

# Algoritmi di consenso distribuito

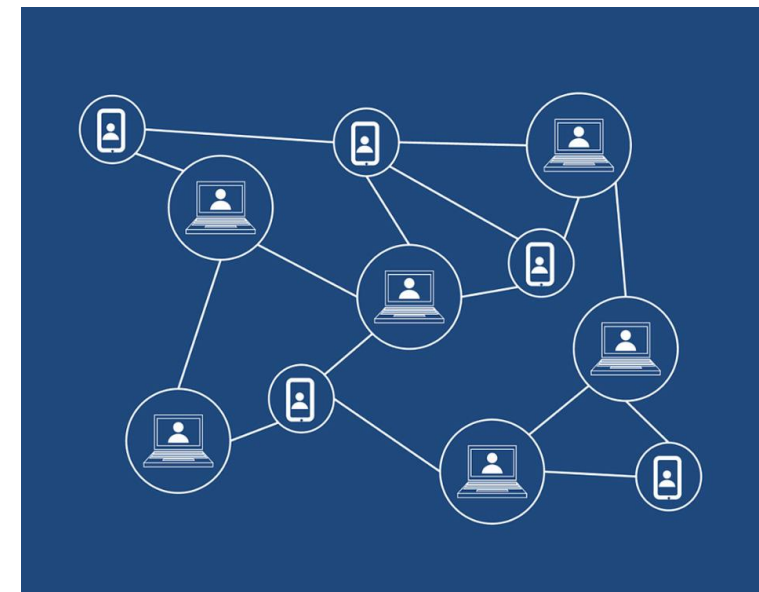
*Francesco Pugliese, PhD*

*neural1977@gmail.com*

# Algoritmi di consenso distribuito

---

- ✓ Un **algoritmo di consenso** è un meccanismo che permette a utenti o dispositivi di coordinarsi in un contesto **distribuito**.
- ✓ Deve garantire che tutti gli agenti nel sistema possano concordare su una singola fonte di verità, anche se alcuni agenti falliscono.
- ✓ In altre parole, il sistema deve essere **fault-tolerant**.



# Algoritmi di consenso distribuito

---

- ✓ In una configurazione centralizzata, una singola entità ha **potere sul sistema**. In gran parte dei casi, possono apportare modifiche come vogliono – non esiste un **complesso sistema** di governance per raggiungere il consenso tra diversi amministratori.
- ✓ In una configurazione decentralizzata, invece, è tutta un'altra storia. Supponiamo di avere un **database distribuito** – come facciamo a raggiungere un accordo su quali voci debbano essere aggiunte?



# Algoritmi di consenso distribuito

---

- ✓ Superare questa sfida in un ambiente in cui sconosciuti non si fidano gli uni degli altri è stato forse lo sviluppo più cruciale per aprire la strada alle **blockchain**.
- ✓ Vediamo come gli algoritmi di consenso sono vitali per il funzionamento delle **criptovalute** e dei registri distribuiti.



# Algoritmi di consenso e criptovalute

---

- ✓ Nelle **criptovalute**, i **saldi** degli utenti vengono registrati in un database – la **blockchain**.
- ✓ E' fondamentale che tutti (o, più precisamente, tutti i **nodi**) mantengano una copia **identica del database**. Altrimenti, finiremmo presto con informazioni contrastanti, compromettendo totalmente lo scopo del **network di criptovaluta**.
- ✓ La **crittografia a chiave pubblica** garantisce che gli utenti non possono spendere le **monete** di altri, ma deve comunque esserci una singola fonte di verità su cui i partecipanti al network si basano, per riuscire a determinare se i fondi sono già stati spesi.

# Funzionamento degli Algoritmi di consenso

---



- ✓ Per prima cosa, chiediamo agli utenti che vogliono aggiungere **blocchi** (chiamiamoli **validatori**) di fornire una **stake**.
- ✓ La **stake** è una qualche sorta di valore che il validatore deve mettere in gioco, con l'obiettivo di **dissuaderlo dall'agire in modo disonesto**. Se imbrogia, perderà la sua posta in gioco.
- ✓ Esempi di **questa stake** includono potenza computazionale, criptovaluta o persino reputazione.

# Funzionamento degli Algoritmi di consenso

---

- ✓ Perché i **validatori** dovrebbero rischiare le proprie risorse? Beh, c'è anche una ricompensa in palio. Questa consiste solitamente nella **criptovaluta nativa** del protocollo ed è composta dalle commissioni pagate da altri utenti, unità di criptovaluta appena generate o entrambi.

L'ultimo elemento di cui abbiamo bisogno è la **trasparenza**. Dobbiamo essere in grado di scoprire quando qualcuno sta imbrogliando. Idealmente, dovrebbe essere costoso produrre **blocchi** ma è un dispendio economico per chiunque verificarli. Ciò garantisce che i **validatori** sono tenuti sotto controllo dagli utenti regolari.

