



Lezione 3 – Blockchain

Prof. Esposito Christian

Corso di Sicurezza dei Dati



::: Sommario

- Introduzione alle Blockchain
 - Transazioni e trusted entities;
 - Distributed ledger;
 - Il problema del double spending;
 - Classificazione delle blockchain;
 - Consenso.

... Riferimenti e Letture

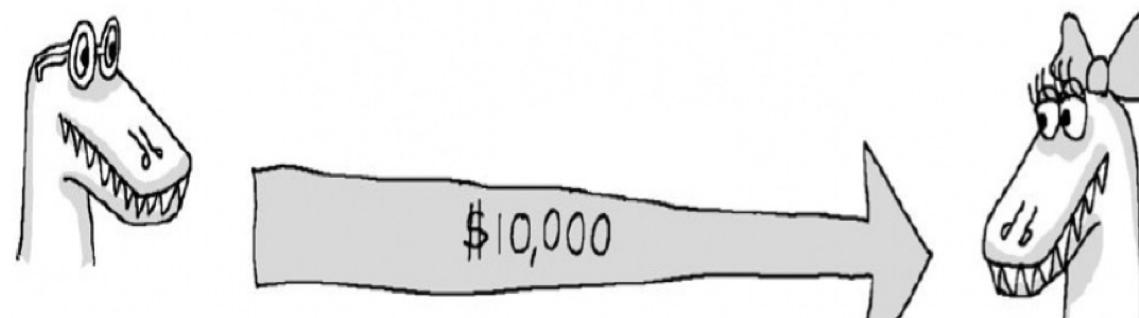
- W. Wang et al., "A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks", in IEEE Access, vol. 7, pp. 22328-22370, 2019.
- Y. Xiao, N. Zhang, W. Lou and Y. T. Hou, "A Survey of Distributed Consensus Protocols for Blockchain Networks", in IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 1432-1465, Second quarter 2020.
- Li, Yi, Jinsong Wang, and Hongwei Zhang. "A survey of state-of-the-art sharding blockchains: Models, components, and attack surfaces." Journal of Network and Computer Applications 217 (2023): 103686.
- Zhou, Qiheng, et al. "Solutions to scalability of blockchain: A survey." Ieee Access 8 (2020): 16440-16455.



Introduzione alle Blockchain

::: Trasferimento di un Bene (1/2)

- Due nodi di una rete intendono scambiarsi un bene.
- L'obiettivo è riuscire a trasferire tale bene in maniera affidabile e sicura.



::: Trasferimento di un Bene (2/2)

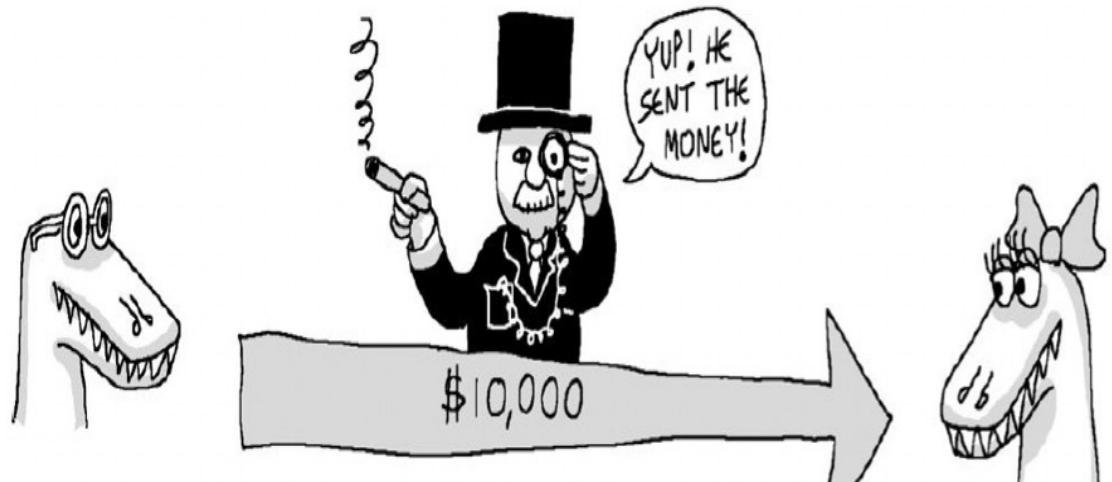
Modello di Fallimento:

- **Fallimenti dei nodi** – crash o hang.
- **Fallimenti dei link** – crash persistente o intermittente.
- **Malfunzionamenti di rete** – perdita, alterazione o ritardo anormale di pacchetti o partizionamento della rete.

Modello di attacco:

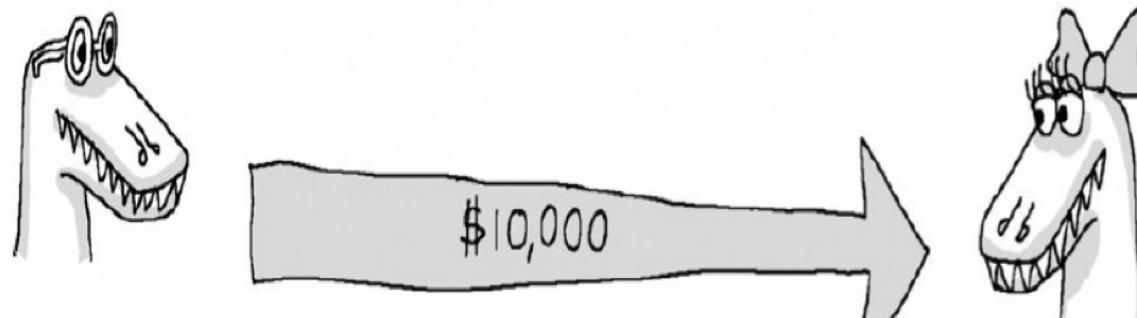
- **Replay Attack**: una transazione valida viene maliziosamente ripetuta o ritardata;
- **Man-in-the-middle Attack**: la comunicazione è osservata e alterata da una terza parte non autorizzata
- **Masquerade Attack**: un nodo malizioso utilizza un'identità forgiata ad hoc per la comunicazione
 - Caso particolare: un nodo impersona un altro (*impersonation*).
- **Byzantine Behaviour**: un nodo si comporta in modo non previsto.

::: Transazioni con Mediatore



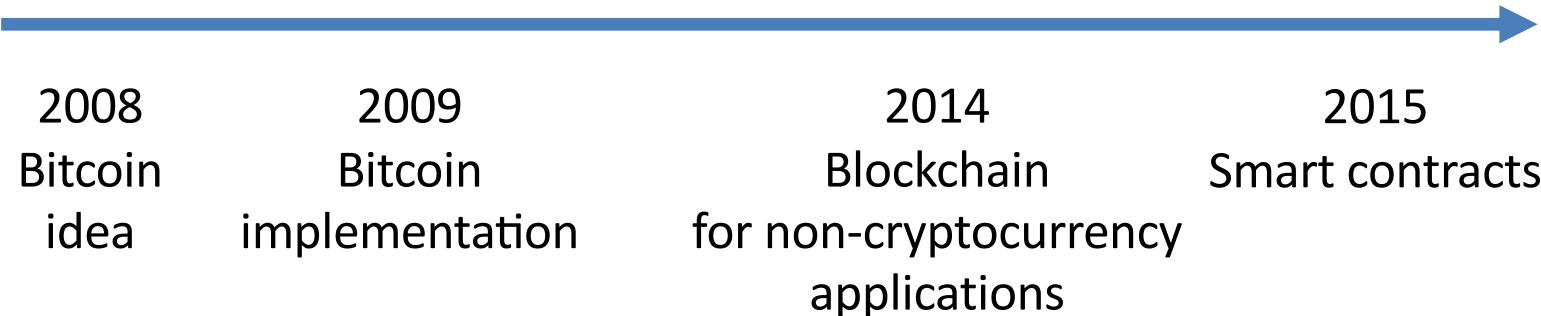
- La gestione delle transazioni è realizzata impiegando una entità di terze parti (e.g. una banca) che *convalida*, *supervisiona* e *preserva* le transazioni.
- Tale soluzione è implementata ad esempio con il protocollo di aggiornamento a due fasi (*2PC* - Two-phase commit protocol).
- Tipicamente, la soluzione con un mediatore consente di gestire attacchi del tipo **Masquerade** o **Byzantine**. Si richiede che la terza parte sia fidata ed affidabile.
- Il costo di tale operazione è elevato (rappresenta un bottleneck prestazionale, un single-point-of-failure, una vulnerabilità).

::: Transazioni senza Mediatore



- Per inviare un bene senza la collaborazione di un'entità di terze parti si può definire un protocollo per eseguire transazioni affidabili e sicure in un ambiente inaffidabile e insicuro.
- Tale soluzione presenta diverse problematiche legate a:
 - Verifica dell'integrità del messaggio ricevuto;
 - Verifica dell'identità dei nodi partecipanti;
 - Verifica della validità delle transazioni;
 - ...

::: Blockchain & Bitcoin

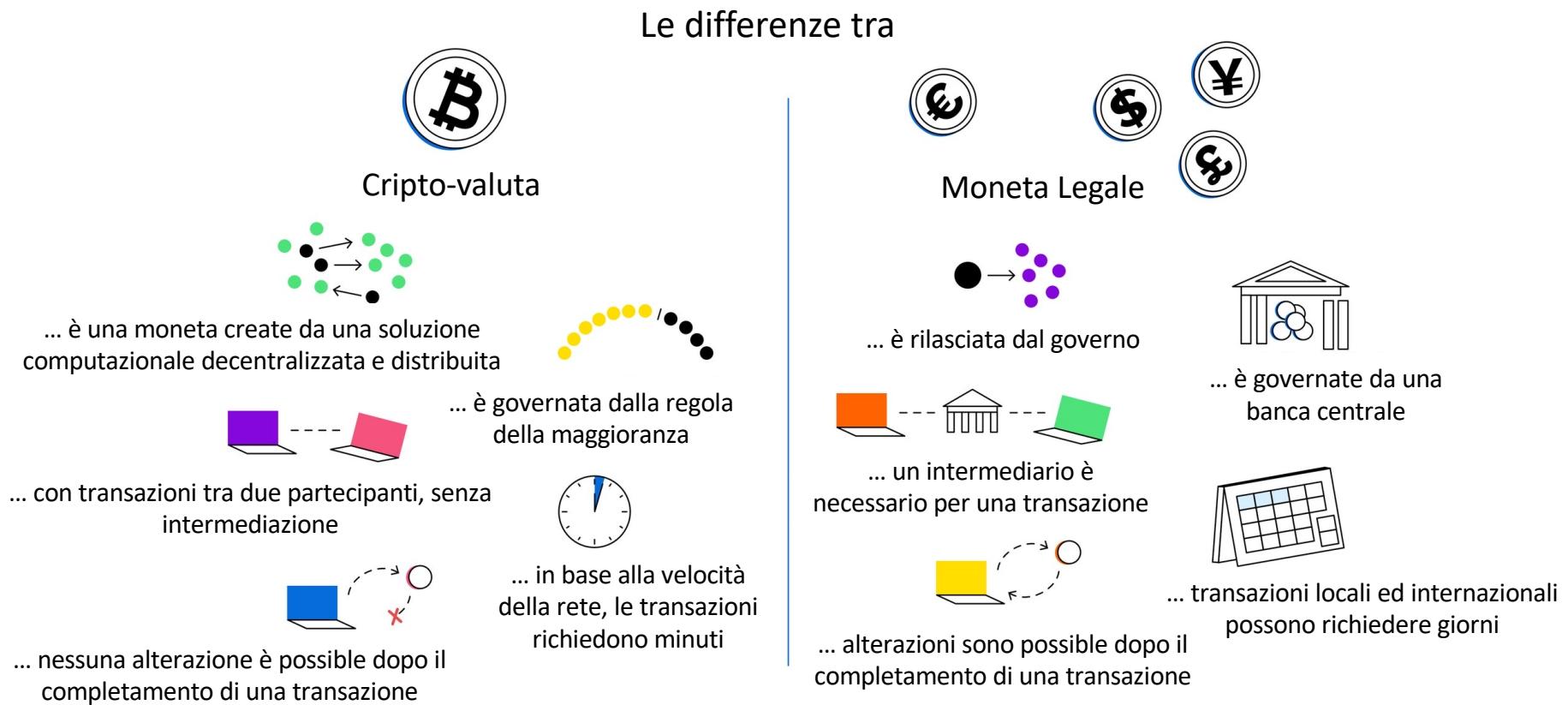


- Il primo ambito di applicazione (2008) è quello finanziario, in particolare per la gestione delle criptovalute. Lo scopo era quello di eseguire transazioni sicure utilizzando un'infrastruttura non sicura.
- In generale, l'obiettivo della tecnologia Blockchain è *creare un ambiente decentralizzato in cui nessuna "terza parte" abbia il controllo delle transazioni e dei dati.*
 - *No trusted entities* (e.g., bank)

::: Cripto-Valute (1/3)

Le blockchain sono state teorizzate a supporto delle cosiddette cripto-valute:

- una rappresentazione digitale di valore basata sulla crittografia e senza intermediazione.



::: Cripto-Valute (1/3)

Le blockchain sono state teorizzate a supporto delle cosiddette cripto-valute:

- una rappresentazione digitale di valore basata sulla crittografia e senza intermediazione.

Ha un valore grazie al fatto che esiste un'autorità (lo Stato) che agisce come se avesse questo valore.

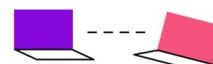
Se un'organizzazione abbastanza grande emette, usa e accetta qualcosa come pagamento, automaticamente quel qualcosa acquisisce valore.

... è una moneta creata da una soluzione computazionale decentralizzata e distribuita

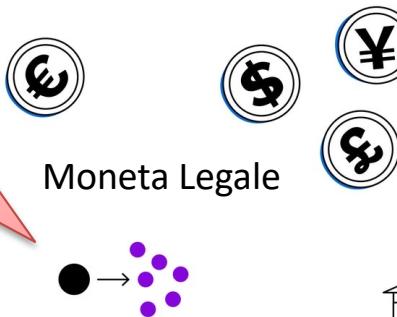
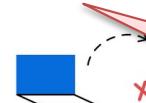


Il valore di una criptovaluta viene regolato dalla domanda e dall'offerta del mercato, e non può essere manipolato in quanto la creazione dei nuovi Bitcoin è regolata matematicamente e non "stampate" su comando delle varie Banche Centrali sparse per il mondo.

... con transazioni tra due partecipanti, senza intermediazione



... nessuna alterazione è possibile dopo il completamento di una transazione

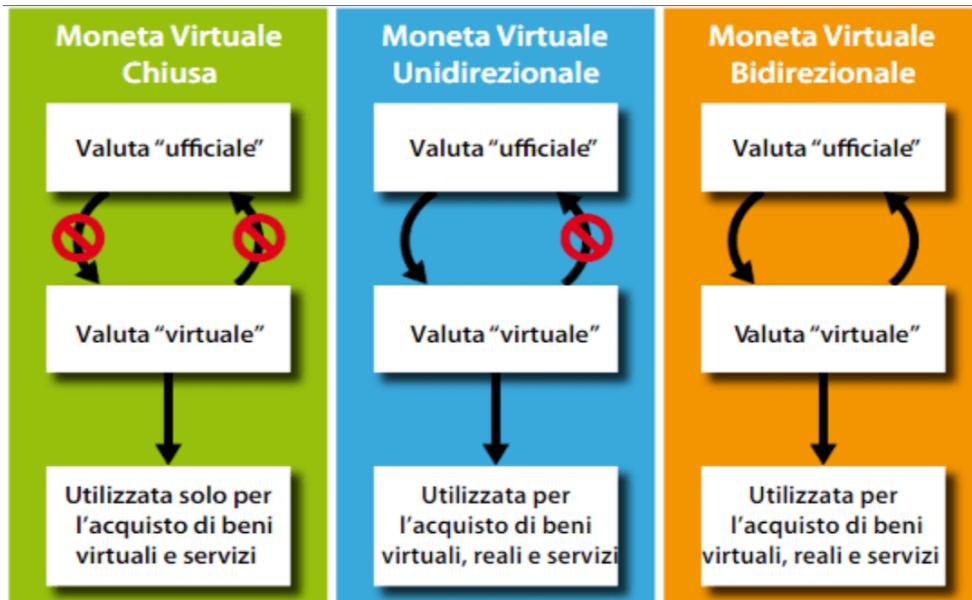


Ha un valore non coperto da riserve di altri materiali (ad esempio: riserve auree), e quindi privo di *valore intrinseco*.

La criptovaluta non esiste in forma fisica, ma si genera e si scambia esclusivamente per via telematica.

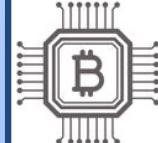
::: Cripto-Valute (2/3)

Le cripto-valute rientrano nel caso delle valute virtuali ma sono diverse dalle valute digitali:

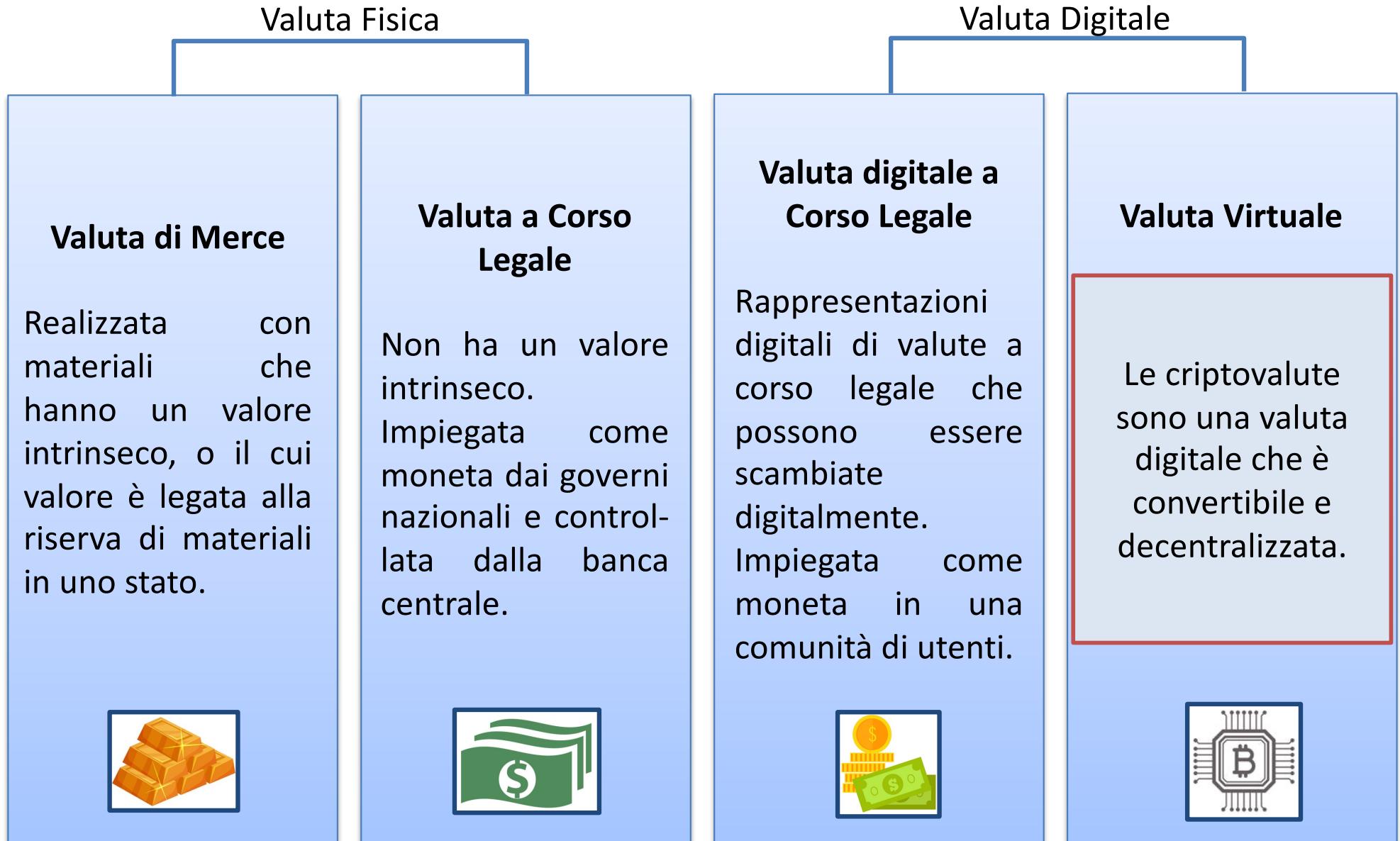


- Una valuta virtuale è una rappresentazione digitale di valore che non è né emessa da una banca centrale né da un'autorità pubblica, né necessariamente collegata a una valuta a corso legale, ma è accettata come mezzo di pagamento e possono essere trasferiti, archiviati o scambiati elettronicamente.
- Le cripto-valute sono valute virtuali bidirezionali.
- Una valuta digitale è una rappresentazione nel virtuale di una valuta a corso legale.

