.org/wiki/Delegated_Proof_of_Stake

[56] Bitpanda Academy “Algoritmi di consenso: Proof of Stake” (consultato il 09/02/2023)

https://www.bitpanda.com/academy/it/lezioni/algoritmi-di-consenso-proof-of-stake/

[57] Tron Whitepaper (2018)

https://assets.ctfassets.net/sdlntm3tthp6/oX7vrBFcwCWyU0kY2Egsc/dead55516662b0e275cfb32ccc9fe594/TronWhitepaper_en.pdf

[58] Francesca Rizzi “Cos’è Tron (TRX) e come funziona?” (consultato il 14/02/2023)

https://rankia.it/criptovalute/cose-tron-trx-e-come-funziona

[59] EOS Whitepaper (2017)

https://www.allcryptowhitepapers.com/eos-whitepaper

[60] Anycoin Direct “Cos'è il EOS?” (consultato il 14/02/2023)

https://anycoindirect.eu/it/criptovalute/cose-il-eos

[61] Bit2Me Academy “Cos'è EOS (EOS)?” (consultato il 14/02/2023)

https://academy.bit2me.com/it/cos%27%C3%A8-la-criptovaluta-eos/

[62] Tomàs Daniel Avila Visintin “Pure Proof of Stake (PPoS) - Spiegazione del consenso in Algorand” (consultato il 14/02/2023)

https://cryptowebacademy.com/it/pure-proof-of-stake-ppos-spiegazione-del-consenso-in-algorand

[63] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, Nickolai Zeldovich “Algorand: Scaling Byzantine Agreements for Cryptocurrencies” (2017)

https://people.csail.mit.edu/nickolai/papers/gilad-algorand-eprint.pdf

[64] Kriptomat “Che cos’è la criptovaluta Algorand (ALGO) e come funziona?” (consultato il 15/02/2023)

https://kriptomat.io/it/criptovalute/algorand/cosa-sono-i-algorand

[65] Stan Higgins “MIT Professor Raises $4 Million to Build a Better Blockchain” (consultato il 15/02/2023)

https://www.coindesk.com/markets/2018/02/15/mit-professor-raises-4-million-to-build-a-better-blockchain

[66] Tutto Crypto Wiki “Cos’è il Trilemma?” (consultato il 09/03/2023)

https://www.tuttocrypto.it/trilemma

[67] Binance Academy “Cos'è il trilemma della blockchain?” (consultato il 10/03/2023)

https://academy.binance.com/it/articles/what-is-the-blockchain-trilemma

[68] Christian Boscolo “Proof of work VS Proof of stake: qual è il miglior algoritmo di consenso?” (consultato il 24/03/2023)

https://www.cellulare-magazine.it/proof-of-work-vs-proof-of-stake-qual-e-il-miglior-algoritmo-di-consenso