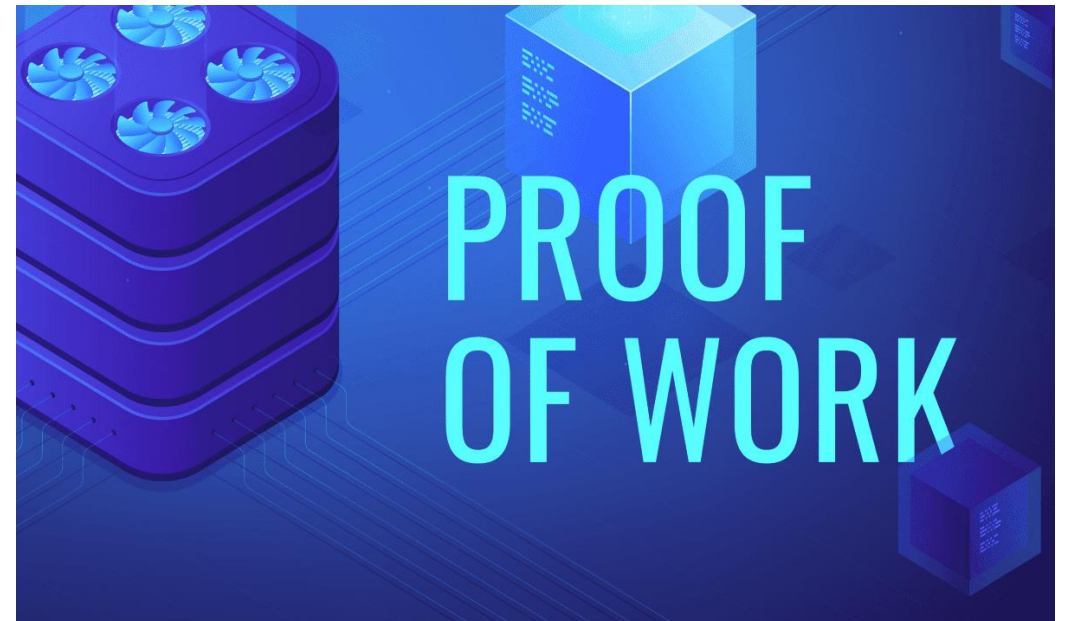




# Tipi di Algoritmi di consenso

---

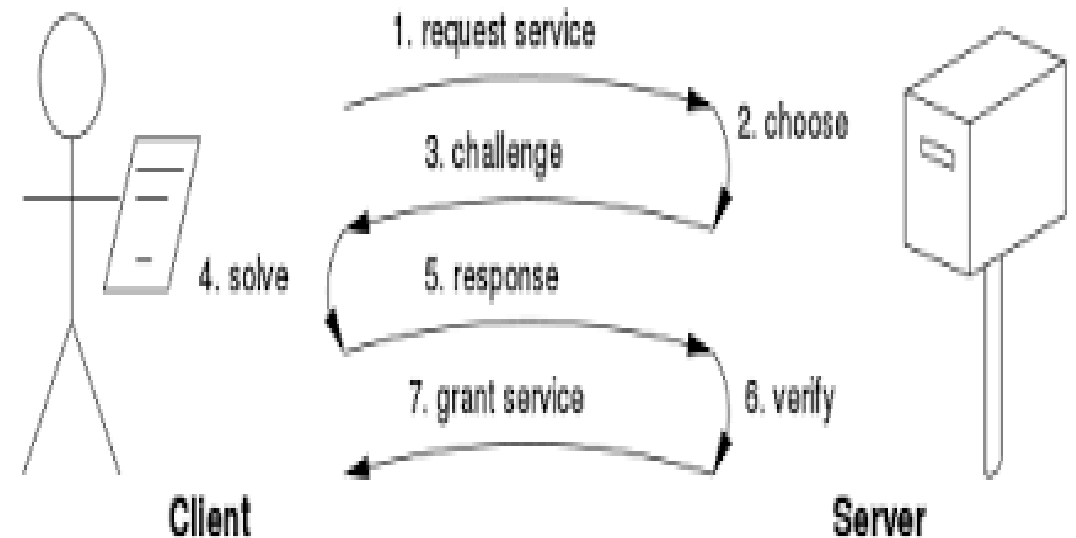
- ✓ La **Proof of Work** è il padrino degli algoritmi di consenso **blockchain**. E' stato implementato per la prima volta in **Bitcoin**, ma il concetto è in circolazione da ben prima. Nella **Proof of Work**, i **validatori** (denominati **miner**) elaborano tramite **hash** i dati che vogliono aggiungere fino a quando non producono una soluzione specifica.