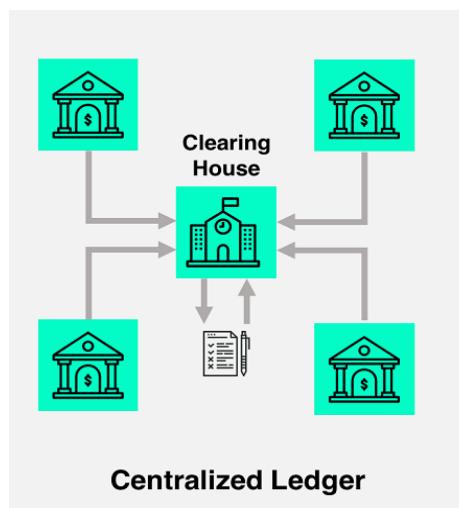
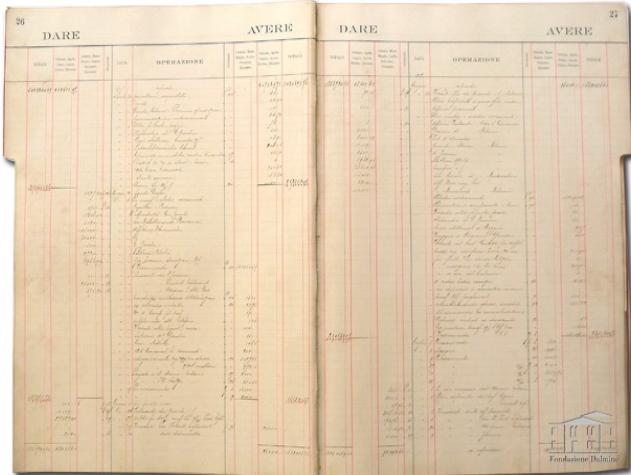
::: Cripto-Valute (3/3)

Valuta Fisica	Valuta Digitale
Valuta di Merce <p>Realizzata materiali che hanno un valore intrinseco, o il cui valore è legata alla riserva di materiali in uno stato.</p> 	Valuta a Corso Legale <p>Non ha un valore intrinseco. Impiegata come moneta dai governi nazionali e controllata dalla banca centrale.</p> 
Valuta digitale a Corso Legale <p>Rappresentazioni digitali di valute a corso legale che possono essere scambiate digitalmente. Impiegata come moneta in una comunità di utenti.</p> 	Valuta Virtuale <p>Rappresentazioni digitali di valori che possono essere scambiate digitalmente. Non hanno lo status di valuta legale.</p> 

::: Cripto-Valute (3/3)



::: Libro Mastro

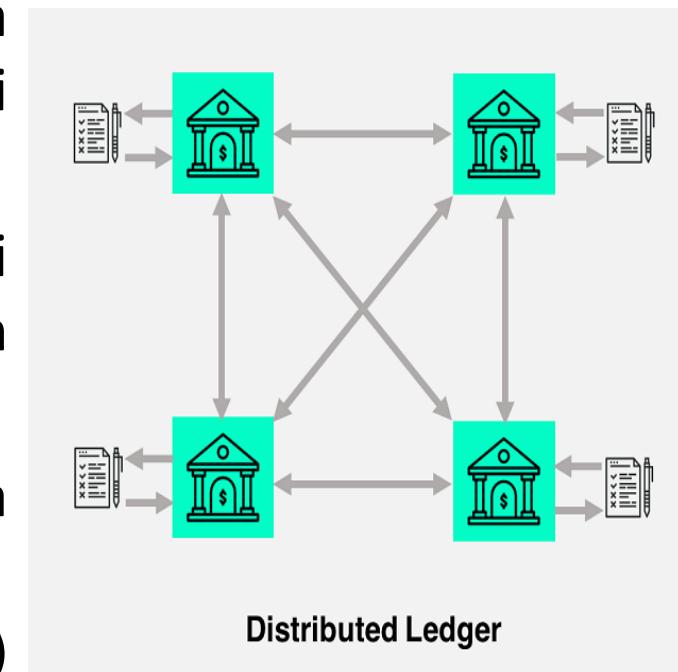


- Il **ledger** (o **libro mastro**) è un registro in cui sono contenute tutte le transazioni eseguite tra i partecipanti di un sistema.
- Il libro mastro è caratterizzato da **scritture sistematiche**; le scritture sono registrate in base all'oggetto a cui si riferiscono.
- Con riferimento ad ogni oggetto le operazioni sono indicate rispettando l'**ordine cronologico**.
- Si tratta di un "*append-only multi-party system of record*", ovvero un "log" di transazioni.

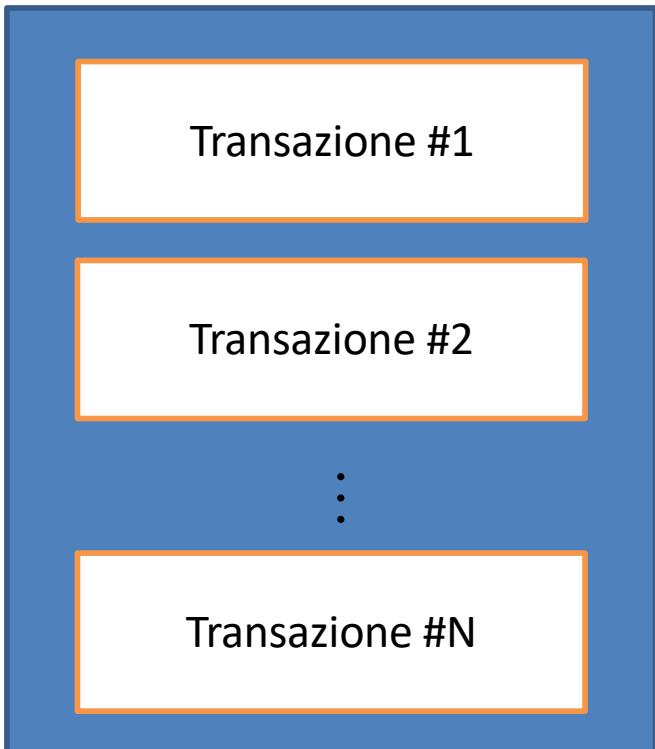


::: Libro Mastro Distribuito

- In un *libro mastro distribuito*, ciascun nodo della rete memorizza record (transazioni) in una copia locale del libro in maniera sequenziale, sotto forma di **blocchi** ordinati temporalmente
- Affinché i blocchi siano considerati validi, i partecipanti devono raggiungere un **quorum** di consenso
- Il *libro mastro* è costruito come una catena di blocchi
 - Ogni blocco (ad eccezione del *genesis block*) contiene un'informazione riguardo il blocco precedente nella catena.



::: Blocco



Un **blocco** è una collezione di transazioni



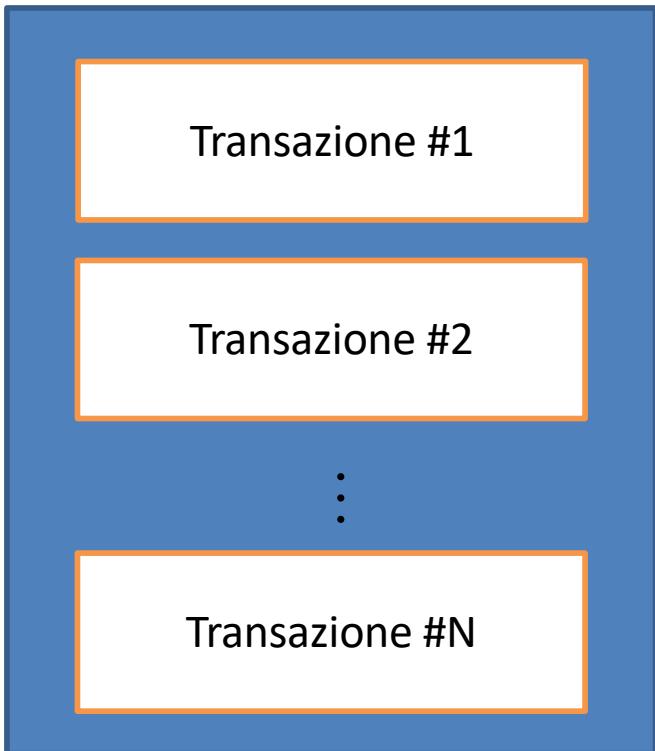
Struttura dati con tre parti:

- nonce (ctr), dati (x), riferimento (s);
 - In genere chiamato intestazione del blocco.
- data (x) dipende dall'applicazione:
- In Bitcoin memorizza i dati finanziari (basati su "UTXO");
 - In Ethereum memorizza i dati del contratto (basato sull'account)
 - In Namecoin memorizza i dati del nome
 - Per ora lo lasciamo indefinito - torneremo su questo più avanti.

Validità del blocco:

- I dati devono essere validi (validità definita dall'applicazione)

::: Blocco



Un **blocco** è una collezione di transazioni



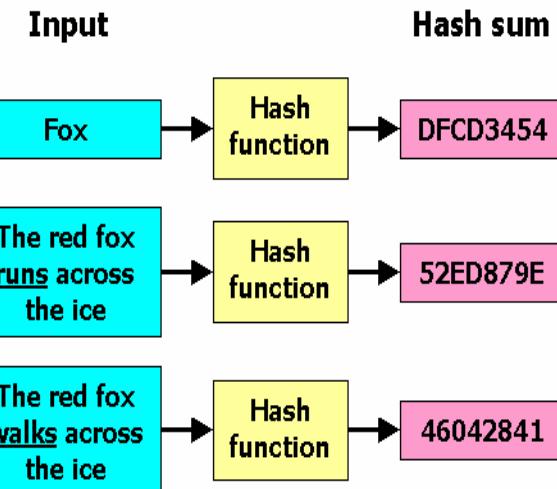
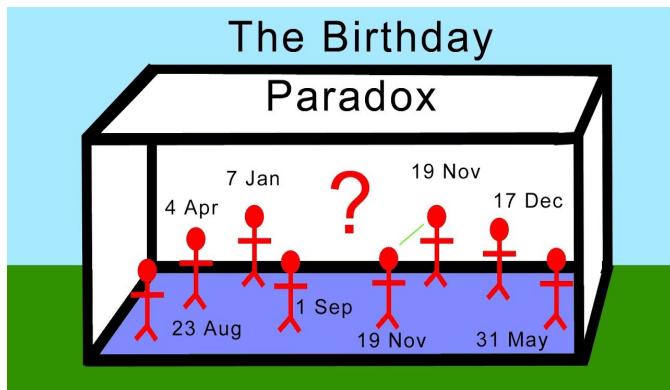
Come collego i blocchi per costruire la catena?

- Per inviare un bene senza la collaborazione di un'entità di terze parti si può definire un protocollo per eseguire transazioni affidabili e sicure in un ambiente inaffidabile e insicuro.
- Tale soluzione presenta diverse problematiche legate a:
 - Verifica dell'**integrità** del messaggio ricevuto;
 - Verifica dell'**identità** dei nodi partecipanti;
 - Verifica della **validità** delle transazioni;
 - ...

Come risolve tali problematiche?

::: Hash e Paradosso del Compleanno

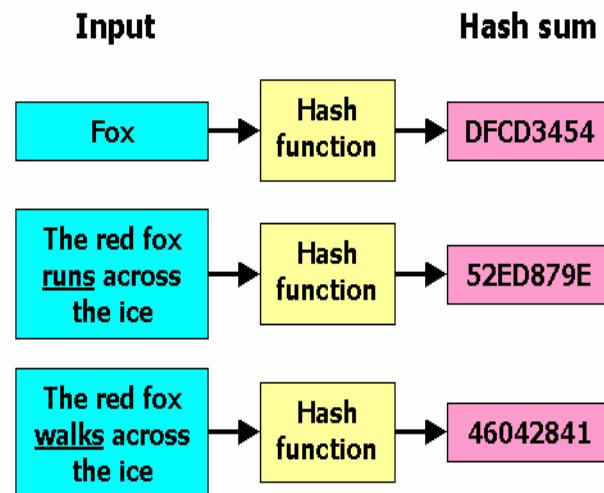
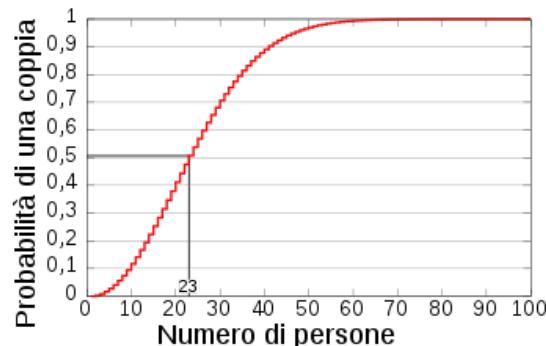
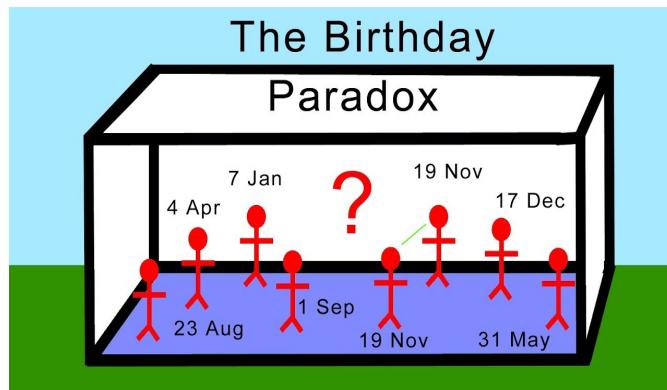
- Le **funzioni di hash** restituiscono una stringa di poche centinaia di bit (definita **hash** o **impronta digitale**) a partire da una sequenza di bit di lunghezza arbitraria.



- La possibilità che due sequenze di bit sottomesse alla stessa funzione di hash diano origine allo stesso valore (*collisione*) è definito «paradosso del compleanno»:

::: Hash e Paradosso del Compleanno

- Le **funzioni di hash** restituiscono una stringa di poche centinaia di bit (definita **hash** o **impronta digitale**) a partire da una sequenza di bit di lunghezza arbitraria.



- La possibilità che due sequenze di bit sottomesse alla stessa funzione di *hash* diano origine allo stesso valore (*collisione*) è definito «paradosso del compleanno»:
 - la probabilità che almeno due persone in un gruppo compiano gli anni lo stesso giorno è largamente superiore a quanto potrebbe dire l'intuito

::: Hash e Paradosso del Compleanno

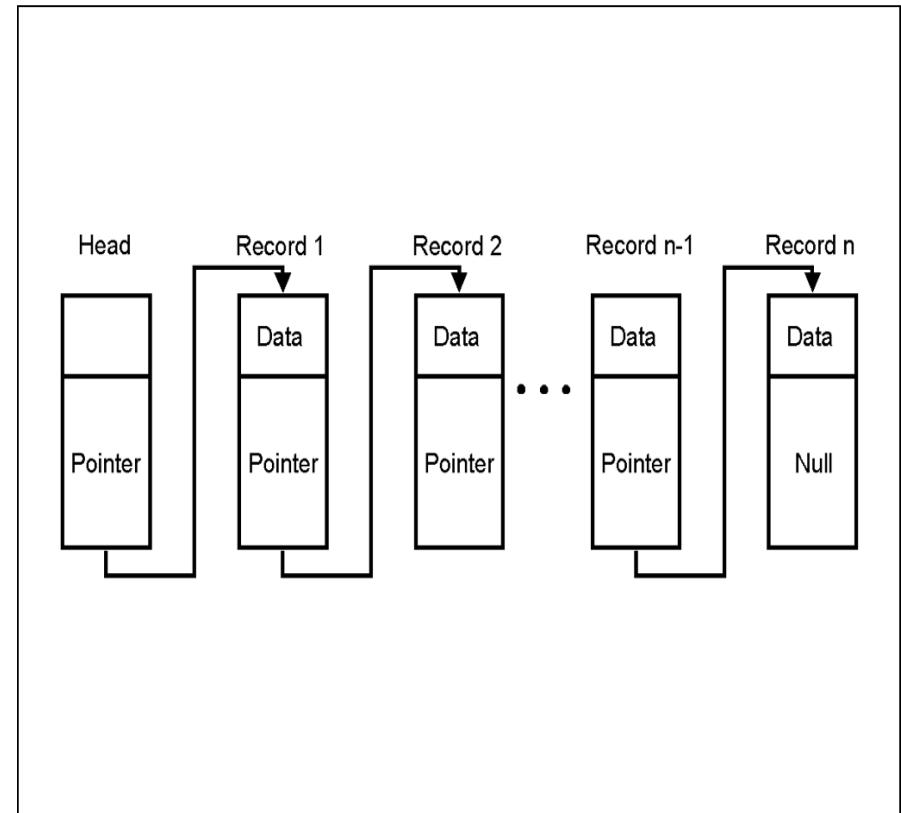
- Le **funzioni di hash** restituiscono una stringa di poche centinaia di bit (definita **hash** o **impronta digitale**) a partire da una sequenza di bit di lunghezza arbitraria.

Una funzione di hash che produce un risultato su N bit sarà reputata insicura quando verranno generati $2^{\frac{N}{2}}$ risultati in quanto si ha la probabilità di oltre il 50% di aver trovato una collisione, il risultato evidentemente è ben al di sotto dei 2^{N-1} elementi necessari suggeriti dall'intuito.

- | Input | Hash sum |
|---|----------|
| Fox | DFCD3454 |
| The red fox
<u>runs across</u>
the ice | 52ED879E |
| The red fox
<u>walks across</u>
the ice | 46042841 |
- La possibilità che due sequenze di bit sottomesse alla stessa funzione di *hash* diano origine allo stesso valore (*collisione*) è definito "paradosso del compleanno".
 - Le funzioni di *hash* crittografiche cercano di rendere **computazionalmente inammissibile** trovare due sequenze che producono la stessa impronta.

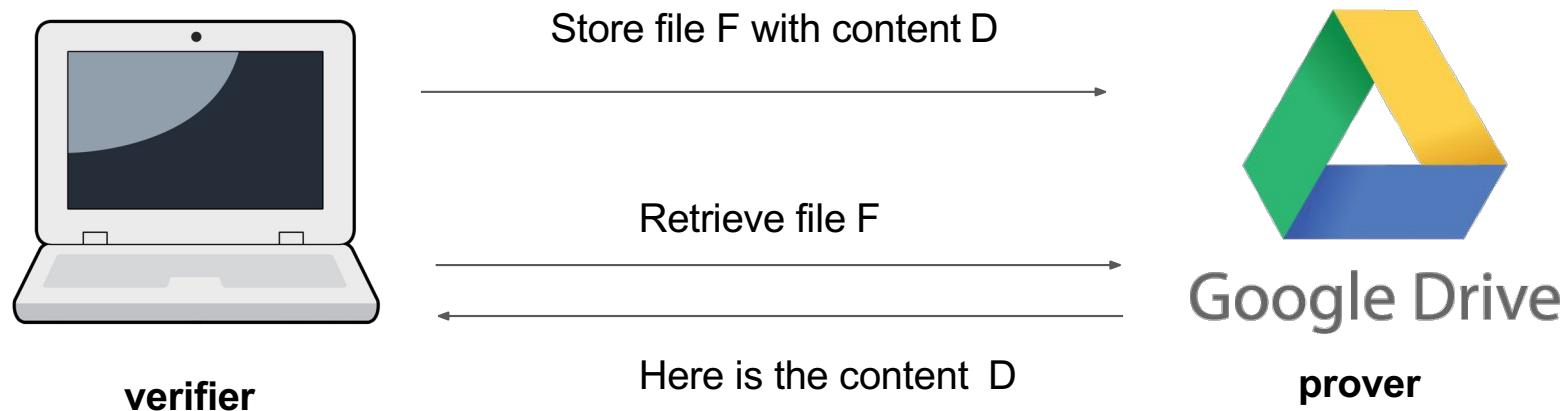
::: Catena di Blocchi

- Un blocco è collegato al precedente mediante un hash del blocco precedente.
- La modifica di un blocco della catena da parte di un nodo comporterebbe la generazione di un valore di *hash* differente
 - Gli altri nodi possono verificare l'**integrità** dei blocchi e rifiutare modifiche di transazioni già validate
 - Il contenuto del libro mastro è reso così immutabile



::: Autenticazione con Alberi di Hash (1/5)

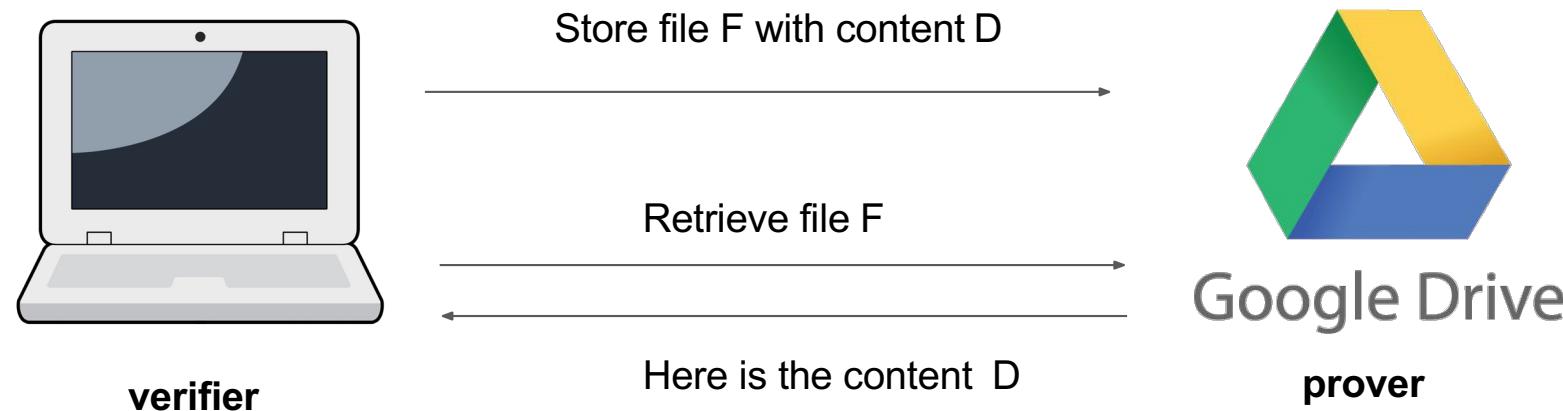
Problema della memorizzazione di file autenticati



- L'utente desidera archiviare un file su un server;
- Il file ha un nome F e un contenuto D;
- Gli utenti desiderano recuperare il file F in un secondo momento.