# Tipi di Algoritmi di consenso

- ✓ Una **hash** è una stringa apparentemente **casuale** di lettere e numeri generata dall'elaborazione di dati attraverso una **funzione di hash**. Tuttavia, elaborando gli stessi dati nella stessa funzione, si otterrà lo stesso output. Cambiando anche un solo dettaglio, però, porterà a una **hash completamente differente**.



# Algoritmo Proof-of-Work (PoW)

---

- ✓ Dunque, con il termine **Proof-of-Work (PoW)** si intende l'algoritmo di consenso alla base della rete **Blockchain**.
- ✓ In una **Blockchain**, questo algoritmo viene utilizzato per confermare le transazioni e produrre i **nuovi blocchi** della catena. La **PoW** incentiva i **miner** a competere tra loro nell'elaborazione degli scambi, ricevendo in cambio una **ricompensa**.
- ✓ All'interno della **Blockchain**, gli utenti inviano beni digitali l'uno all'altro. Un **registro** decentralizzato raccoglie ogni singola transazione: tuttavia, per poter essere considerate valide, queste devono essere prima approvate e organizzate in blocchi.

# Algoritmo Proof-of-Work (PoW)

---

- ✓ Tale responsabilità ricade su speciali nodi chiamati **miner**; l'intero processo viene invece definito **mining**.
- ✓ Alla base di questo sistema troviamo **complessi problemi matematici** e la necessità di dimostrare semplicemente la soluzione.
- ✓ La **velocità** e l'**esattezza** di un sistema di **Blockchain** dipendono dalla difficoltà dei problemi. Ma i problemi non dovrebbero essere eccessivamente complessi, poiché in tal caso la **generazione di nuovi blocchi** richiederebbe troppo tempo, le transazioni non verrebbero elaborate ed il flusso della rete si bloccherebbe. Se il problema non ha un tempo di risoluzione ben definito, generare **nuovi blocchi** sarebbe praticamente impossibile.

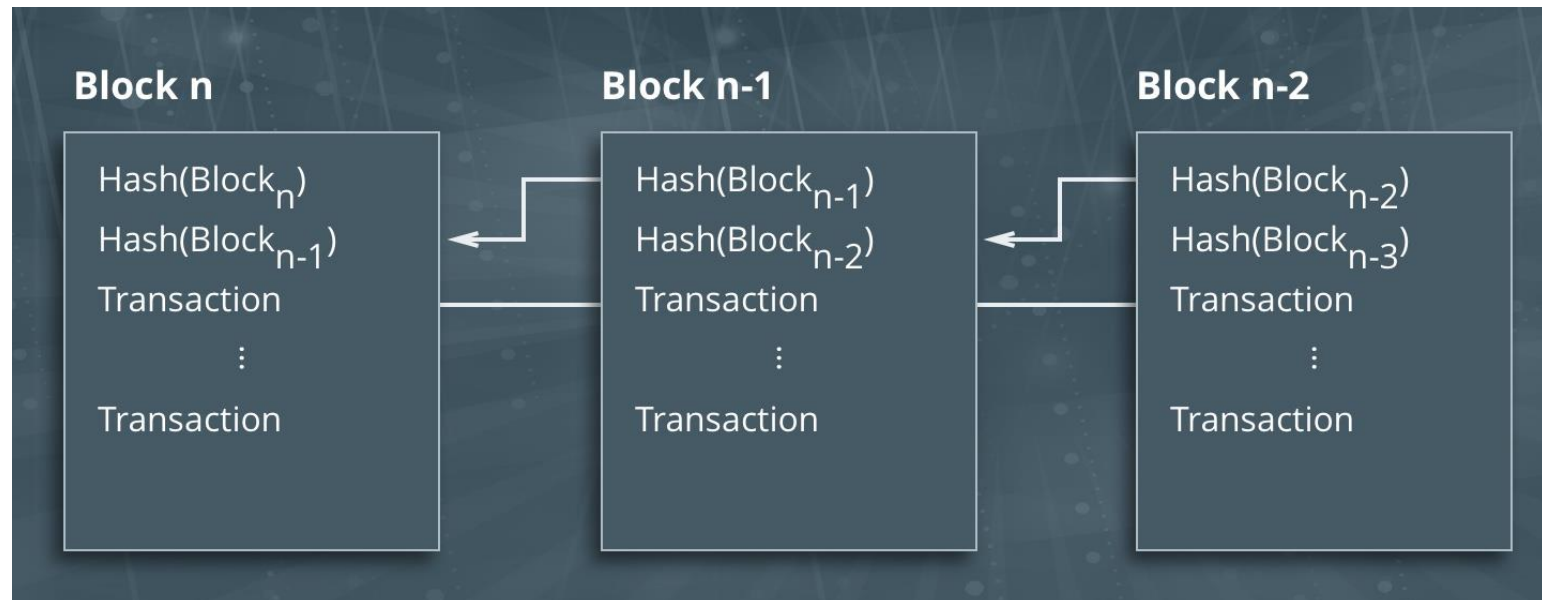
# Algoritmo Proof-of-Work (PoW)