::: Autenticazione con Alberi di Hash (1/5)

Problema della memorizzazione di file autenticati



- L'utente invia il file F con il contenuto D al server;
- Il server archivia (F, D);
- L'utente elimina D;
- L'utente richiede F dal server;
- Il server restituisce D;
- Il cliente ha recuperato D.

Cosa succede se il server è contraddittorio e restituisce $D' \neq D$?



::: Autenticazione con Alberi di Hash (2/5)

Soluzione banale:

- L'utente non elimina D;
- Quando il server restituisce D', l'utente confronta D e D'.

... cosa succede se l'utente non dispone
di memoria sufficiente per archiviare D
per molto tempo?



::: Autenticazione con Alberi di Hash (2/5)

Soluzione banale:

- L'utente non elimina D;
- Quando il server restituisce D', l'utente confronta D e D'.

... cosa succede se l'utente non dispone
di memoria sufficiente per archiviare D
per molto tempo?



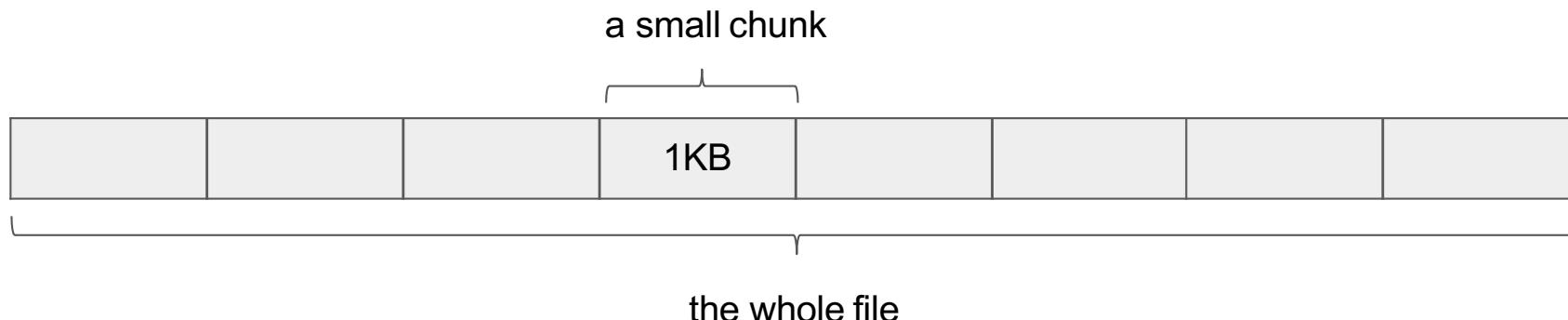
Soluzione con Hash

- L'utente invia il file F con i dati D al server;
- Il server archivia (F, D);
- L'utente memorizza H (D), elimina D;
- L'utente richiede F dal server;
- Il server restituisce D';
- L'utente confronta H (D') $\stackrel{?}{=}$ H (D).

::: Autenticazione con Alberi di Hash (3/5)

File suddiviso in chunk:

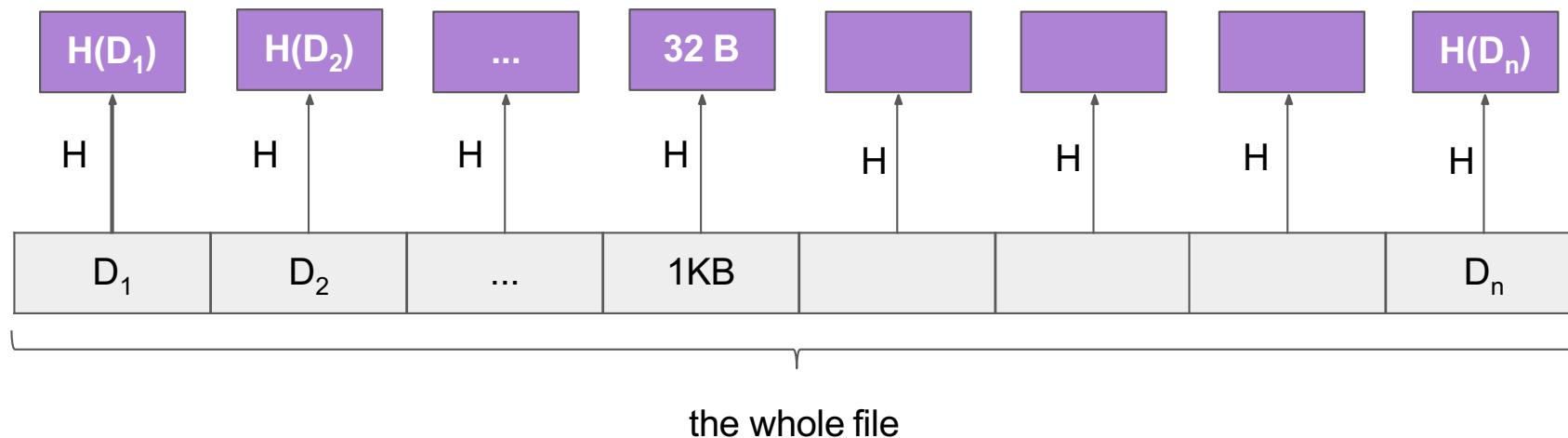
- Cosa succede se il client desidera recuperare il 200.019 byte del file?
- È necessario scaricare l'intero file ... ma per evitare ciò possiamo usare gli Alberi di Merkle!



::: Autenticazione con Alberi di Hash (3/5)

File suddiviso in chunk:

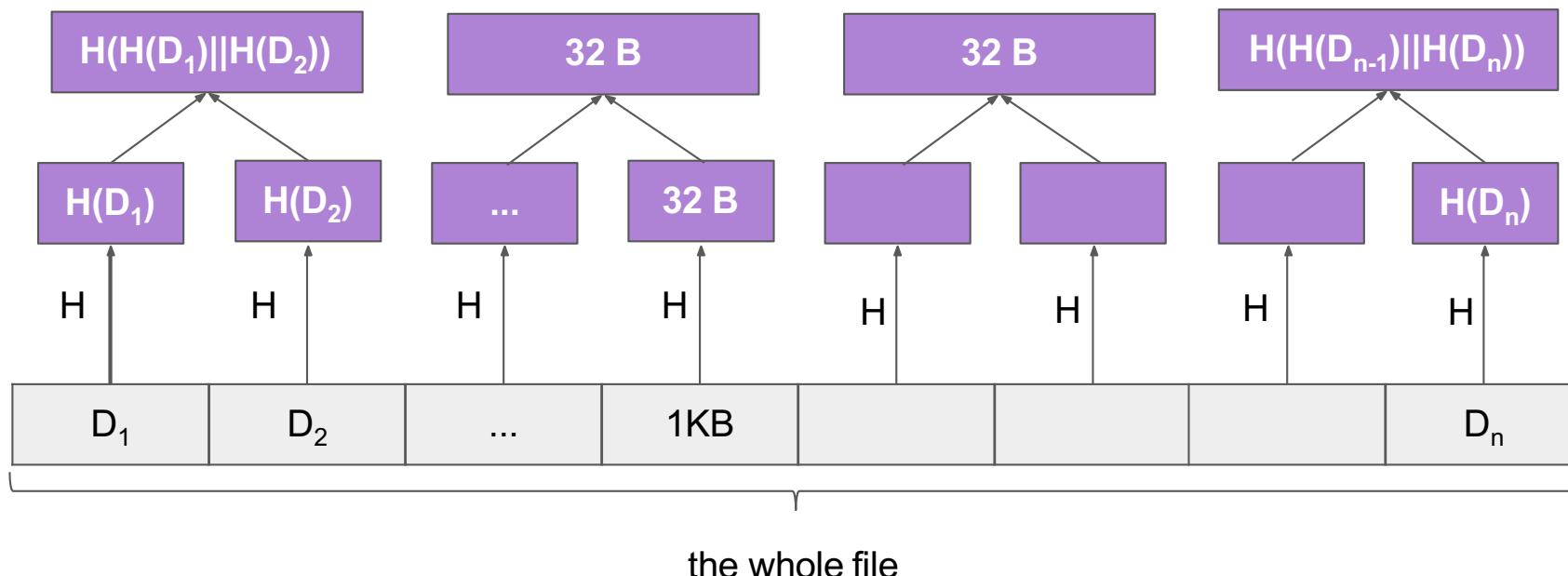
- Calcolo l'Hash di ogni blocco utilizzando una funzione hash crittografica (SHA256);
- Convenzione: le frecce mostrano la direzione dell'applicazione della funzione hash.



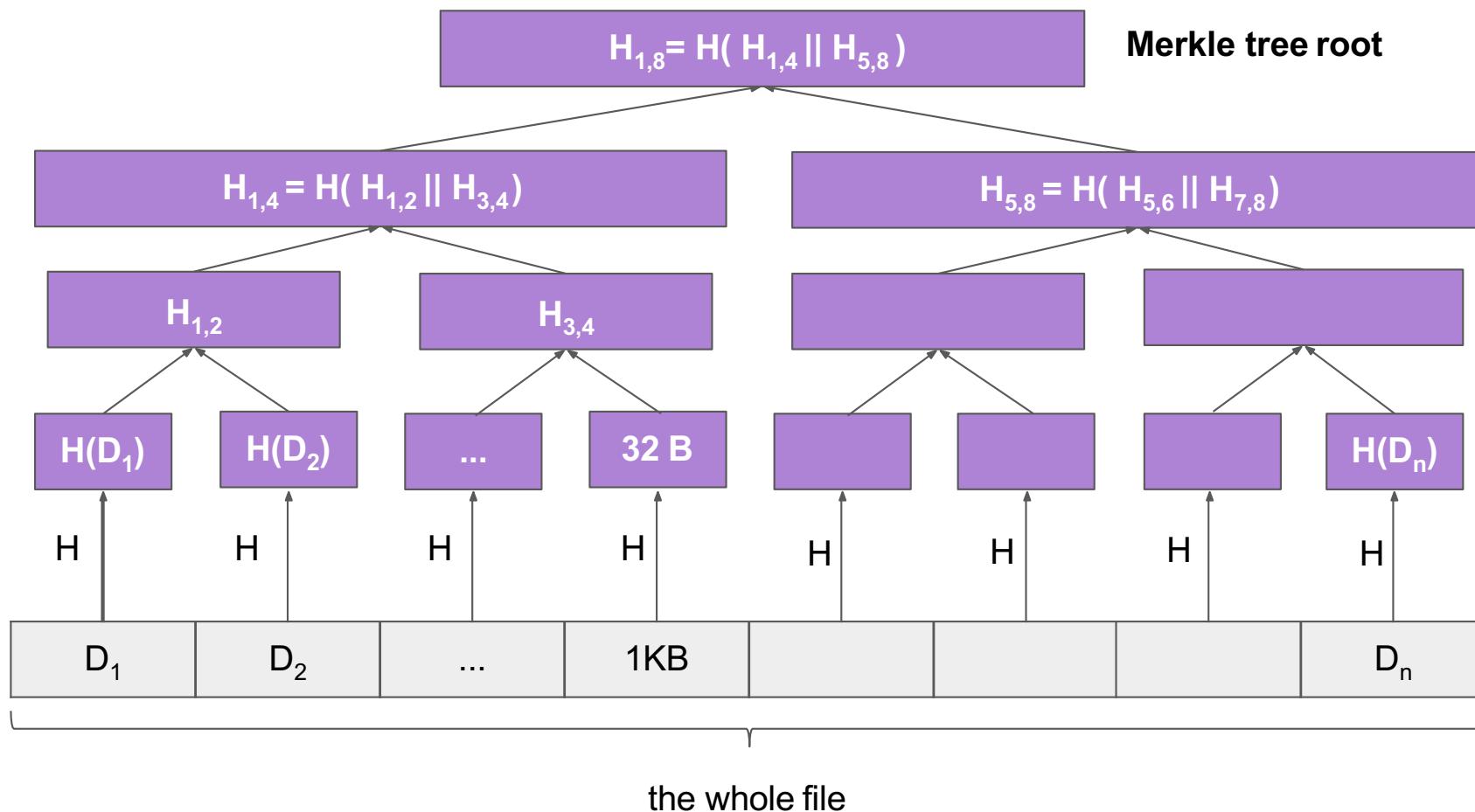
::: Autenticazione con Alberi di Hash (3/5)

File suddiviso in chunk:

- Combina gli hash a coppie per creare un albero binario;
- Ogni nodo memorizza l'hash del concat dei suoi figli.



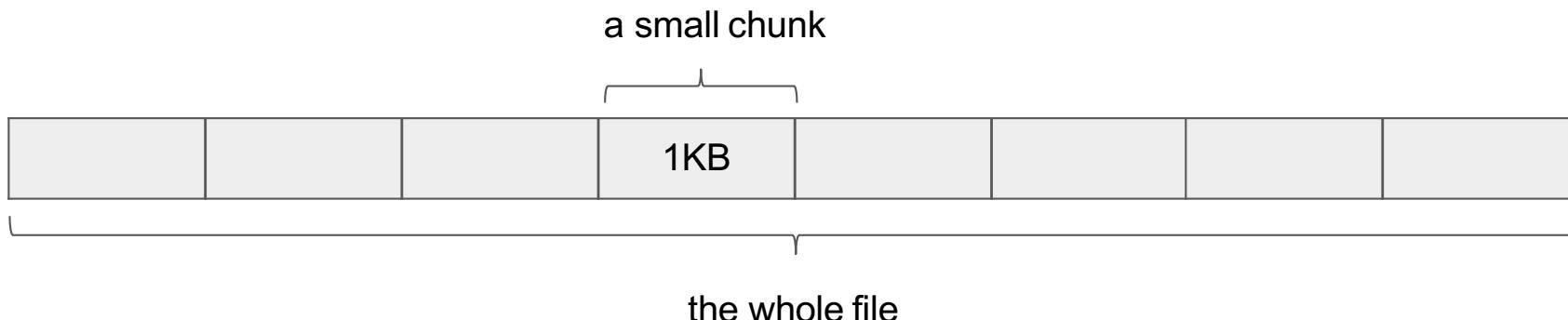
::: Autenticazione con Alberi di Hash (3/5)



::: Autenticazione con Alberi di Hash (3/5)

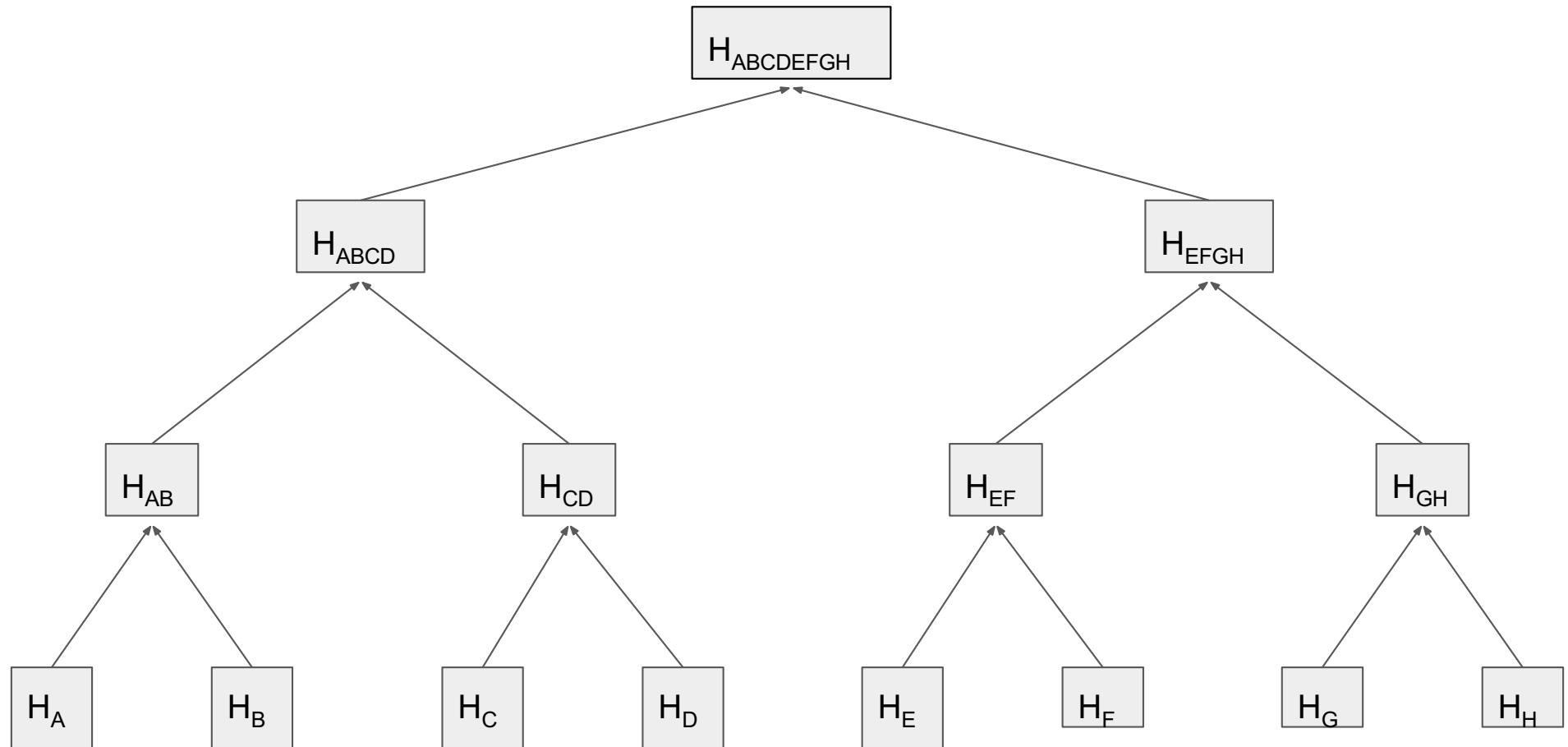
File suddiviso in chunk – Soluzione con Alberi di Merkle

- L'utente crea l'MTR radice del Merkle Tree dai dati del file iniziale D;
- L'utente invia i dati del file D al server;
- L'utente elimina i dati D, ma memorizza MTR (32 byte);
- L'utente richiede il blocco x dal server;
- L'utente restituisce il blocco x e una breve prova di inclusione π ;
- L'utente controlla che il blocco x sia incluso in MTR usando la prova π .



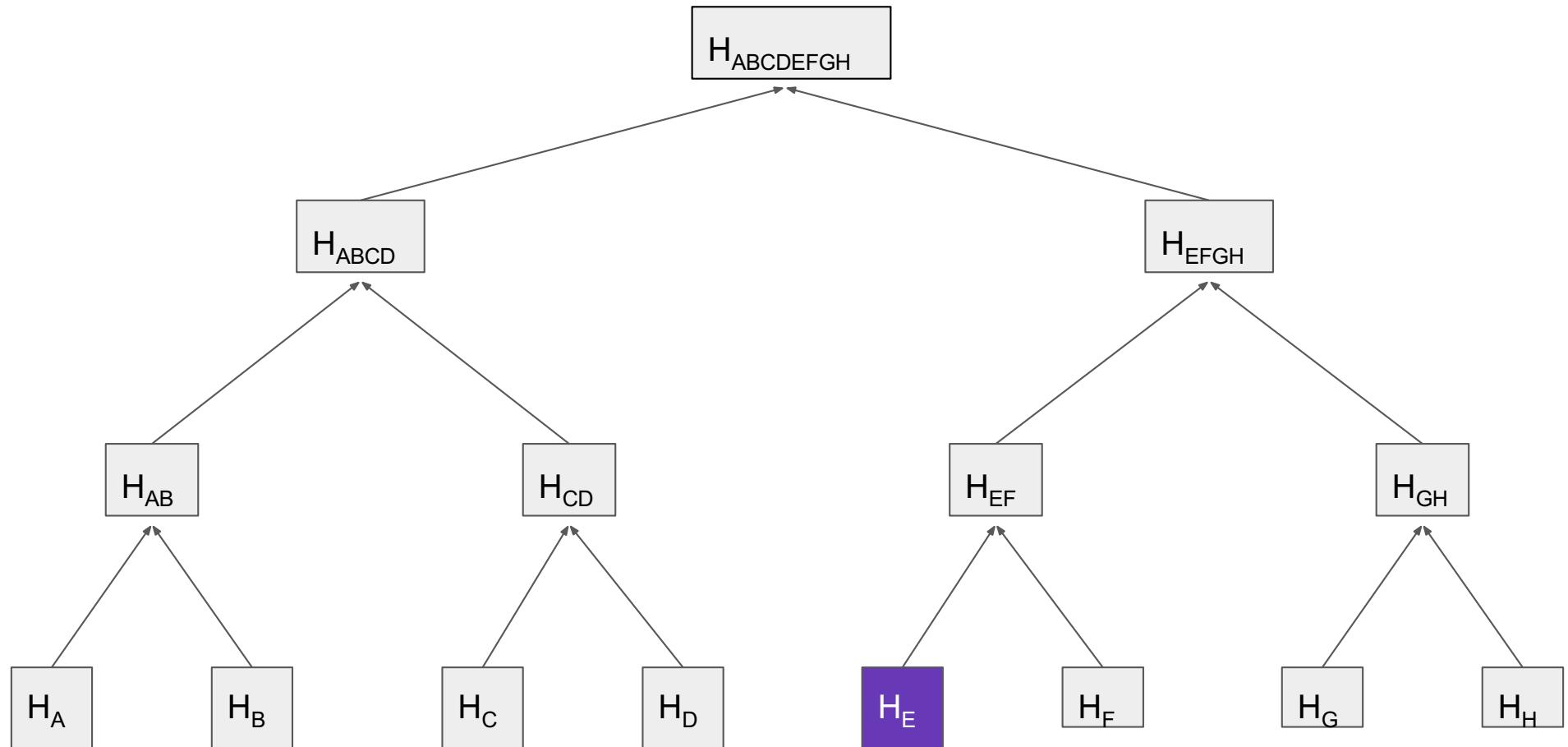
::: Autenticazione con Alberi di Hash (4/5)

Merkle tree: proof of inclusion



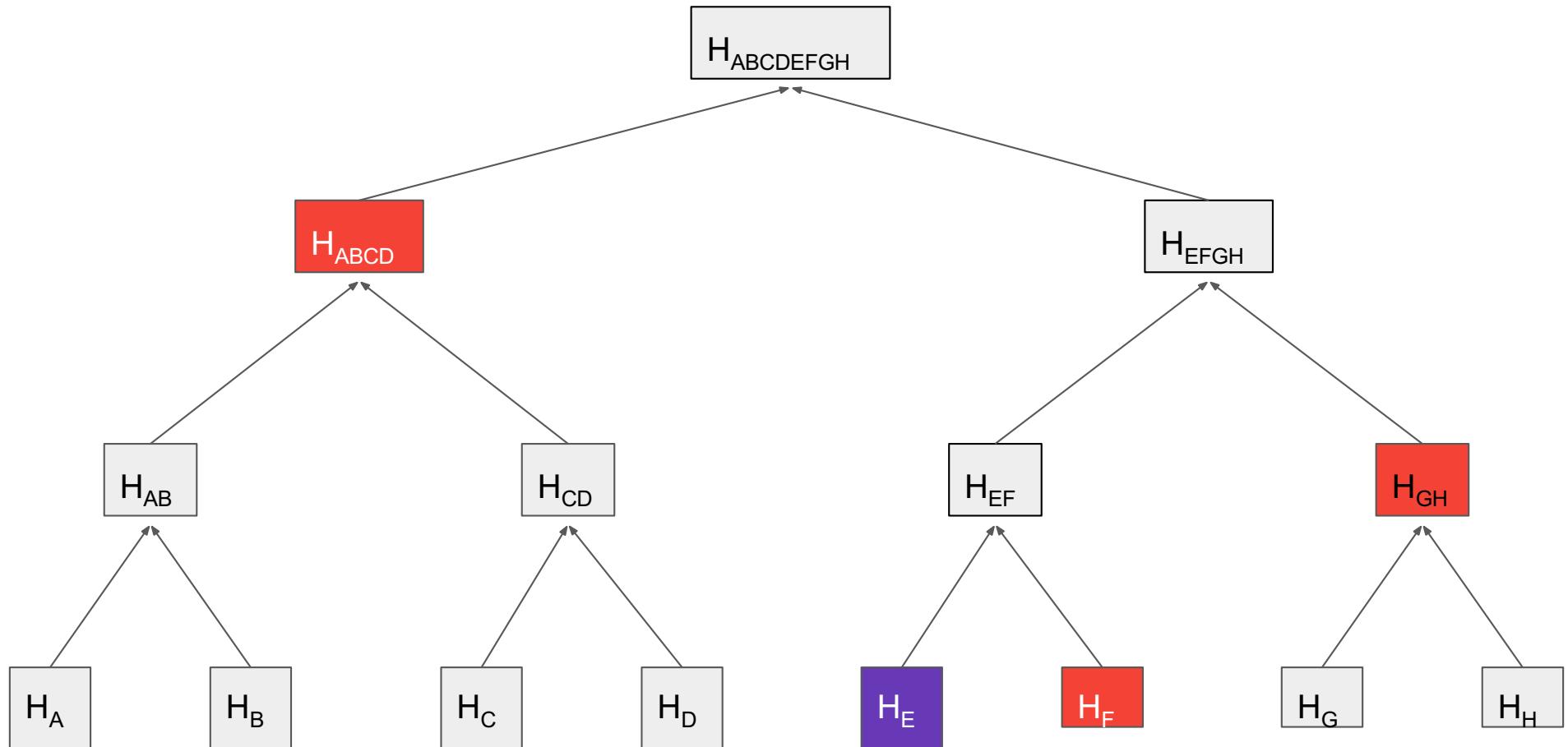
::: Autenticazione con Alberi di Hash (4/5)

Merkle tree: proof of inclusion



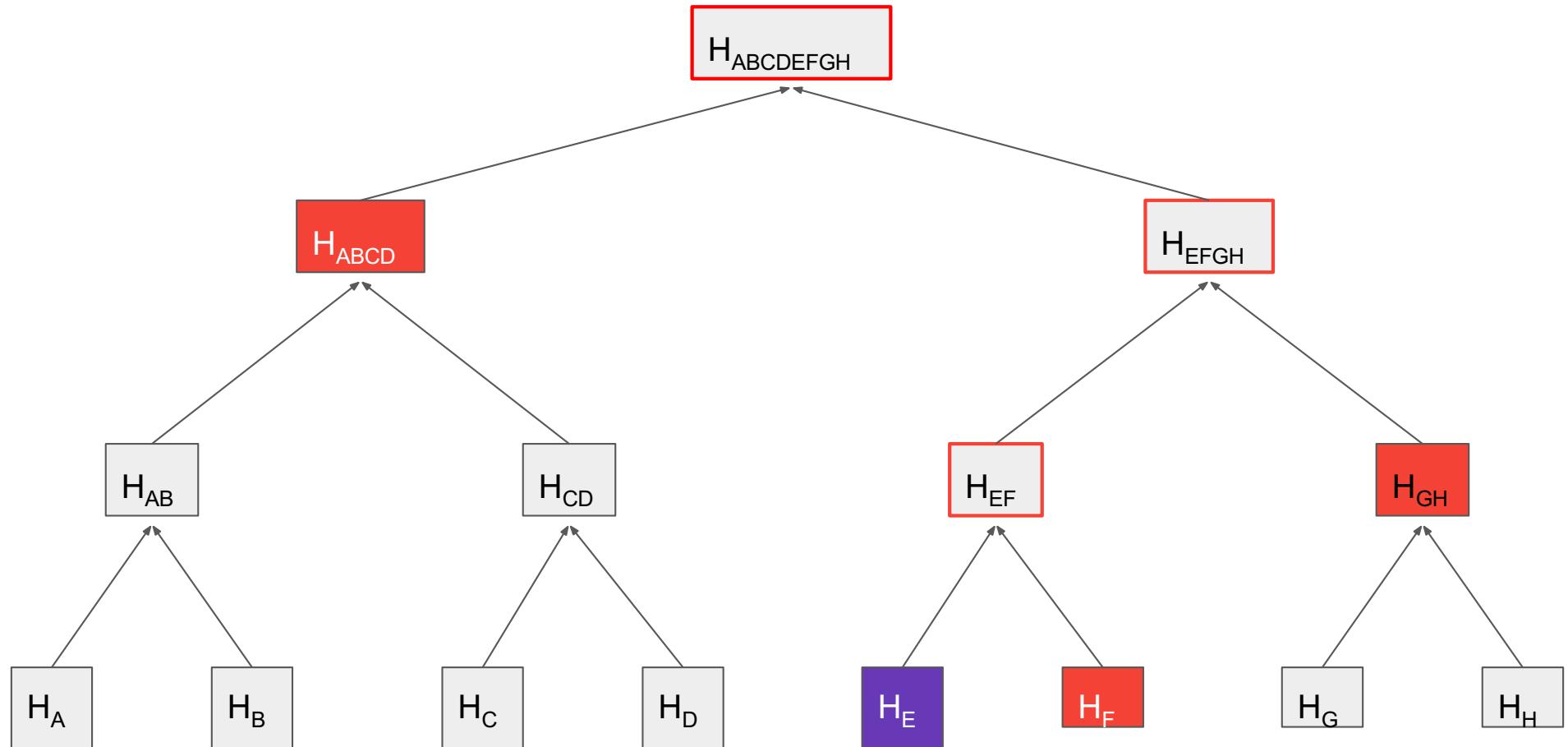
::: Autenticazione con Alberi di Hash (4/5)

Merkle tree: proof of inclusion



::: Autenticazione con Alberi di Hash (4/5)

Merkle tree: proof of inclusion



::: Autenticazione con Alberi di Hash (5/5)

File suddiviso in chunck – Soluzione con Alberi di Merkle

- Il Prover invia un chunck;
- Il Prover invia i vicini lungo il percorso che collega la foglia a MTR;
- Il Verifier calcola gli hash lungo il percorso che collega la foglia a MTR;
- Il Verifier controlla che radice calcolata corrisponde a MTR
- Quanto è grande la prova di inclusione?

$$|\pi| \in \Theta(\lg |D|)$$

Se l'avversario può presentare la prova di inclusione per un nodo foglia errato, allora possiamo compromettere la funzione hash.

::: Identità dei Nodi

- Sebbene applicate a problemi in cui si vuole fare a meno di *trusted entities*, le tecnologie blockchain utilizzano anch'esse la firma digitale (e le *trusted authorities* di certificazione) per problemi di sicurezza tipici dei sistemi distribuiti:
 - **Authentication:** autenticazione dei nodi della rete;
 - **Integrity:** verifica di violazioni dell'integrità dei messaggi scambiati;
 - **Non-repudiation:** Non ripudiabilità da parte dei nodi dei messaggi inviati.

::: Validità delle Transazioni

- Si consideri un nodo che desidera "vendere" un bene attraverso una transazione
- Il nodo predisponde da sé la transazione, che deve sottoporre agli altri nodi affinché sia approvata (validata) da una maggioranza
- È possibile che il nodo provi a “spendere” concorrentemente due volte lo stesso bene (double spending):
 - In maniera consapevole: **byzantine behaviour**
 - In maniera inconsapevole: **malfunzionamenti di rete** (duplicazione di pacchetti), **replay attack**.

::: Nonce

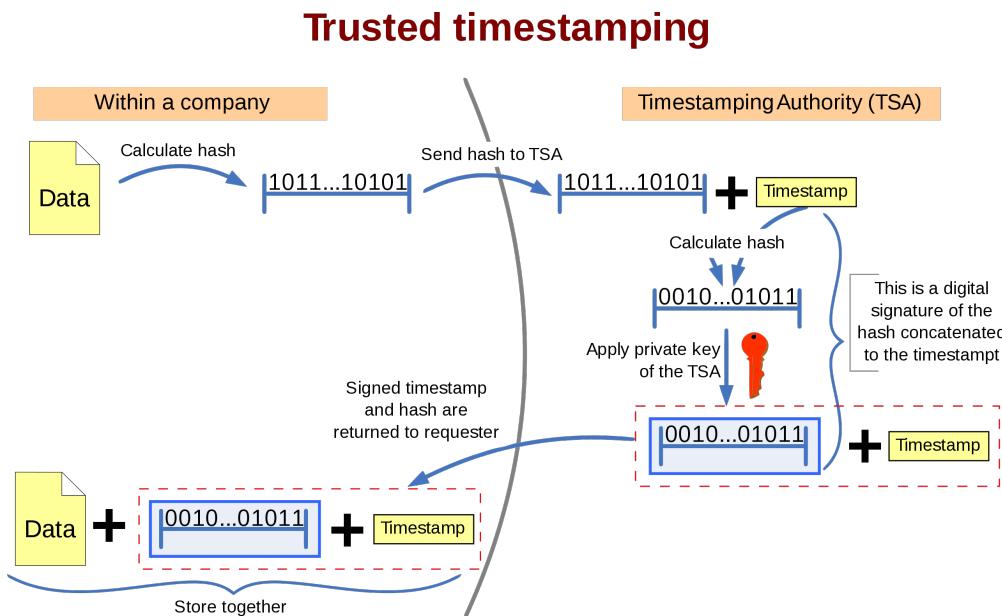
- Un primo passo per risolvere il problema del *double spending* è determinare se un messaggio è "fresco" o è una copia ritrasmessa.
- È necessario poter identificare univocamente un messaggio contenente un blocco di transazioni eseguite.
- Un ***nonce*** è definito come "*a time-varying value that has at most a negligible chance of repeating, e.g., a random value that is generated anew for each use, a timestamp, a sequence number, or some combination of these.*" [NIST SP-800-90].
- Un *nonce* può quindi essere ottenuto attraverso un "timestamp":
 - Per garantire che ogni blocco sia caratterizzato da un valore differente è necessario che tale valore sia generato da una fonte "fidata".

::: Certificazione del Tempo (1/4)

- La certificazione del tempo serve a verificare quando un documento è stato creato, oppure è stata apportata l'ultima modifica in maniera affidabile e non falsificabile.
- Si impiega un *Trusted Time Server* (TTS)
 - Prima soluzione:
 - L'utente invia al TTS una copia del documento
 - Il TTS aggiunge data e ora. Conserva il documento nella cassetta di deposito
 - Seconda soluzione: Si impiega una soluzione crittografica per proteggere la privacy del documento, e ridurre i costi di trasmissione e memorizzazione:
 - L'utente invia al TTS l'hash del documento
 - Il TTS aggiunge data, ora e firma e invia questo certificato all'utente.

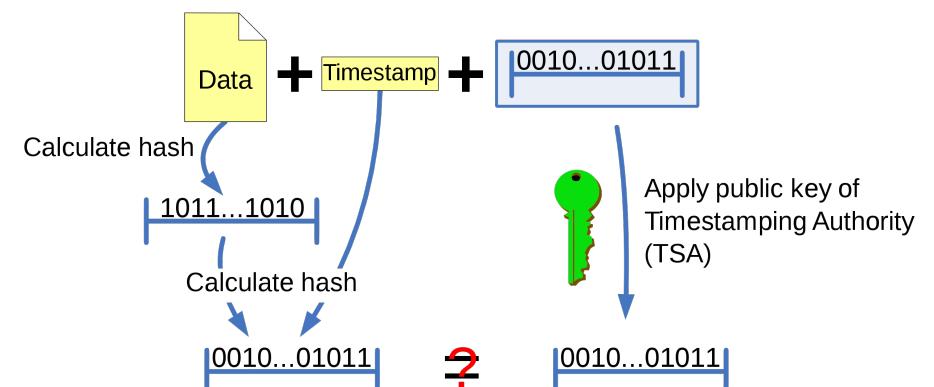
::: Certificazione del Tempo (2/4)

- Lo standard ANSI ASC X9.95 indica che un timestamp attendibile viene emesso da un'autorità di timestamp (TSA).



Un hash viene calcolato e inviato alla TSA. La TSA concatena un timestamp all'hash e calcola un hash, che viene firmato digitalmente con la chiave privata della TSA. Questo hash firmato e il timestamp vengono restituiti.

Checking the trusted timestamp



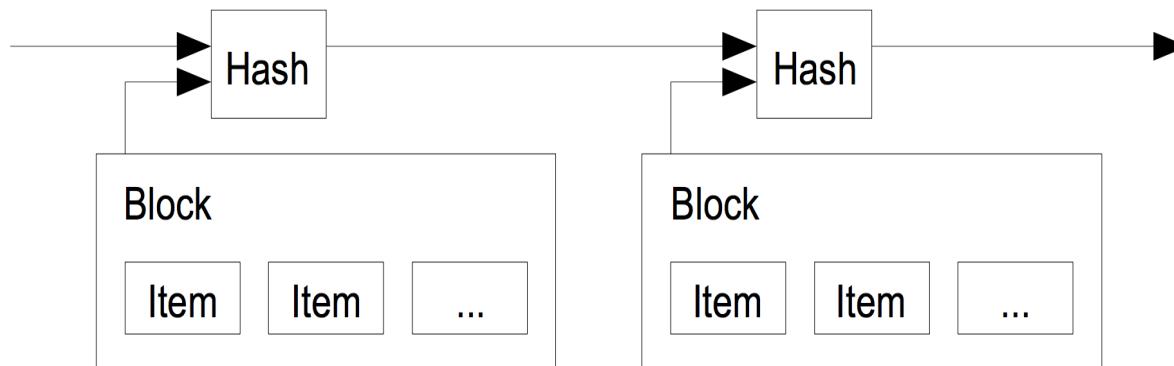
Chiunque si fidi del timestamper può quindi verificare che i dati non siano stati creati dopo la data che il timestamper garantisce.

::: Certificazione del Tempo (3/4)

- Indipendentemente della correttezza del tempo certificato, si vuole che due documenti sottoposti sequenzialmente al TTS abbiano un nonce che esprima questa sequenzialità:
 - Ogni certificato rilasciato contiene l'hash del documento, data, ora, firma del TSS e l'hash del certificato precedente (o di parte di esso)
 - Complessivamente i certificati formano una catena di blocchi
 - È inammissibile inserire successivamente un certificato all'interno della catena, il TTS dovrebbe rilevare collisioni per la funzione hash

::: Certificazione del Tempo (4/4)

- Questa soluzione comporta una serie di vantaggi:
 - Aggiungere il timestamp rende l'hash più resistente rispetto alle collisioni;
 - Risolve il problema del double spending assegnando un ordine temporale alle transazioni e pubblicandole.



::: Proof-of-Work (1/3)

- Per svincolarsi dall'utilizzo di un *timestamp server*, alcune soluzioni blockchain sfruttano un meccanismo differente per la generazione dei **nonce**.
- Tale meccanismo è definito **Proof-of-Work (PoW)**.

::: Proof-of-Work (2/3)

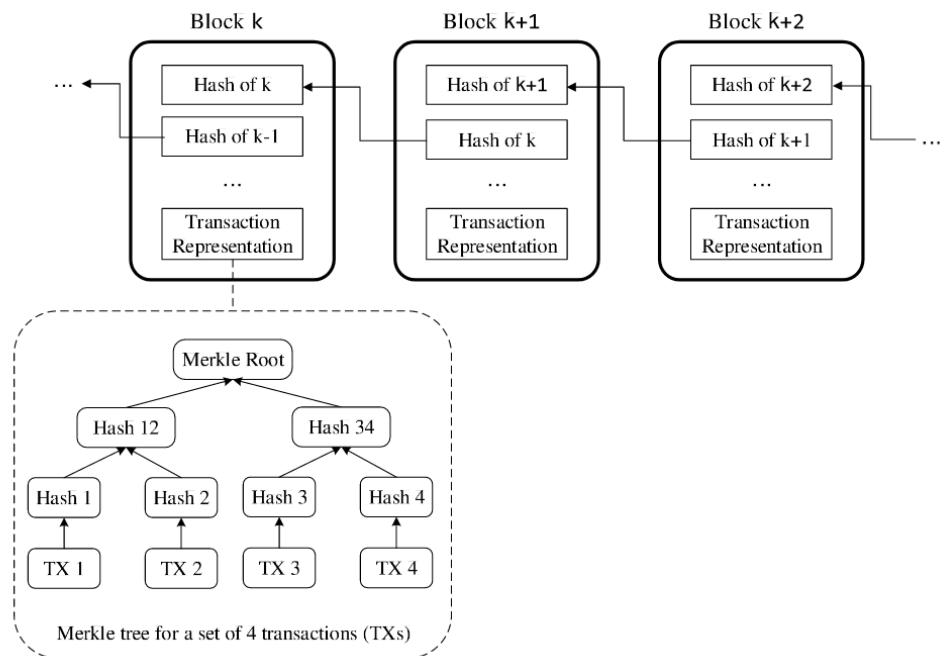
- **Proof-of-Work:** per proporre un nuovo blocco da aggiungere al *ledger*, un nodo deve dimostrare di aver utilizzato abbastanza risorse di calcolo per risolvere un *enigma matematico*, il cui risultato è utilizzato inserito nel blocco stesso.
- La Proof-of-Work implica **la computazione di un valore di hash con un determinato numero di bit iniziali pari a zero**.

$$H(\text{ctr} \parallel x \parallel s) \leq T$$

- Ogni nodo (**miner**) cerca di trovare un **nonce** tale da soddisfare il vincolo sul numero di zero prodotti dalla funzione di hash.

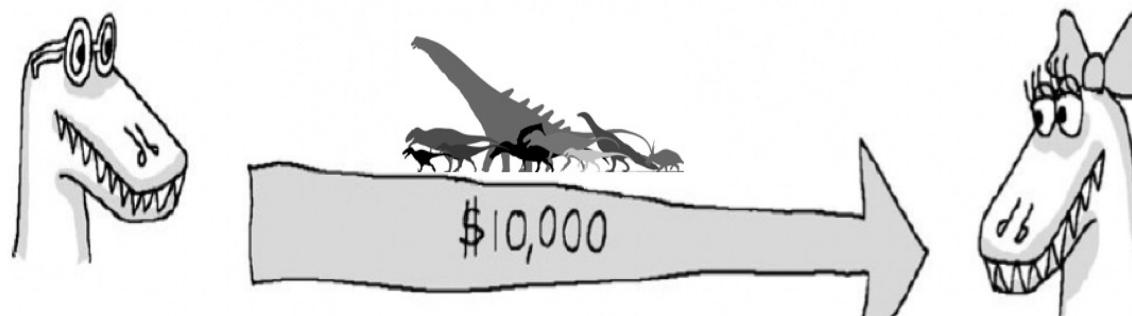
::: Proof-of-Work (3/3)

- Il lavoro medio richiesto è esponenziale rispetto al numero di bit zero richiesti e può essere verificato eseguendo un singolo hash.
- Tale operazione è molto onerosa dal punto di vista computazionale e richiede la collaborazione di diverse CPU per poter essere realizzata.
- L'attacco del 51% diventa molto oneroso



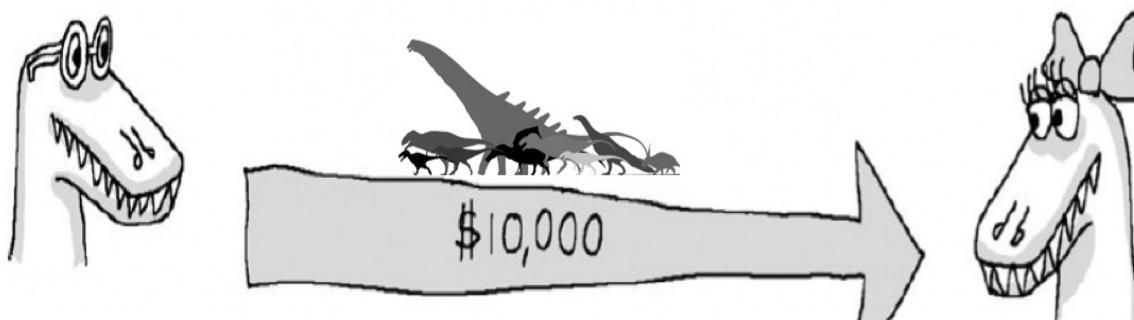
::: Transazioni con Consenso Distribuito (1/3)

- Tutti i nodi della rete sono a conoscenza di tutte le transazioni e partecipano attivamente alla validazione dei blocchi attraverso il raggiungimento del consenso
 - Il modello di sicurezza deve contemplare anche il cosiddetto "51% Attack": un gruppo di nodi maliziosi possiede più del 50% della potenza computazionale della rete.