---

- ✓ Al contrario, se il problema fosse troppo semplice, la rete diverrebbe estremamente vulnerabile ad **attacchi esterni**.
- ✓ Inoltre la soluzione deve poter essere **controllata** con estrema **semplicità** da ogni macchina, in quanto non tutti i **nodi** potrebbero essere capaci di appurare che i calcoli siano stati eseguiti correttamente. In tal caso questi nodi dovrebbero far affidamento su altri utenti, violando uno dei principi fondamentali della Blockchain: la **trasparenza**.
- ✓ I **miner** risolvono il problema, danno vita ad un nuovo blocco e confermano tutte le transazioni al suo interno.

# Algoritmo Proof-of-Work (PoW)

- ✓ La **complessità** del problema dipende dal **numero di utenti**, dalla potenza di calcolo disponibile e dal carico della rete. La **hash** di ogni blocco contiene la hash del blocco precedente, incrementando la sicurezza ed impedendo ogni sorta di violazione informatica.



# Algoritmo Proof-of-Work (PoW)

---

- ✓ Quando un **miner** riesce a risolvere il problema, il nuovo blocco viene creato e le **transazioni** vengono piazzate al suo interno.
- ✓ La **Proof-of-Work** sta alla base di parecchie **criptovalute**.
- ✓ La più popolare applicazione della **PoW** è il **Bitcoin**: è stata questa **criptovaluta** a gettare le basi per tale tipologia di consenso.
- ✓ Il problema viene definito **Hashcash**, e l'algoritmo cambia la propria difficoltà in maniera **dinamica** a seconda della potenza di calcolo disponibile nella rete. Il tempo di creazione di un blocco è di circa **10 minuti**. Anche altre valute basate sul Bitcoin, come il **Litecoin**, utilizzano un simile sistema.



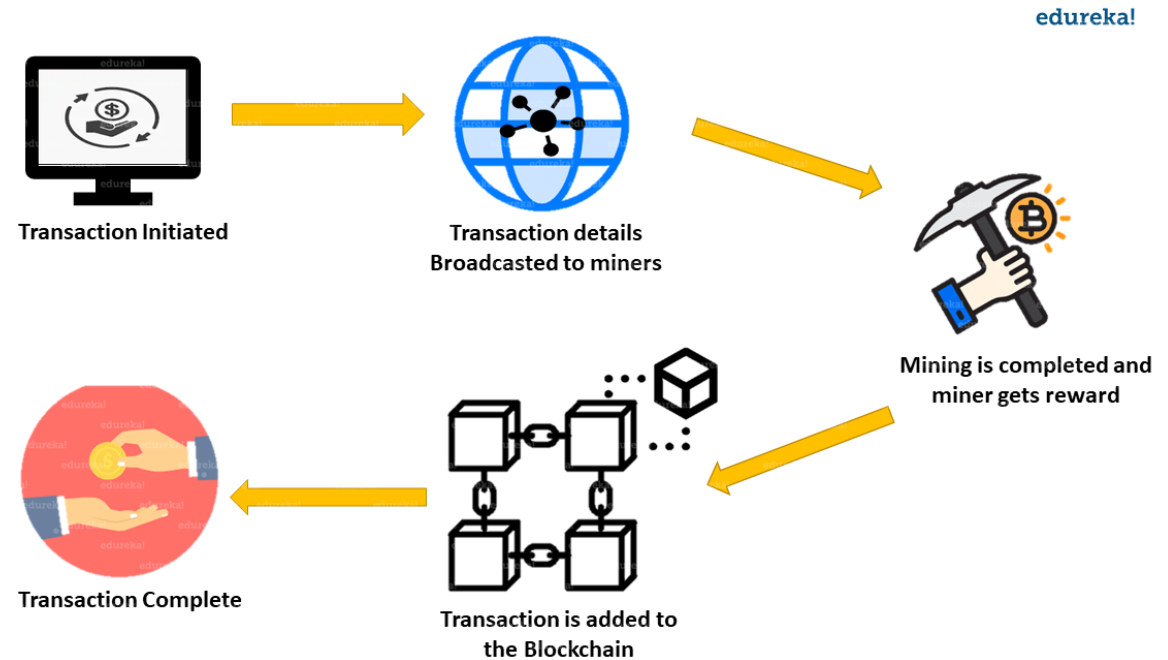
# Algoritmo Proof-of-Work (PoW)