::: Transazioni con Consenso Distribuito (2/3)

- "Nessun algoritmo può garantire il raggiungimento del consenso in un sistema asincrono nel caso di anche un unico fallimento per crash di un processo" (Teorema di Impossibilità).
- Per raggiungere il consenso in sistemi asincroni in presenza di fallimenti si può:
 - Rilassare i vincoli di consenso;
 - Rendere meno "asincrono" il sistema
 - Sfruttare i periodi di sincronia.



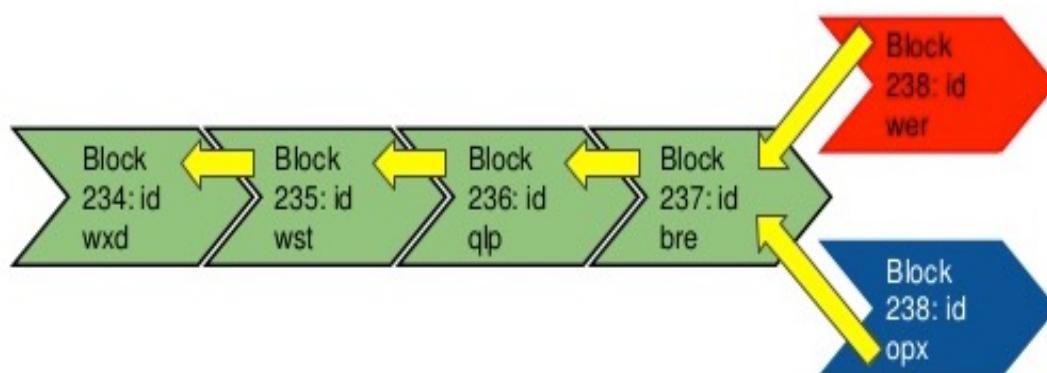
::: Transazioni con Consenso Distribuito (3/3)

Rilassare i Vincoli del Consenso

- È consentito che in alcuni casi non si raggiunga il consenso tra tutti i nodi della rete, ovvero solo un sottoinsieme di nodi raggiungono il consenso.
- "51% Attack" possibile.
- Esistono molte soluzioni diverse che consentono di irrobustire il consenso per evitare questo tipo di attacchi.
 - Aggiunta di un “controllo collettivo”: Chiunque valida una transazione deve spendere tempo computazionale (Quanto? Tale da scoraggiare l’attacco)
 - È previsto un meccanismo di incentivazione: una ricompensa per il lavoro svolto.

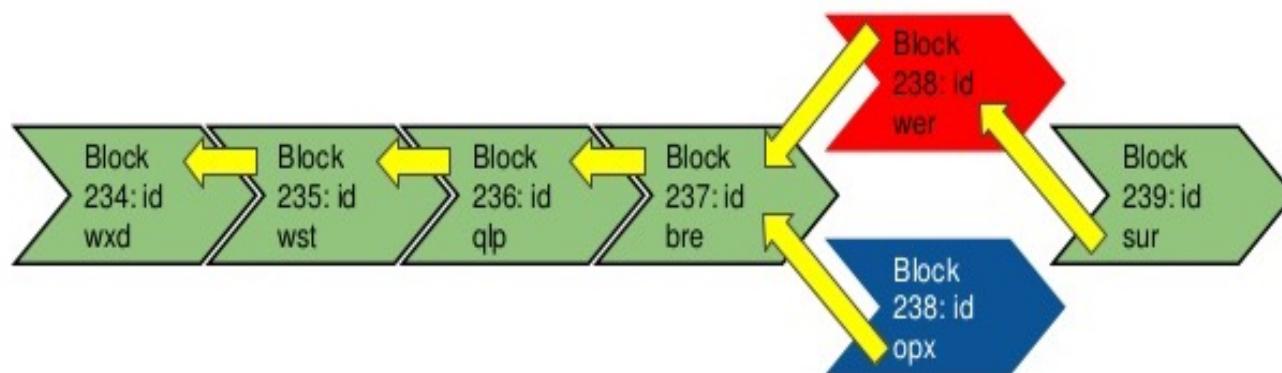
::: Blockchain Forking (1/3)

- Il rilassamento dei vincoli di consenso può dare origine a biforazioni della blockchain.
- Nel caso di produzione "simutanea" di due blocchi alcuni blocchi aggiungeranno alla propria blockchain il *blocco rosso* ed altri il *blocco blu*, fino ad arrivare alla realizzazione di una "forked blockchain".



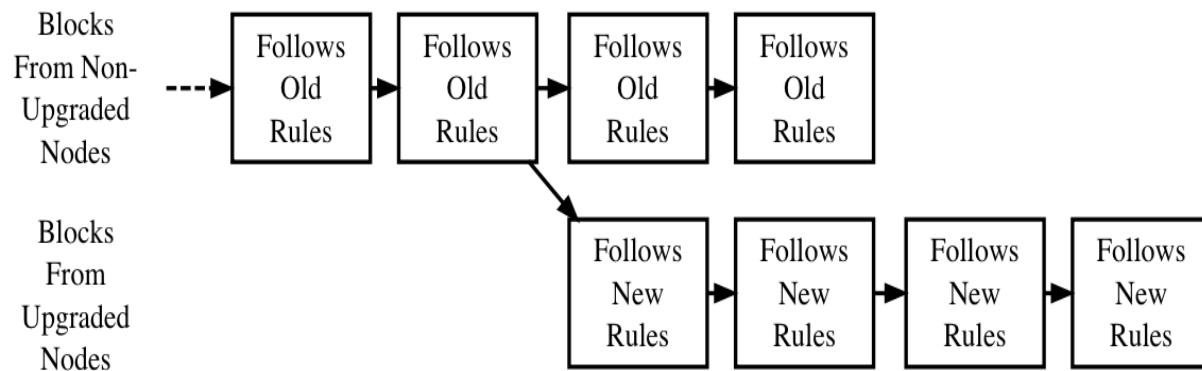
::: Blockchain Forking (2/3)

- All'atto della produzione del blocco successivo, esso farà riferimento ad uno solo dei due blocchi precedentemente creati.
- Tale operazione comporta la rimozione del ramo della blockchain composto dal nodo blu.



::: Blockchain Forking (3/3)

- In un certo istante di tempo la comunità può ritenere necessario cambiare le regole della blockchain in modo che determinate transazioni ritenute "non valide" fino a tale istante siano considerate "valide" a partire da tale istante e nel futuro
- Tale condizione genera un **hard fork**, che comporta la generazione "voluta" di un nuovo ramo della blockchain per distinguere i nodi che seguono il nuovo regolamento rispetto quelli fedeli al vecchio
- In qualsiasi momento i nodi possono decidere di abbandonare un ramo per seguire l'altro



::: Consenso Nakamoto-BitCoin (1/4)

1. Ogni nuova transazione viene inviata in broadcast a tutti i nodi
2. Ogni nodo colleziona le nuove transazioni in un blocco
3. Ogni nodo cerca una PoW "difficile" per il proprio blocco

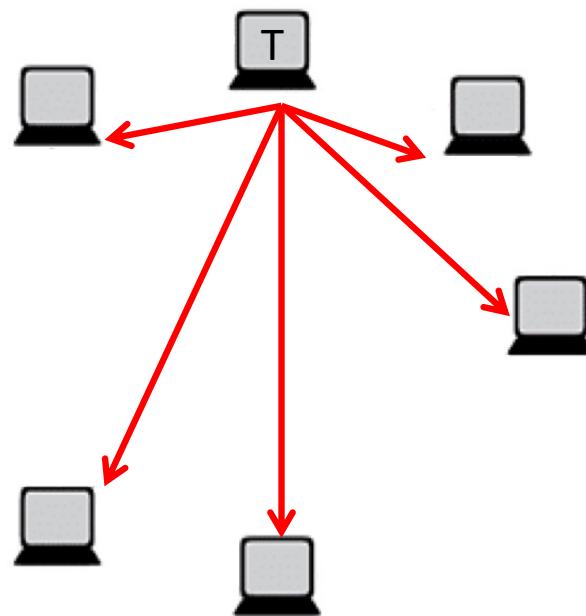
La PoW è il calcolo di un valore (anch'esso un *nonce*) tale che un *hash* di blocco+*nonce* inizi con uno stabilito numero di zeri
4. Quando un nodo trova una PoW, invia in broadcast a tutti i nodi il blocco, contenente le transazioni e la PoW
5. Un nodo accetta il blocco solo se:
 - Tutte le transazioni sono valide
 - Le transazioni non sono relative ad un bene già speso
 - La PoW è valida. A tale scopo il nodo calcola a sua volta l'*hash* del blocco, e verifica che abbia lo stesso numero di zeri iniziali
6. I nodi manifestano l'accettazione del blocco aggiungendolo alla *blockchain* all'atto della creazione del blocco successivo, utilizzando l'*hash* del blocco accettato per creare il prossimo blocco

::: Consenso Nakamoto-BitCoin (2/4)

- Due nodi possono fare concorrentemente il broadcast di blocchi differenti (passo 4)
- Nodi diversi possono ricevere i due broadcast in ordine diverso
- Ciascun nodo lavora sul primo blocco che riceve, ma conserva l'altro ramo
- L'indecisione si risolve quando un ramo diventa più lungo
 - I nodi che lavorano sul ramo meno lungo lo abbandonano e proseguono a lavorare sulla catena più lungo

::: Consenso Nakamoto-BitCoin (3/4)

1. Un nodo genera una nuova transazione T e ne fa il broadcast a tutti i nodi



Blockchain corrente

::: Consenso Nakamoto-BitCoin (3/4)

1. Un nodo genera una nuova transazione T e ne fa il broadcast a tutti i nodi



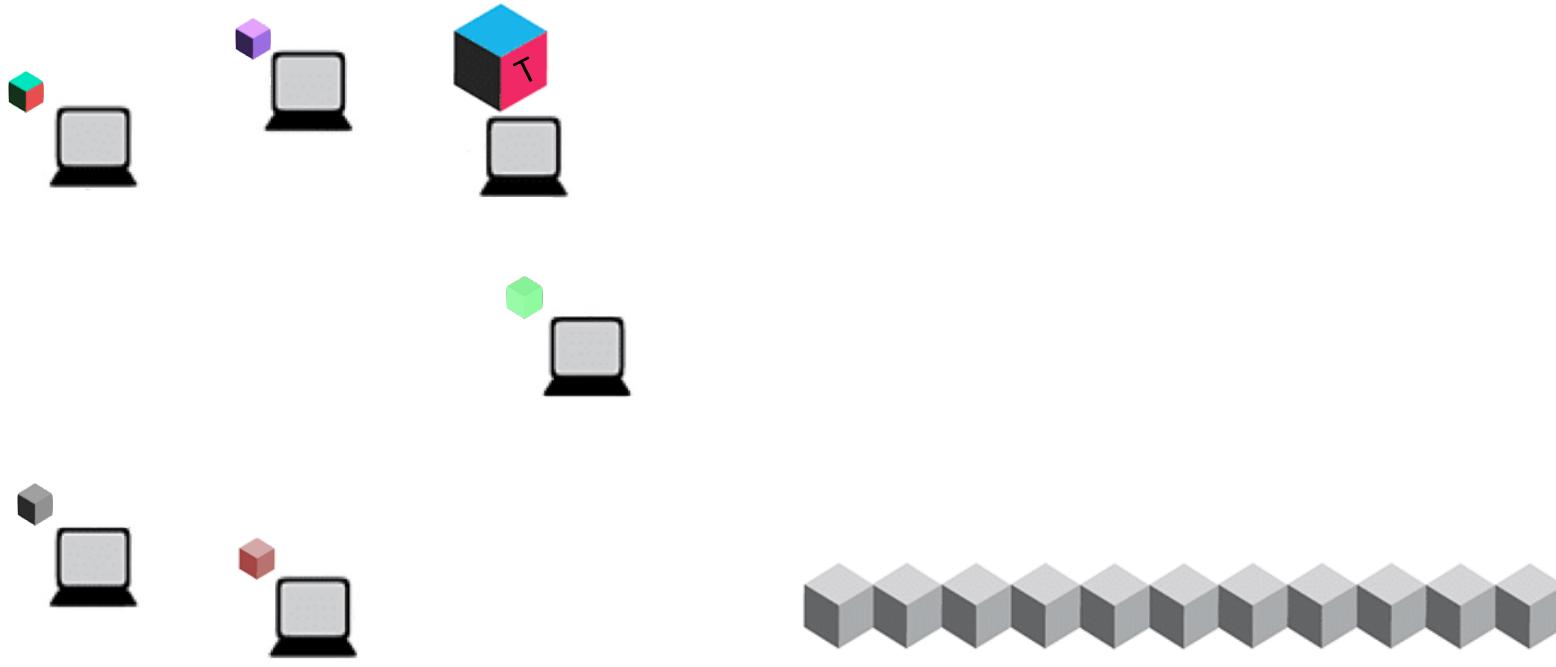
Gli altri nodi ricevono T



Blockchain corrente

::: Consenso Nakamoto-BitCoin (3/4)

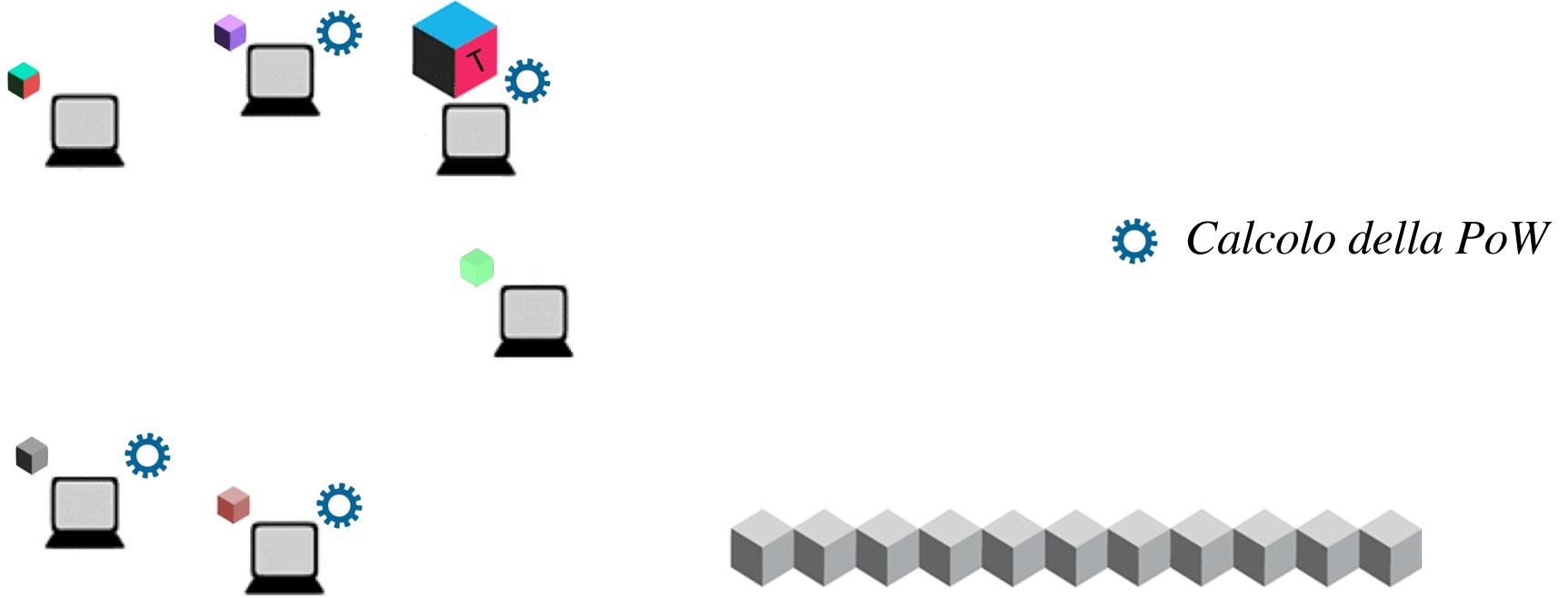
2. Ogni nodo colleziona le nuove transazioni in un proprio blocco



Blockchain corrente

::: Consenso Nakamoto-BitCoin (3/4)

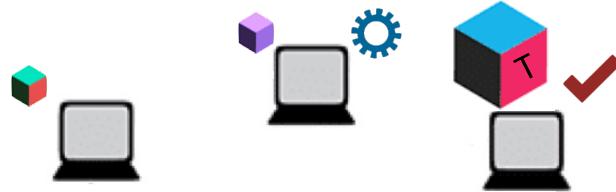
3. I nodi cercano una Proof-of-Work per il proprio blocco



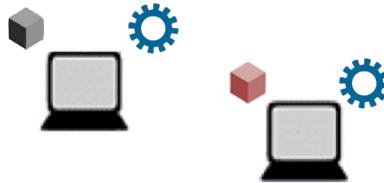
N.B.: I blocchi su cui nodi diversi calcolano la PoW sono in generale diversi

::: Consenso Nakamoto-BitCoin (3/4)