---

- ✓ Un altro importante progetto basato sulla **PoW** è **Ethereum**: poiché nel mondo delle **criptovalute** circa il **75%** dei progetti si basano su **Ethereum**, è possibile affermare che la maggior parte delle applicazioni **Blockchain** sfruttano il modello di consenso **PoW**.
- ✓ I principali vantaggi offerti da un sistema **PoW** sono **un'ottima difesa** contro gli attacchi **DoS (Denial of Service)** e l'impatto marginale delle quote nel mining.
- ✓ **Difesa** contro gli attacchi **DoS**. La **PoW** impone parecchi limiti alle azioni che è possibile intraprendere sulla rete, ed un attacco efficiente richiederebbe moltissimo tempo ed una potenza di calcolo incredibile.
- ✓ Nonostante quindi gli attacchi **DoS** ad una **Blockchain** siano in teoria possibili, in pratica i risultati sarebbero deludenti ed i costi estremamente elevati.

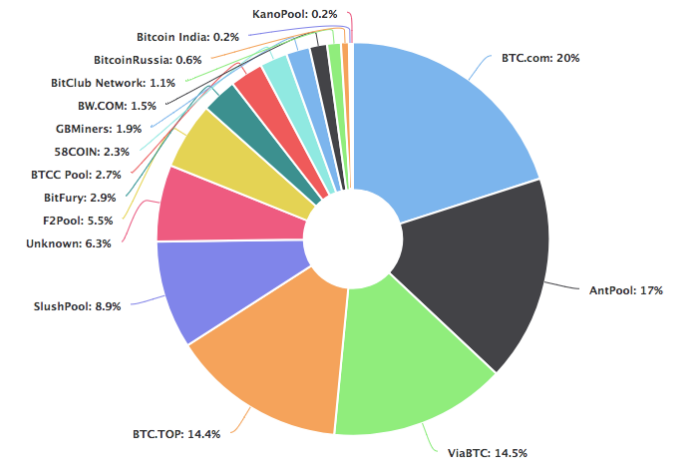
# Algoritmo Proof-of-Work (PoW)

- ✓ **Mining:** Non importa quanto sia alta la percentuale delle quote nel proprio **portafoglio**: in un sistema **PoW** l'unica cosa che conta è la potenza di calcolo utilizzata per risolvere i problemi matematici e generare nuovi blocchi. Chi possiede grosse quantità di denaro, quindi, non ha maggiore controllo sulla rete.



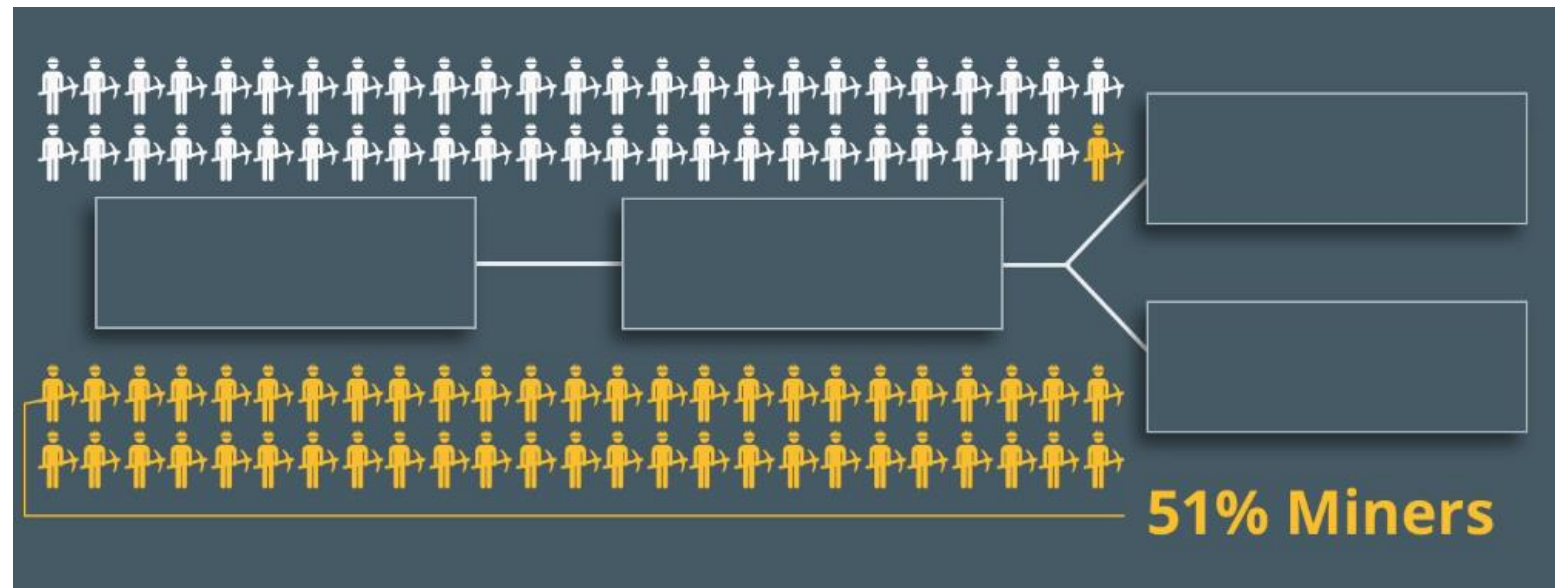
# Algoritmo Proof-of-Work (PoW)

- ✓ I principali svantaggi di un sistema **PoW** sono invece i costi elevati, la totale inutilità dei calcoli e il rischio di attacchi sono al **51%**.
- ✓ **Costi elevati:** Il processo di **mining** richiede macchine altamente specializzate, capaci di risolvere in tempi brevi algoritmi estremamente complessi.
- ✓ Questi dispositivi non sono solo estremamente costosi, ma consumano anche **enormi quantità di energia elettrica**, incrementando ulteriormente i prezzi. Si tratta di una pericolosa minaccia alla **decentralizzazione del sistema**, in quanto solo una piccola fetta dell'utenza può permettersi questo genere di investimenti. Diamo ad esempio un'occhiata al seguente grafico sul **Bitcoin**.



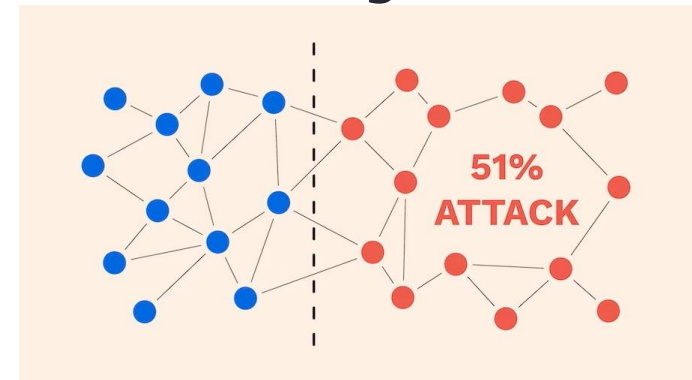
# Attacchi 51%

- ✓ **Inutilità dei calcoli:** I **miner** consumano moltissimo tempo ed energie per generare nuovi blocchi, **eseguendo calcoli** fine a se stessi, non applicabili a nessun altro settore. I problemi garantiscono quindi la sicurezza della rete, ma non possono essere utilizzati in campo **economico o scientifico**.
- ✓ Con **attacco 51%**, o **attacco di maggioranza**, s'intende il caso in cui un singolo utente o un gruppo di individui riesca a controllare la **maggior parte della potenza di mining** di una rete.