3. I nodi cercano una Proof-of-Work per il proprio blocco

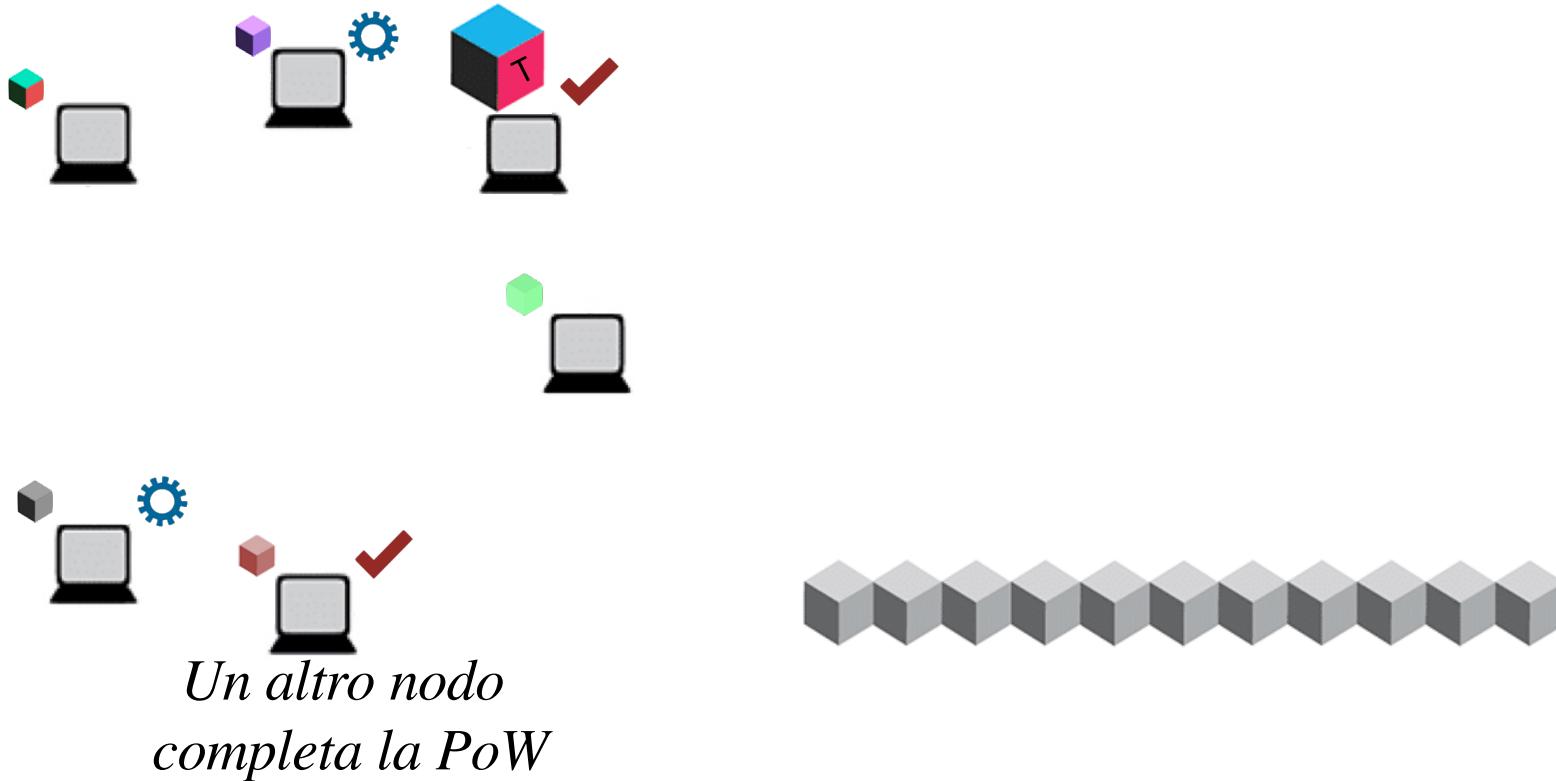


Un nodo completa la PoW



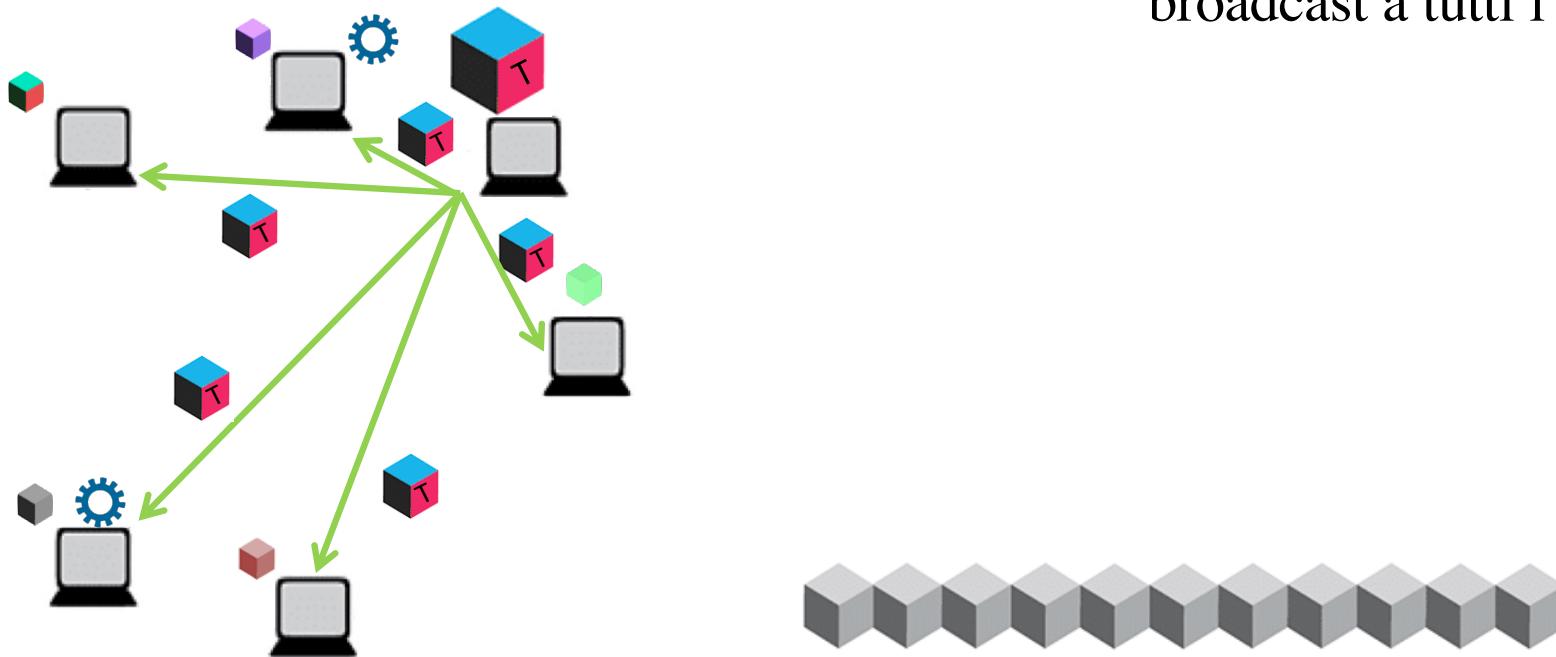
::: Consenso Nakamoto-BitCoin (3/4)

3. I nodi cercano una Proof-of-Work per il proprio blocco



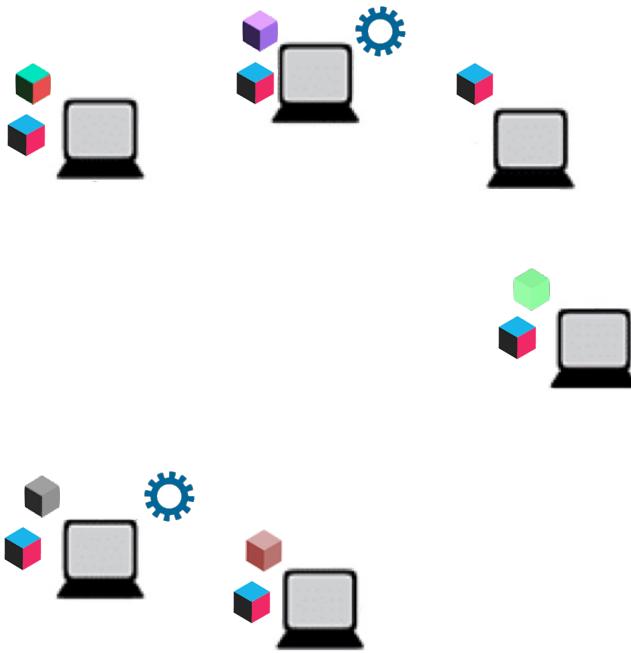
::: Consenso Nakamoto-BitCoin (3/4)

4. Quando un nodo trova una Proof-of-Work, invia il blocco in broadcast a tutti i nodi



::: Consenso Nakamoto-BitCoin (3/4)

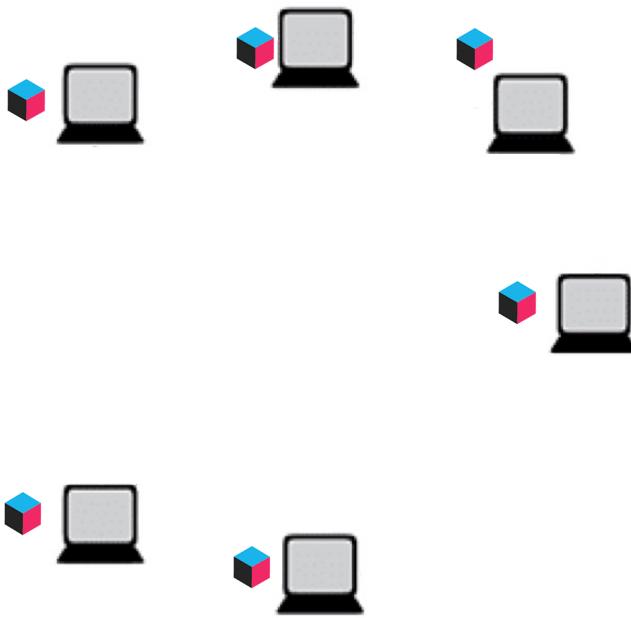
4. Quando un nodo trova una Proof-of-Work, invia il blocco in broadcast a tutti i nodi



I nodi ricevono il blocco



::: Consenso Nakamoto-BitCoin (3/4)



5. Un nodo accetta il blocco solo se:

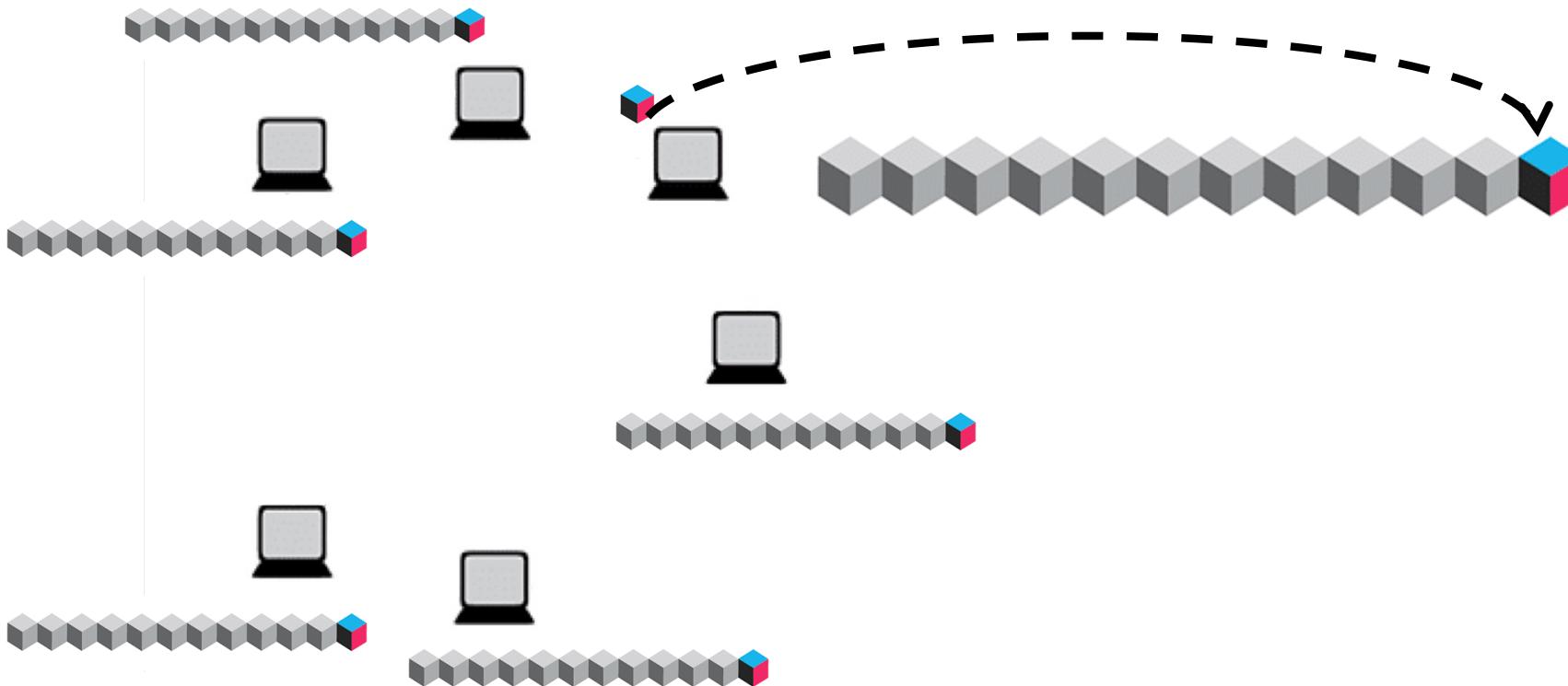
- Tutte le transazioni sono valide
- Le transazioni non sono relative ad un bene già speso
- La PoW è valida. A tale scopo il nodo calcola a sua volta l'*hash* del blocco, e verifica che abbia lo stesso numero di zeri iniziali



*Tutti i nodi accettano il blocco
(la blockchain è ancora inalterata)*

::: Consenso Nakamoto-BitCoin (3/4)

6. I nodi manifestano l'accettazione del blocco aggiungendolo alla (copia locale della) *blockchain* all'atto della creazione del blocco successivo, utilizzando l'*hash* del blocco accettato per creare il prossimo blocco



::: Consenso Nakamoto-BitCoin (4/4)

- È opportuno incoraggiare la cooperazione dei nodi alla creazione del consenso (validazione delle transazioni “corrette”)
 - affinché un blocco sia validato ci deve essere una maggioranza di nodi corretti e che spendono risorse (costruendo la PoW);
 - poiché la PoW è onerosa anche i nodi corretti potrebbero non essere ben disposti a cooperare
- È prevista una politica di incentivi
 - incoraggiare processi corretti a collaborare nonostante l'onere computazionale, mettendo in minoranza i processi maliziosi
 - i nodi maliziosi dovranno avere ancora più potenza di calcolo per “contrastare” molti nodi corretti che validano i blocchi
- La prima transazione di ogni blocco assegna un compenso al nodo creatore del blocco.

::: Blockchain (1/6)

- Una blockchain è un libro mastro pubblico distribuito (*distributed public ledger*) di transazioni o eventi digitali eseguiti e condivisi tra i partecipanti
- Ogni transazione riportata nella *blockchain* è validata tramite il raggiungimento del consenso tra i nodi del sistema
- Una volta memorizzate, le informazioni non possono essere cancellate né modificate.
- Ogni blocco contiene uno o più record (un record per transazione).
- Come in un libro mastro, la storia delle registrazioni è immutabile e può essere verificata a partire dal primo blocco (*genesis block*)

::: Blockchain (2/6)

- Le proprietà che una Blockchain fornisce sono:
 - Pubblica Verificabilità: ogni transazione può essere verificata da ogni partecipante
 - Trasparenza: ogni partecipante ha accesso ad un sottoinsieme di informazioni
 - Privacy: l'identità di chi esegue una determinata transazione deve essere tutelata
 - Integrità: le informazioni non vengono modificate da fonti non autorizzate
 - Ridondanza: dati ripetuti per ogni partecipante del sistema
 - Assenza di una "Trust Anchor"

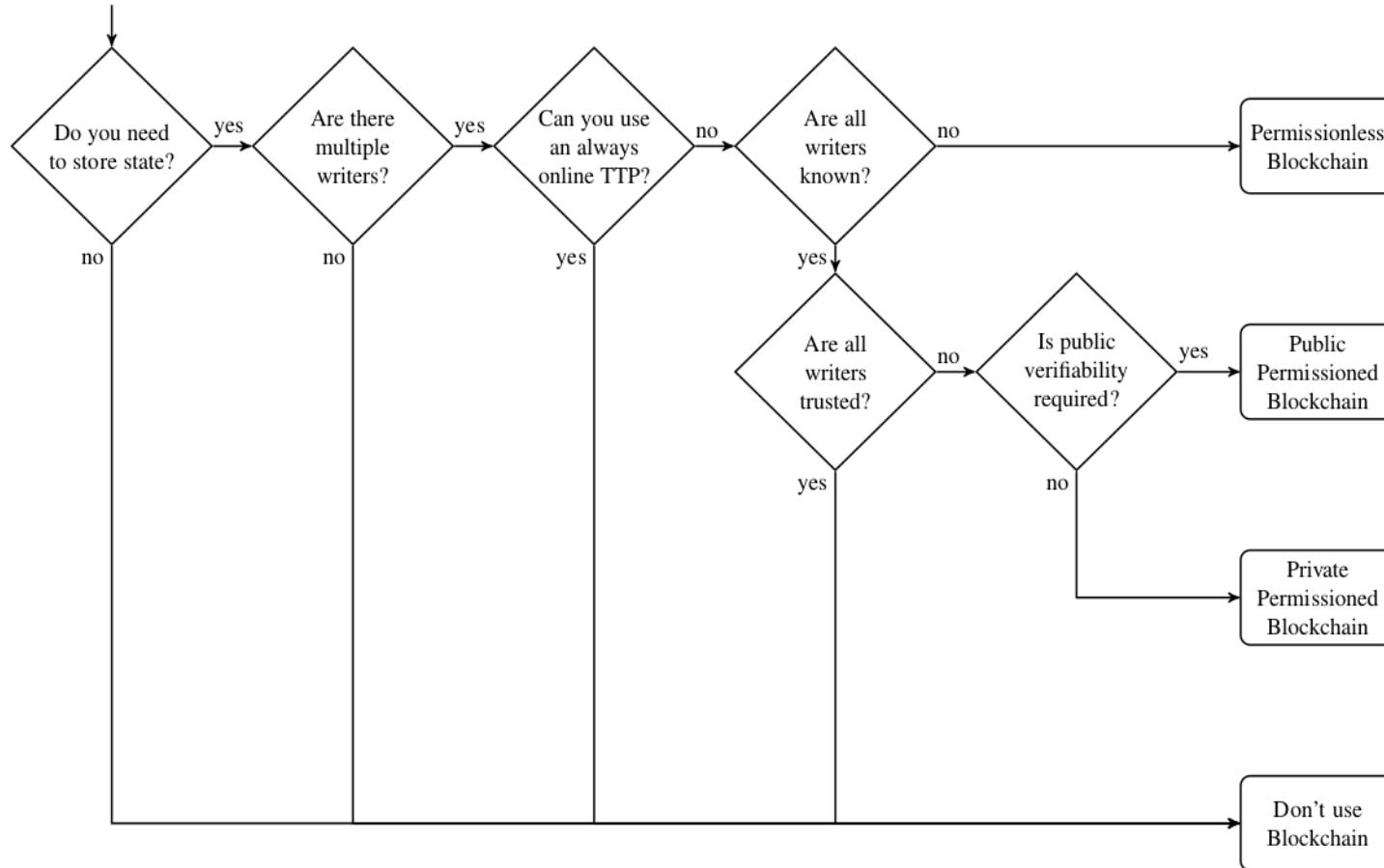
::: Blockchain (3/6)

- L'effettiva implementazione di questa tecnologia è fortemente dipendente dal tipo di Blockchain che si intende realizzare:
 - **Permissionless**: i partecipanti non devono essere preventivamente “autorizzati” per il ruolo che intendono svolgere
 - **Permissioned**: le operazioni (tutte o anche solo alcune) possono essere svolte solo da nodi preventivamente autorizzati

::: Blockchain (4/6)

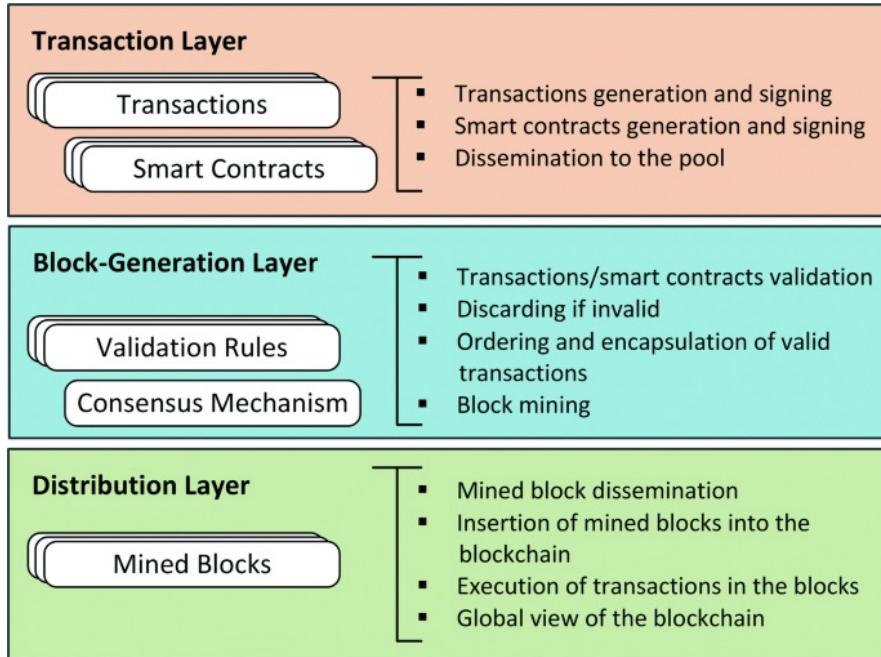
- **Sono classificabili in base all'ambito:**
 - **Public:** tutti i blocchi sono visibili a **tutti** i nodi e ogni nodo può partecipare al consenso
 - **Private:** **una specifica organizzazione** decide quali nodi possono leggere i blocchi e partecipare al consenso (throughput delle transazioni molto alto)
 - **Consortium:** **pochi nodi predeterminati** possono leggere i blocchi e partecipare al consenso

::: Blockchain (5/6)

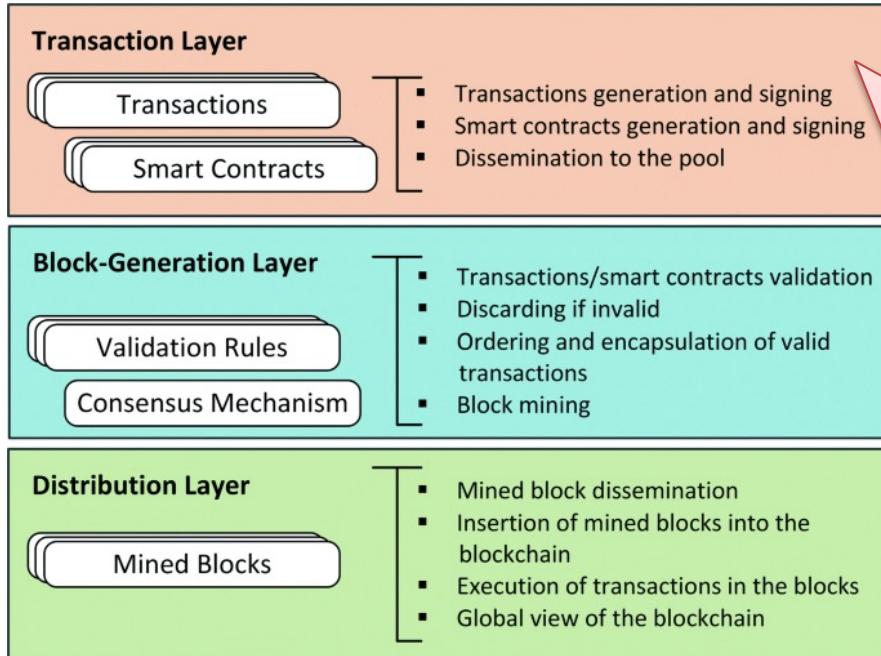


*TTP: Trusted Third Party

::: Blockchain (6/6)