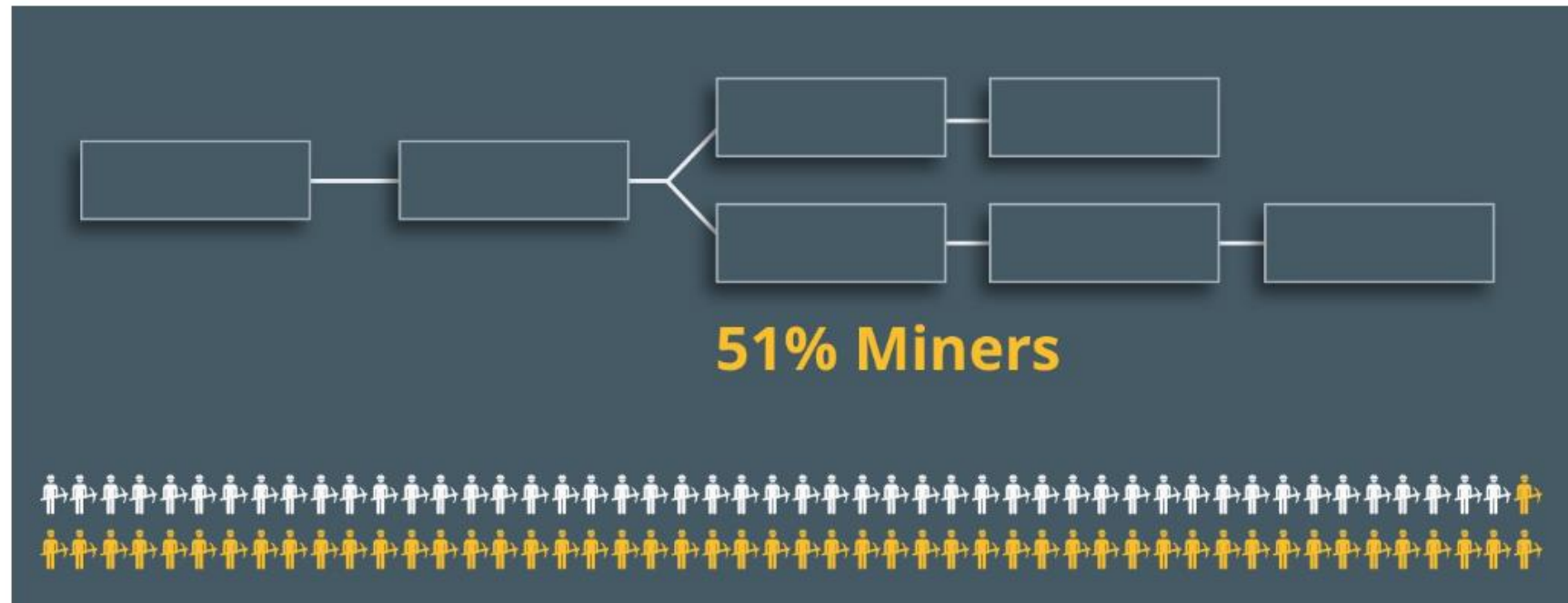
# Attacchi 51%

- ✓ In questo caso gli **aggressori** ottengono una tale influenza nella rete da poter controllare l'esito degli **eventi** che avvengono al suo interno.
- ✓ Potrebbero ad esempio monopolizzare la generazione di **nuovi blocchi**, ostacolando gli altri **miner** e impedendo loro di ricevere le ricompense. Oppure potrebbero addirittura **annullare** le transazioni.
- ✓ Supponiamo che **Alice** abbia inviato dei fondi a **Bob** utilizzando una **Blockchain**. Alice controlla il **51%** della rete, mentre Bob ovviamente no: la transazione viene effettivamente piazzata all'interno del blocco, ma la prima non acconsente al trasferimento di denaro. In tal caso viene generata una **'fork'** della catena.



# Attacchi 51%

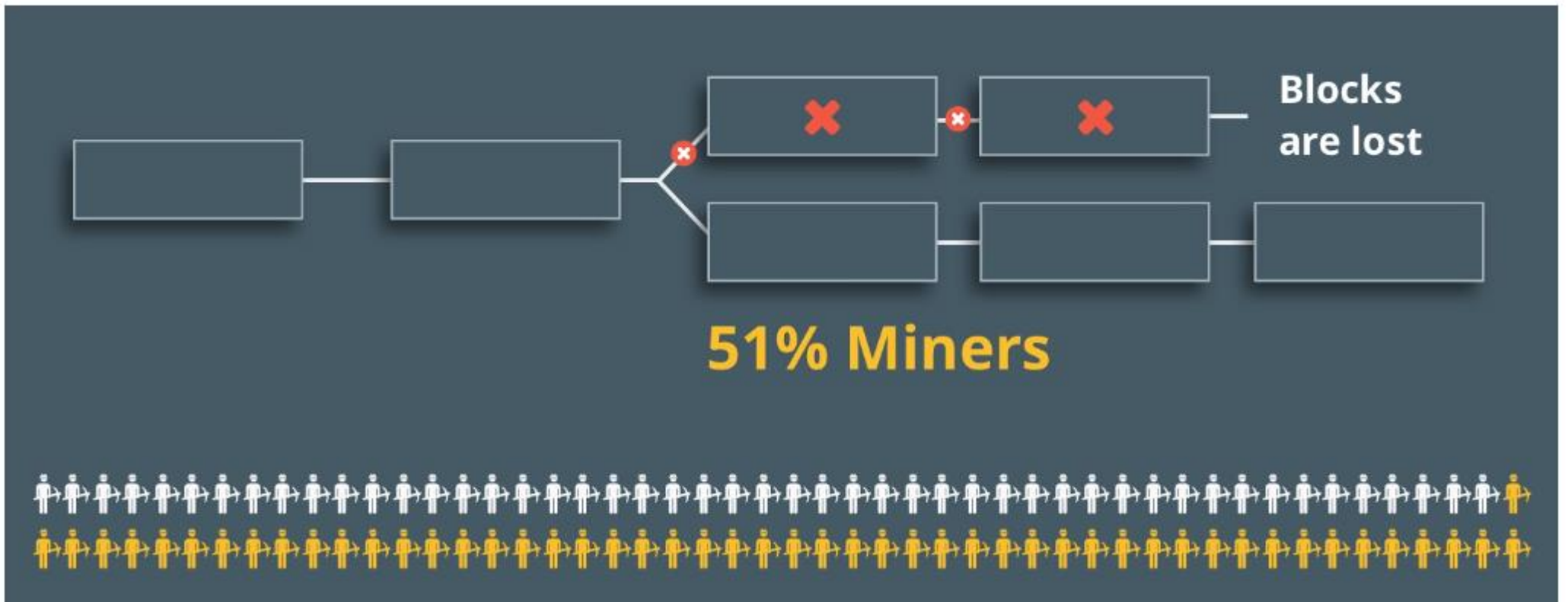
- ✓ In questo caso gli **aggressori** ottengono una tale **influenza** nella rete da poter controllare l'esito degli **eventi** che avvengono al suo interno. Successivamente i miner si uniscono ad uno dei due rami appena generati. In quanto Alice possiede la maggior parte della **potenza computazionale** della rete, la sua **catena** conterrà sempre più blocchi dell'altra.