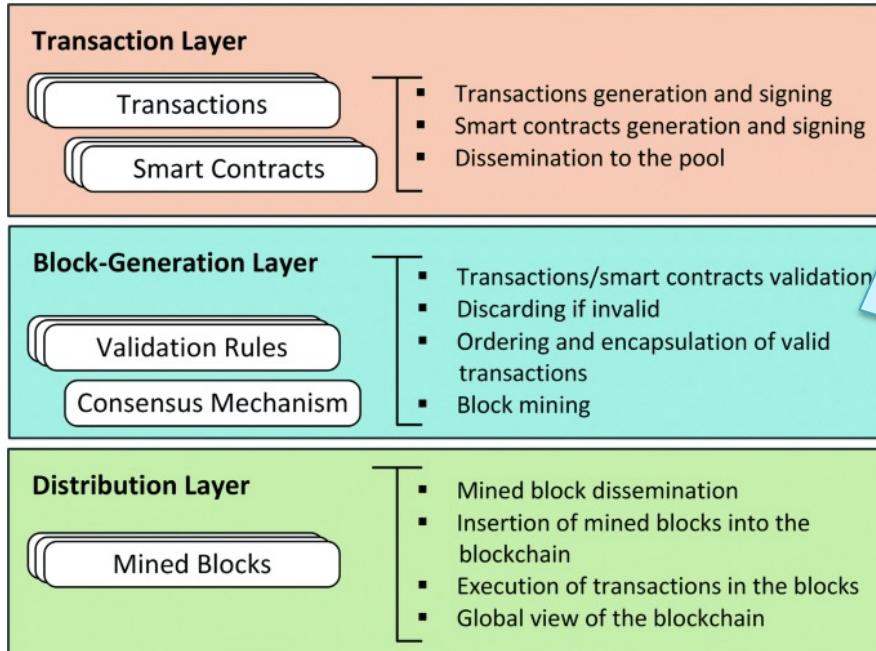


::: Blockchain (6/6)



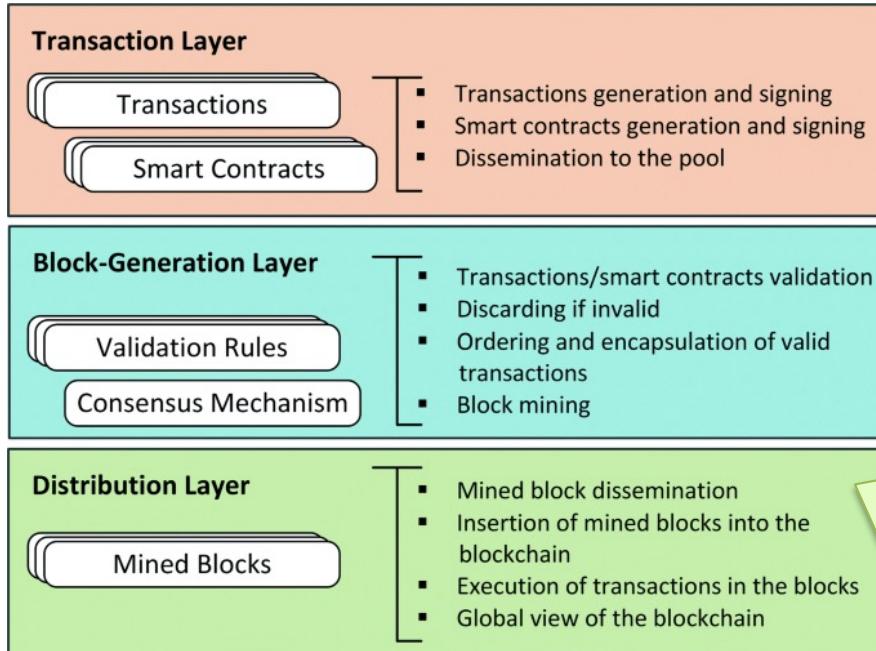
Il Transaction Layer definisce il linguaggio di codifica e i criteri utilizzati durante la generazione di transazioni e smart contract. Entrambi devono essere firmati prima della loro diffusione per garantire il non ripudio e consentire il controllo dell'accesso e l'autenticazione del loro contenuto.

::: Blockchain (6/6)



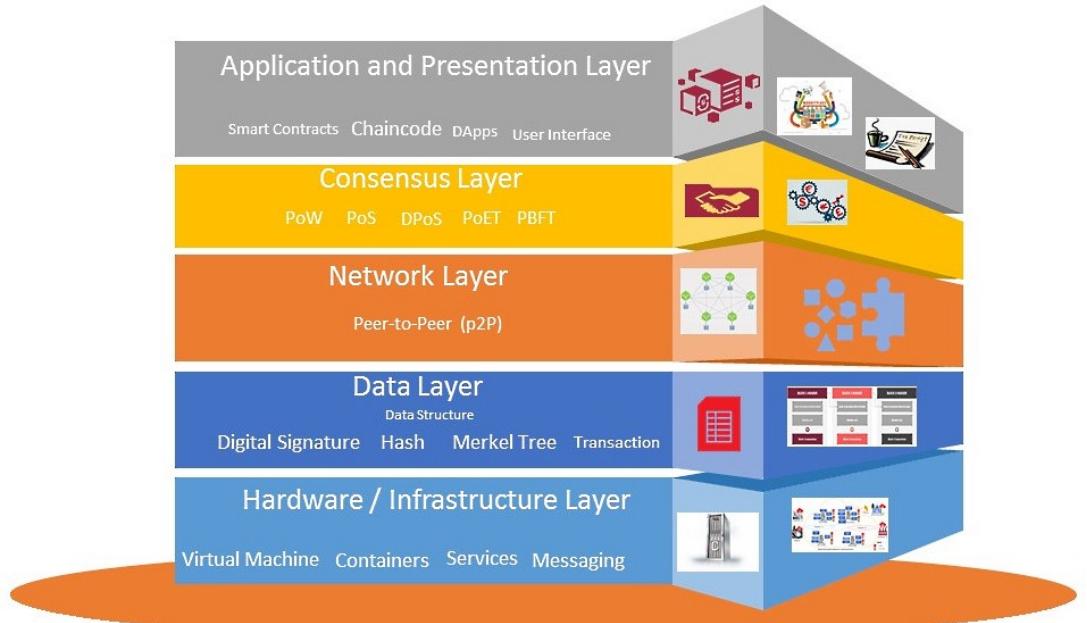
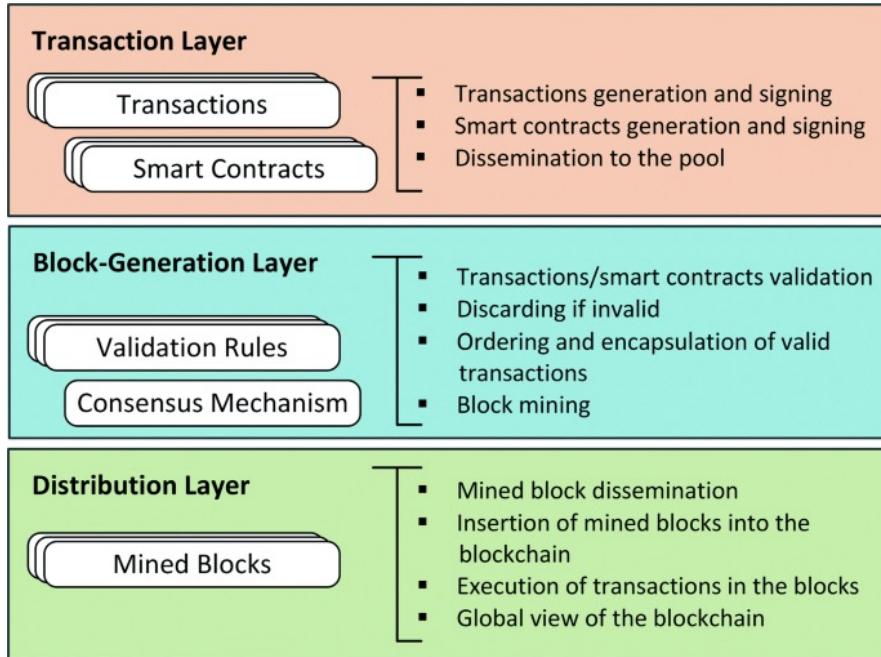
I processi di convalida delle transazioni e mining dei blocchi risiedono nel livello block-generation. Tutte le transazioni sono inserite in un blocco candidato, secondo regole di convalida. L'algoritmo di consenso viene adottato per garantire che tutti i nodi della rete abbiano una vista consistente dello stato della blockchain.

::: Blockchain (6/6)

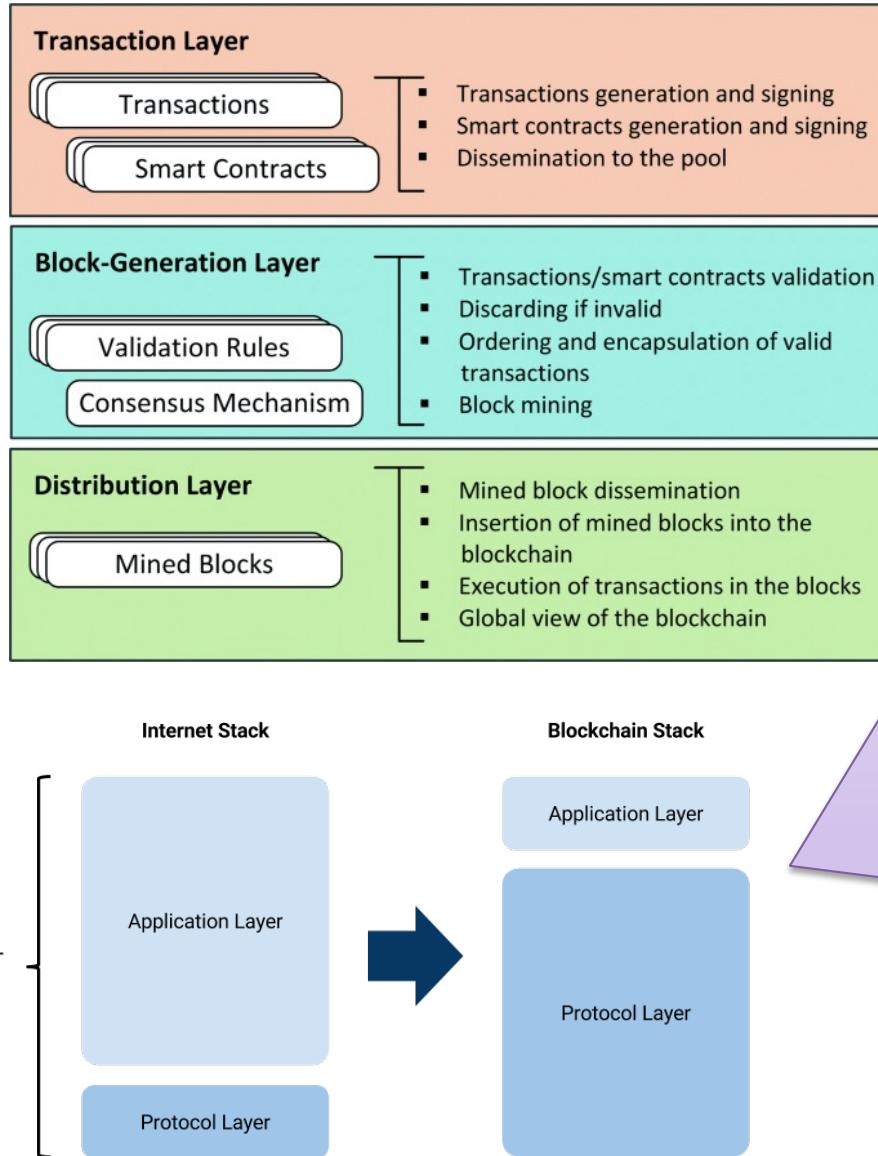


Il processo di distribuzione fa parte del livello di distribuzione, così come l'inserimento del mined block nella blockchain. Tale inserimento riesce solo se l'hash del blocco estratto è corretto, altrimenti viene scartato. Al momento dell'inserimento di un blocco, lo stato globale della blockchain cambia e la vista globale della blockchain viene aggiornata.

::: Blockchain (6/6)

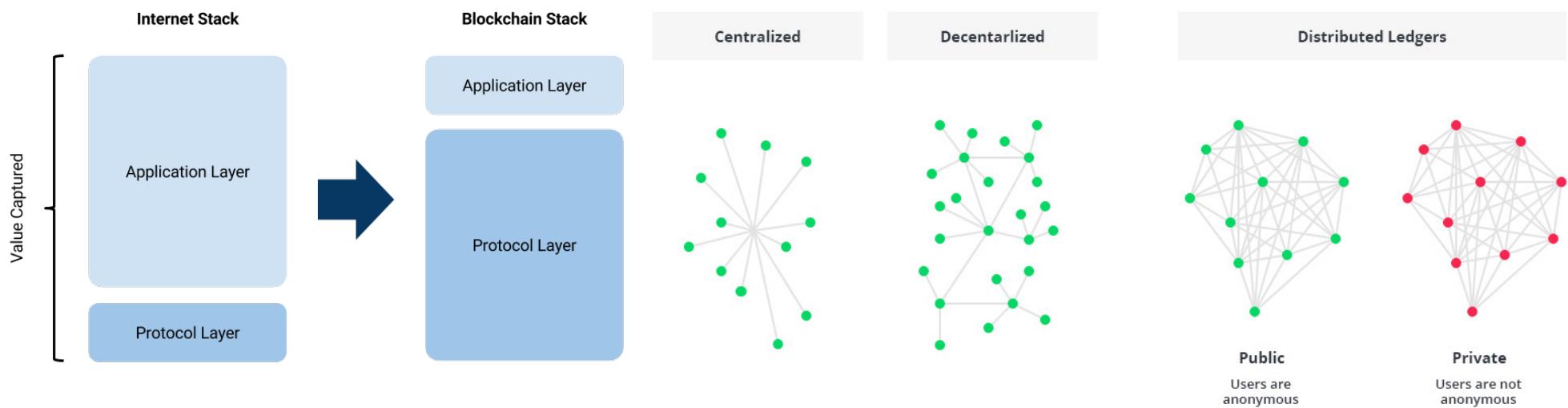
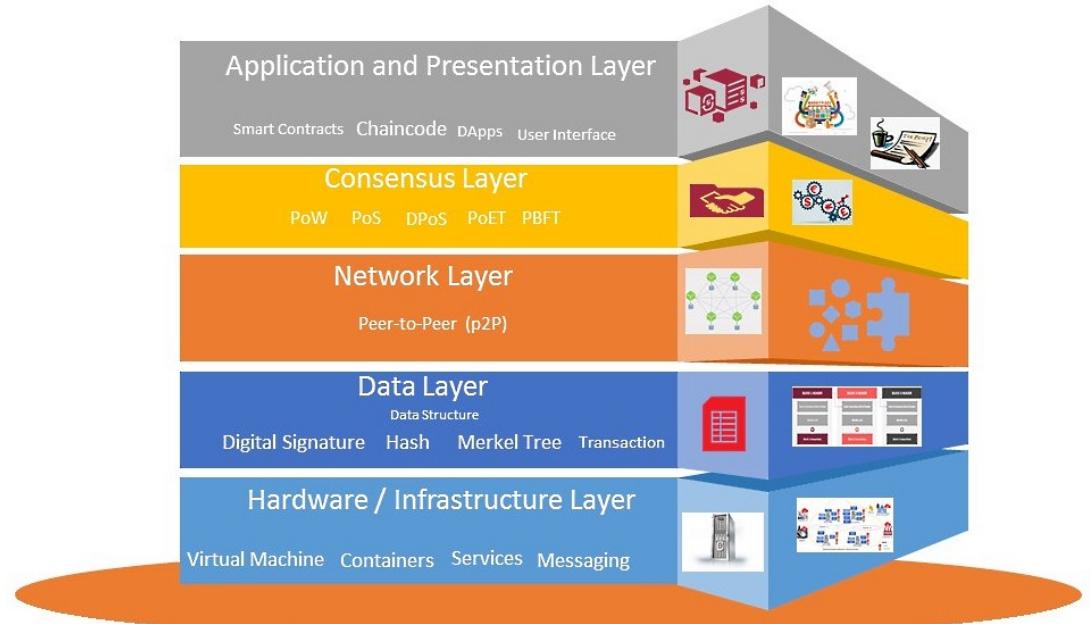
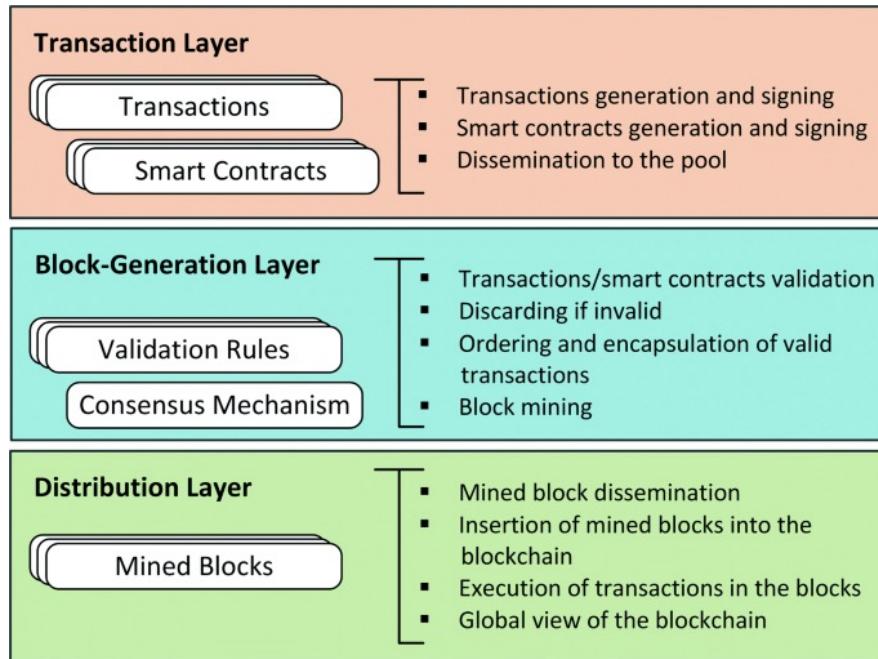


::: Blockchain (6/6)



Il tradizionale stack Internet localizza la maggior parte del valore economico a livello di applicazione (Facebook, Amazon, ecc.), mentre lo stack blockchain vedrà la maggior parte del valore economico detenuto a livello di protocollo. In fatti, il livello delle applicazioni blockchain svolge principalmente il ruolo di interfaccia utente, mentre i vari protocolli gestiscono gran parte delle funzionalità sottostanti e quindi acquisiscono la maggior parte del valore.

.... Blockchain (6/6)



::: Smart Contract

Layer 3 Applications	Financial (Example: Philippines bank system)	Supply Chain (Example: Walmart/IBM food supply chain initiative)	Biomedical and Healthcare (Example: HHS sepsis use case)	Critical Infrastructures (Example: Malaysia's blockchain city)
Layer 2 Smart Contracts	Smart Contracts (Examples: Ethereum Virtual Machine, Hyperledger Chaincode)			
Layer 1 Consensus	Byzantine Fault Tolerance (BFT) Low energy cost Low latency Immediate finality	Proof-of-Work (PoW) High energy cost No immediate finality Allow anybody to join Proof-of-Something (e.g., Proof-of-Elapsed-Time)	Hybrid of Proof-of-Work (Proof-of-Something) and other approaches (e.g., Byzantine Fault Tolerance)	
Category	Permissioned (Participants have to know the identities of each other)		Permissionless (Anybody can join)	Hybrid (Hybrid of both permissioned and permissionless)

L'espressione "smart contract" non sono "contratti" in senso strettamente giuridico, ma funzioni "if/then" incorporate in software o protocolli informatici. Tramite gli smart contract, può anche avvenire una trasposizione "informatica" di accordi che si concludono al di fuori dalla piattaforma tecnologica.



Il Consenso nelle Blockchain

::: Il Consenso nelle Blockchain (1/5)

In una rete blockchain, ogni partecipante può validare transazioni e proporre nuovi blocchi. L'obiettivo del protocollo di consenso nelle blockchain è di garantire che tutti i nodi partecipanti concordino sulla storia comune delle transazioni nella rete.

- Termination: Da ogni nodo onesto, una nuova transazione è sia scartata o accettata nella blockchain, all'interno del contenuto di un blocco.
- Agreement: Ogni nuova transazione e il suo blocco che la contiene deve essere sia scartata o accettata da tutti i nodi onesti. Un blocco accettato deve avere lo stesso numero di sequenza per ogni nodo onesto.
- Validity: Se ogni nodo riceve uno stesso blocco/transazione valida, esso deve essere accettato nella blockchain.
- Integrity: Per ogni nodo onesto, tutte le transazioni accettate devono essere consistenti tra loro (senza double spending). Tutti i blocchi accettati devono essere correttamente generati e collegati con hash in ordine cronologico.

::: Il Consenso nelle Blockchain (1/5)

Sono simili a quanto visto nel classico consenso distribuito, siccome rappresentano la liveliness del sistema.

concordano sulla storia comune delle transazioni nella rete.

- Termination: Da ogni nodo onesto, una nuova transazione è sia scartata o accettata nella blockchain, all'interno del contenuto di un blocco.
- Agreement: Ogni nuova transazione e il suo blocco che la contiene deve essere sia scartata o accettata da tutti i nodi onesti. Un blocco accettato deve avere lo stesso numero di sequenza per ogni nodo onesto.
- Validity: Se ogni nodo riceve uno stesso blocco/transazione valida, esso deve essere accettato nella blockchain.
- Integrity: Per ogni nodo onesto, tutte le transazioni accettate devono essere consistenti tra loro (senza double spending). Tutti i blocchi accettati devono essere correttamente generati e collegati con hash in ordine cronologico.

::: Il Consenso nelle Blockchain (1/5)

Sono simili a quanto visto nel classico consenso distribuito, siccome rappresentano la safety del sistema.

concordin comune delle transazioni nella rete.