# Attacchi 51%

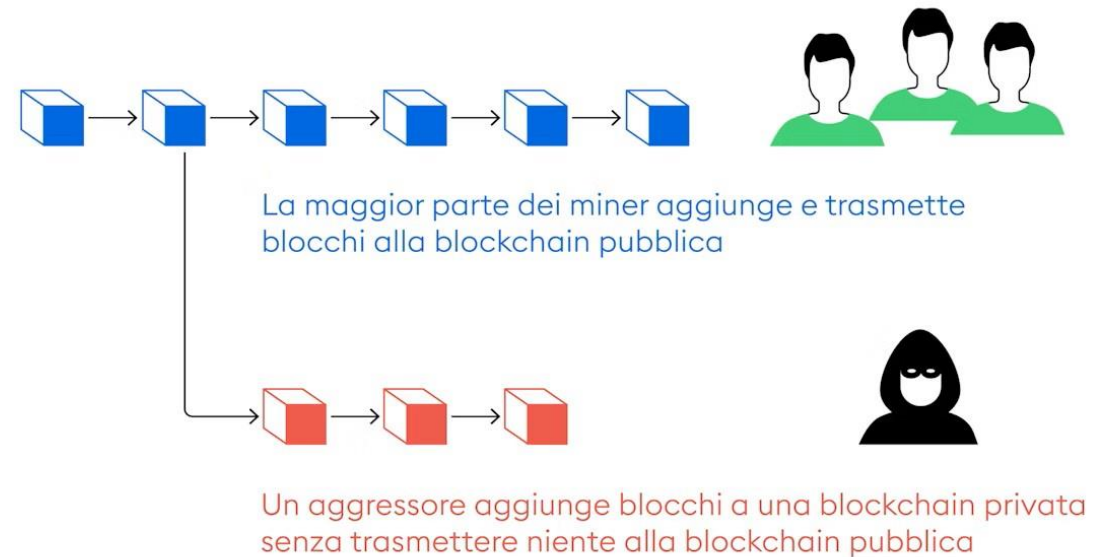
- ✓ In una **Blockchain**, il ramo più lungo rimane, mentre quello più corto viene scartato. Quindi la transazione tra Alice e Bob è come se non fosse mai avvenuta, e quest'ultimo non riceverà mai i propri soldi.



# Attacchi 51%

- ✓ Fortunatamente un attacco 51% non è **per nulla redditizio**. Servirebbe una **potenza di calcolo** incredibile per poter controllare l'intera **Blockchain**, ed una volta che la notizia raggiunge il resto degli utenti, la rete viene considerata compromessa e **prontamente abbandonata**.
- ✓ Questo porterebbe inevitabilmente ad una **diminuzione del valore della criptovaluta**, e di conseguenza dei fondi in possesso degli esecutori dell'attacco.

## Cos'è un attacco al 51%?



# Attacchi 51%

---

- ✓ In sostanza, un "**attacco al 51%**" si verifica quando un **singolo miner** o un **gruppo di miner** prende il controllo della maggioranza di una blockchain basata sulla **Proof of Work** e spende due volte alcune delle sue monete.
- ✓ Un **attacco al 51%** è probabilmente il problema **più temuto** nell'intero settore delle blockchain.
- ✓ In un **attacco al 51%**, un miner o un gruppo di miner guadagna o acquista **abbastanza potenza di hash** da prendere il **controllo al 51%** o più di una rete blockchain e spendere due volte le stesse criptovalute.
- ✓ Finora **nessun** attacco al 51% ha avuto successo sulla **blockchain di Bitcoin**, ma è successo in reti di altcoin con molto meno hashpower e con una sicurezza più scarsa.

# Bibliografia

---

<https://academy.binance.com/it/articles/what-is-a-blockchain-consensus-algorithm>

<https://it.cointelegraph.com/explained/proof-of-work-explained>

<https://www.edureka.co/blog/blockchain-mining/>

<https://www.bitpanda.com/academy/it/lezioni/cos-e-un-attacco-al-51-e-come-si-previene/>