- Term: Da ogni nodo onesto, una nuova transazione è sia scartata o accettata nella blockchain, all'interno del contenuto di un blocco.
- Agreement: Ogni nuova transazione e il suo blocco che la contiene deve essere sia scartata o accettata da tutti i nodi onesti. Un blocco accettato deve avere lo stesso numero di sequenza per ogni nodo onesto.
- Validity: Se ogni nodo riceve uno stesso blocco/transazione valida, esso deve essere accettato nella blockchain.
- Integrity: Per ogni nodo onesto, tutte le transazioni accettate devono essere consistenti tra loro (senza double spending). Tutti i blocchi accettati devono essere correttamente generati e collegati con hash in ordine cronologico.

::: Il Consenso nelle Blockchain (1/5)

Questo requisito è potenziato con il total ordering, che rappresenta la serializzazione di blocchi e transazioni.

concordino le regole per accettare alcune delle transazioni nella rete.

- Termination: Per ogni nodo onesto, una nuova transazione è sia scartata o sia accettata nella blockchain, all'interno del contenuto di un blocco.
- Agreement: Ogni nuova transazione e il suo blocco che la contiene deve essere sia scartata o accettata da tutti i nodi onesti. Un blocco accettato deve avere lo stesso numero di sequenza per ogni nodo onesto.
- Validity: Se ogni nodo riceve uno stesso blocco/transazione valida, esso deve essere accettato nella blockchain.
- Integrity: Per ogni nodo onesto, tutte le transazioni accettate devono essere consistenti tra loro (senza double spending). Tutti i blocchi accettati devono essere correttamente generati e collegati con hash in ordine cronologico.

::: Il Consenso nelle Blockchain (1/5)

In una rete blockchain, ogni partecipante può validare transazioni e proporre nuovi blocchi. L'obiettivo del protocollo di consenso nelle blockchain è di garantire che tutti i nodi partecipanti concordino sulla storia comune delle transazioni nella rete.

- Termination: Da ogni nodo onesto, una nuova transazione è sia scartata o accettata nella blockchain, all'interno del ~~contenuto di un blocco~~.

Questo requisito impone la correttezza dell'origine di transazioni e blocchi, consentendo una protezione nel confronti del double-spending e favorendo la tamper-proofing nella blockchain.

sequenza di blocchi.

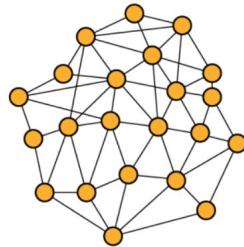
sono generati da un solo nodo onesto.

- Validity: Per ogni nodo riceve uno stesso blocco/transazione valida, essa deve essere accettato nella blockchain.
- Integrity: Per ogni nodo onesto, tutte le transazioni accettate devono essere consistenti tra loro (senza double spending). Tutti i blocchi accettati devono essere correttamente generati e collegati con hash in ordine cronologico.

::: Il Consenso nelle Blockchain (2/5)

C: Consistency

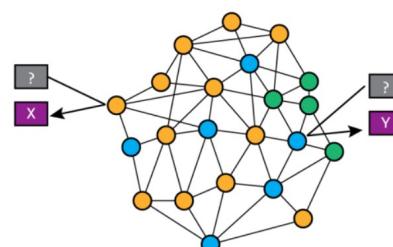
At any given time, all nodes in the network have exactly the same (most recent) **value**.



○ = Value: X @ 2018-05-03T08:52:40

A: Availability

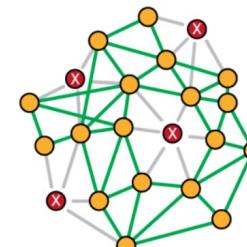
Every **request** to the network receives a **response**, though without any guarantee that returned data is the most recent.



○ = Value: X @ 2018-05-03T08:52:40
● = Value: Z @ 2018-05-03T08:32:58
■ = Value: Y @ 2018-05-03T07:12:12

P : Partition tolerance

The network continues to operate, even if an arbitrary number of nodes are **failing**.



Il teorema CAP si applica anche alle blockchain.

Per i meccanismi di consenso, liveliness e safety hanno una diretta correlazione con il teorema CAP:

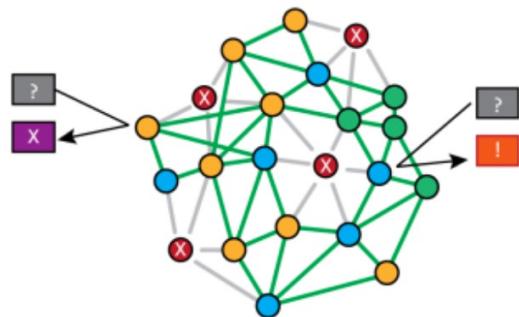
- La **liveliness** garantisce che il processo di consenso completa sempre i suoi round. Anche se non si arriva a consenso, il meccanismo non attende indefinitamente, garantendo la sua **availability**;
- La **safety** garantisce che i suoi partecipanti sono nello stesso stato dopo un round, garantendo la **consistenza** nella rete.

::: Il Consenso nelle Blockchain (3/5)

Nelle reali implementazioni di soluzioni blockchain, non è mai possibile ottenere sia Consistenza che Availability, perché devono affrontare la tolleranza alle partizioni.

Consistency over availability (C+P)

The system will return an **error** or a time-out if particular information cannot be guaranteed to be up to date due to network partitioning (**failing** nodes).



Yellow circle = Value: X @ 2018-05-03T08:52:40

Green circle = Value: Z @ 2018-05-03T08:32:58

Blue circle = Value: Y @ 2018-05-03T07:12:12

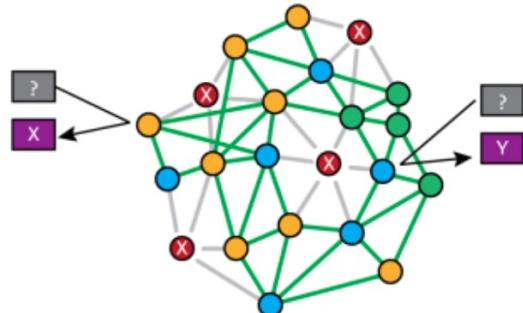
Il sistema restituirà un errore (rosso) o un timeout se non è possibile garantire che particolari informazioni siano aggiornate a causa del partizionamento di rete (nodi (rossi) in errore).

::: Il Consenso nelle Blockchain (3/5)

Nelle reali implementazioni di soluzioni blockchain, non è mai possibile ottenere sia Consistenza che Availability, perché devono affrontare la tolleranza alle partizioni.

Availability over consistency (A+P)

Every request to the network receives a **response**, even if the network cannot guarantee it is up to date due to network partitioning (**failing nodes**).



Orange circle = Value: X @ 2018-05-03T08:52:40

Green circle = Value: Z @ 2018-05-03T08:32:58

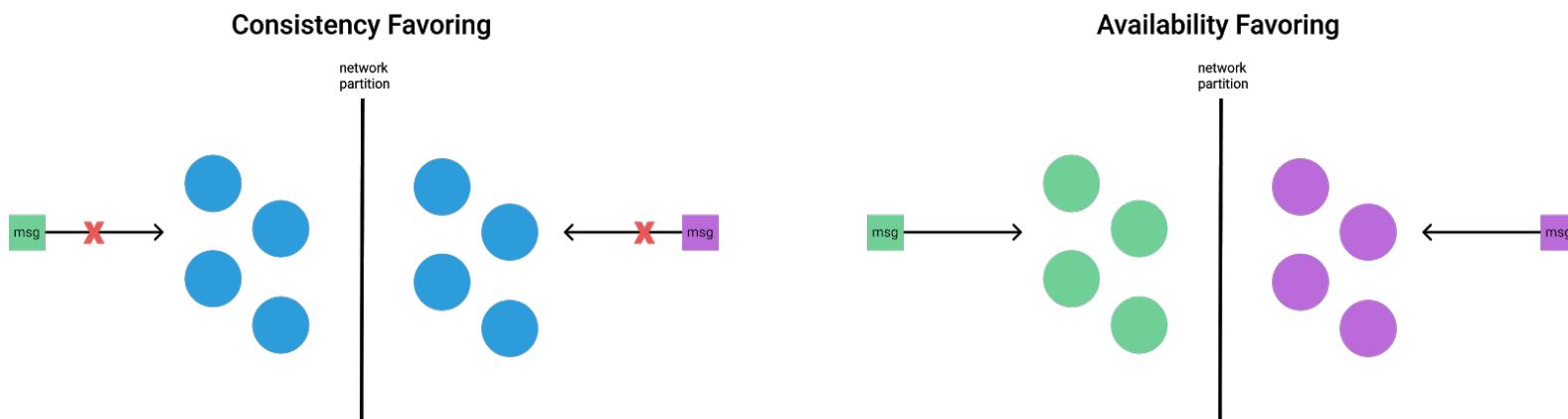
Blue circle = Value: Y @ 2018-05-03T07:12:12

Ogni richiesta (grigia) alla rete riceve una risposta (viola), anche se la rete non può garantire che sia aggiornata a causa del partizionamento di rete (nodi (rossi) in errore).

::: Il Consenso nelle Blockchain (3/5)

Nelle reali implementazioni di soluzioni blockchain, non è mai possibile ottenere sia Consistenza che Availability, perché devono affrontare la tolleranza alle partizioni.

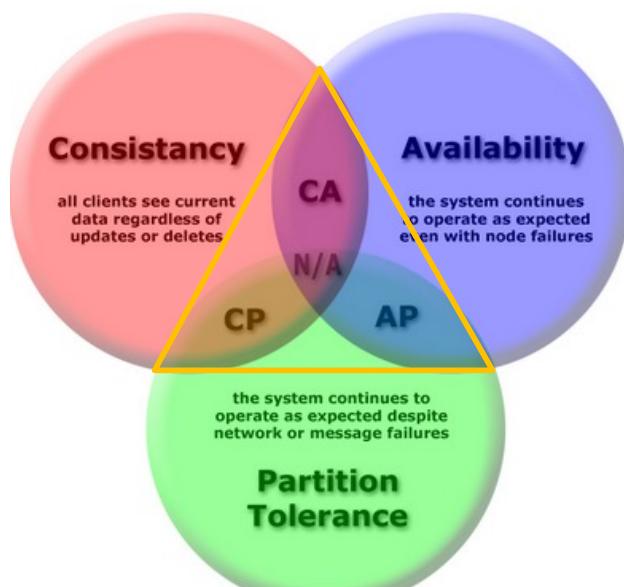
Tra le due opzioni rimanenti, i sistemi di tipo permissionless (come Bitcoin) in un gruppo aperto di nodi scelgono di privilegiare di garantire l'Availability anziché la Consistency, per poter essere in grado di utilizzare/inviare/ricevere criptovalute. I casi di fork che rappresentano i momenti di inconsistenza sono risolti nel tempo, così da garantire l'Eventual Consistency.



::: Il Consenso nelle Blockchain (3/5)

Nelle reali implementazioni di soluzioni blockchain, non è mai possibile ottenere sia Consistenza che Availability, perché devono affrontare la tolleranza alle partizioni.

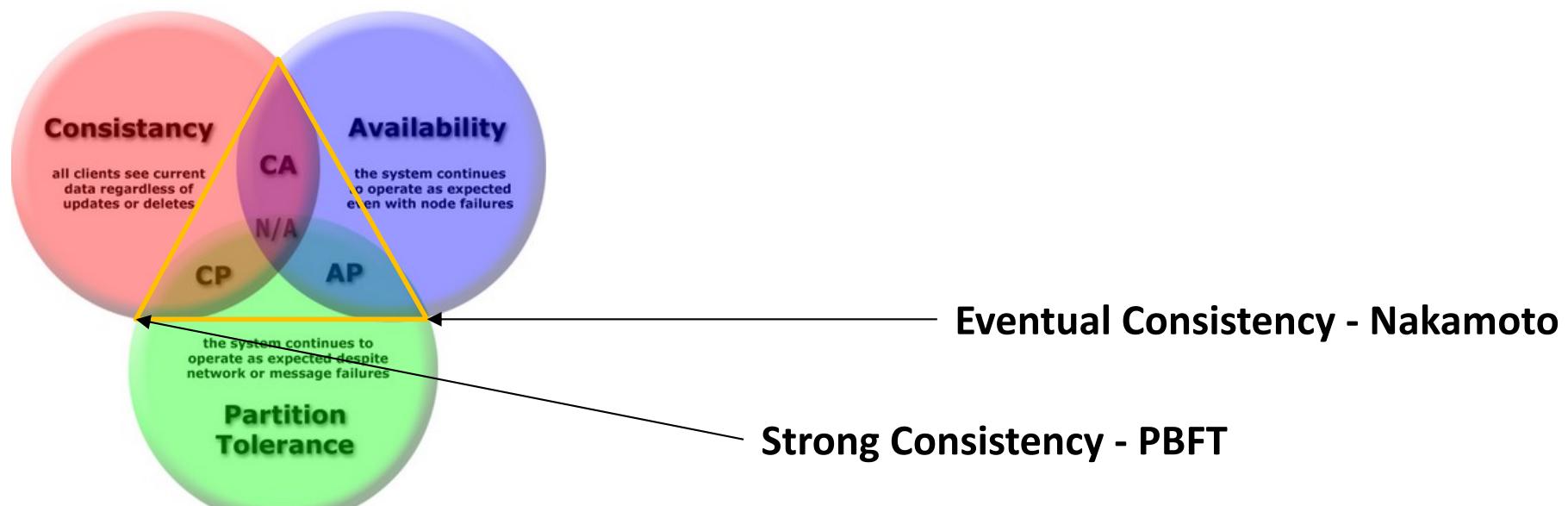
Tra le due opzioni rimanenti, i sistemi di tipo permissioned in un gruppo chiuso di nodi scelgono di privilegiare di garantire la Consistency anziché l'Availability, perché non sono focalizzate su cryptovalute ma la gestione di dati in ambito distribuito.



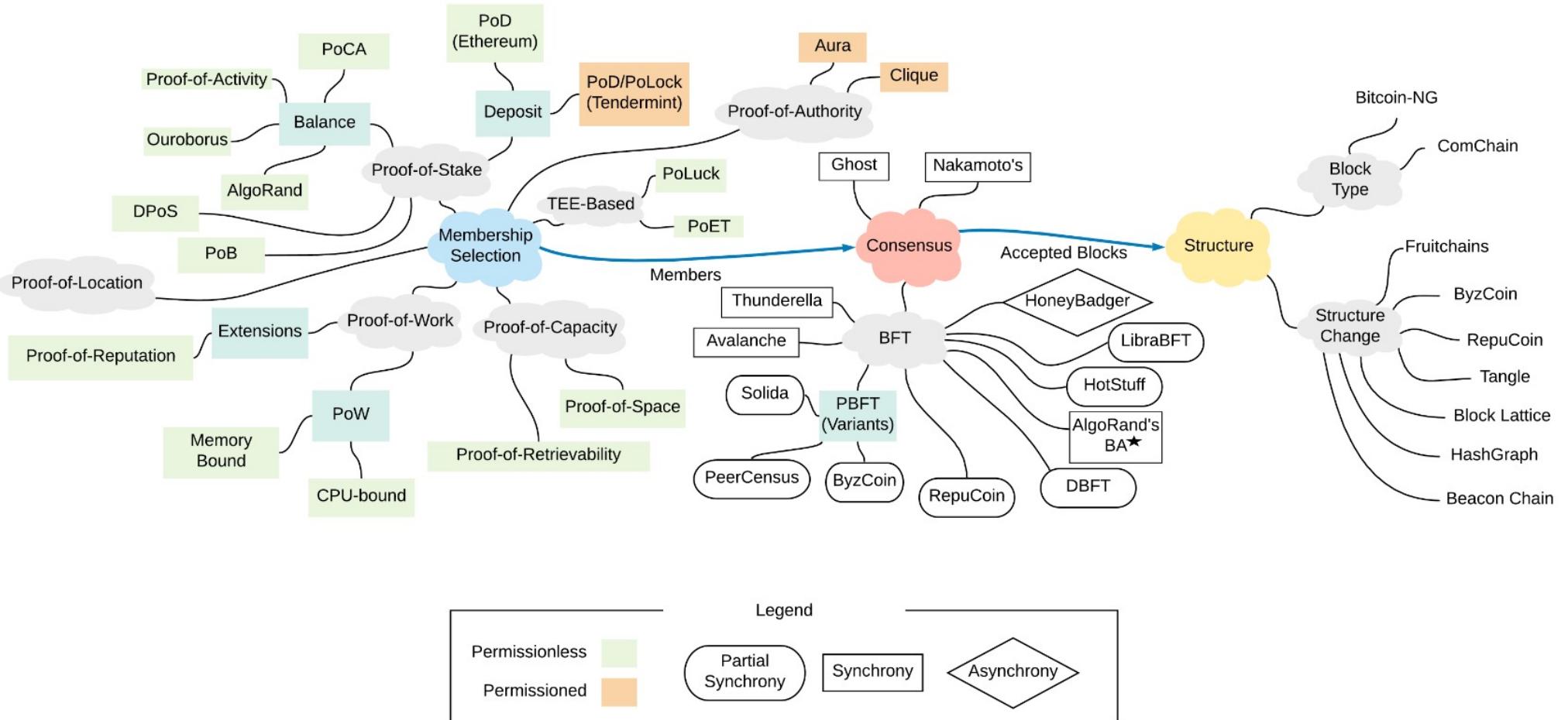
::: Il Consenso nelle Blockchain (3/5)

Nelle reali implementazioni di soluzioni blockchain, non è mai possibile ottenere sia Consistenza che Availability, perché devono affrontare la tolleranza alle partizioni.

Tra le due opzioni rimanenti, i sistemi di tipo permissioned in un gruppo chiuso di nodi scelgono di privilegiare di garantire la Consistency anziché l'Availability, perché non sono focalizzate su cryptovalute ma la gestione di dati in ambito distribuito.

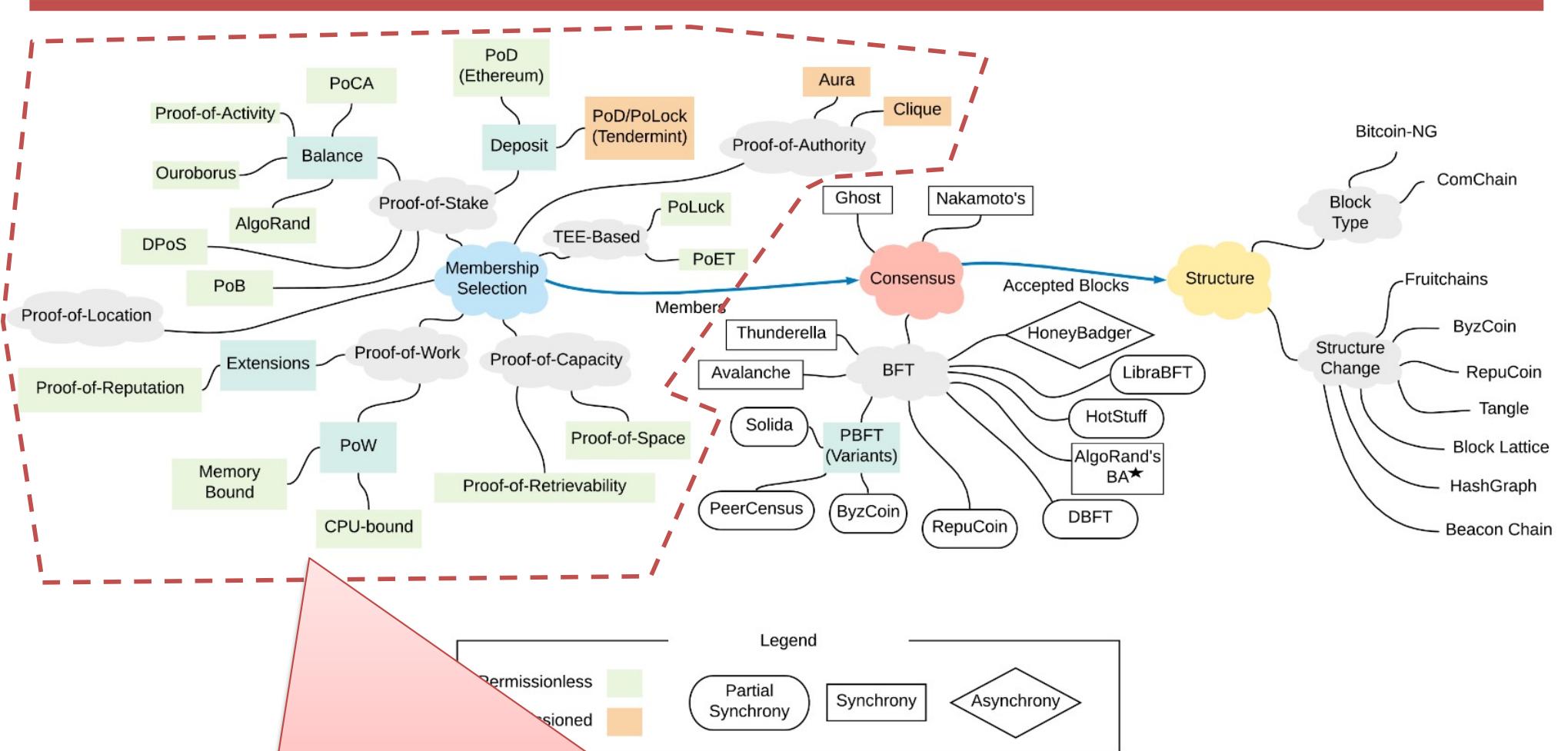


::: Il Consenso nelle Blockchain (4/5)



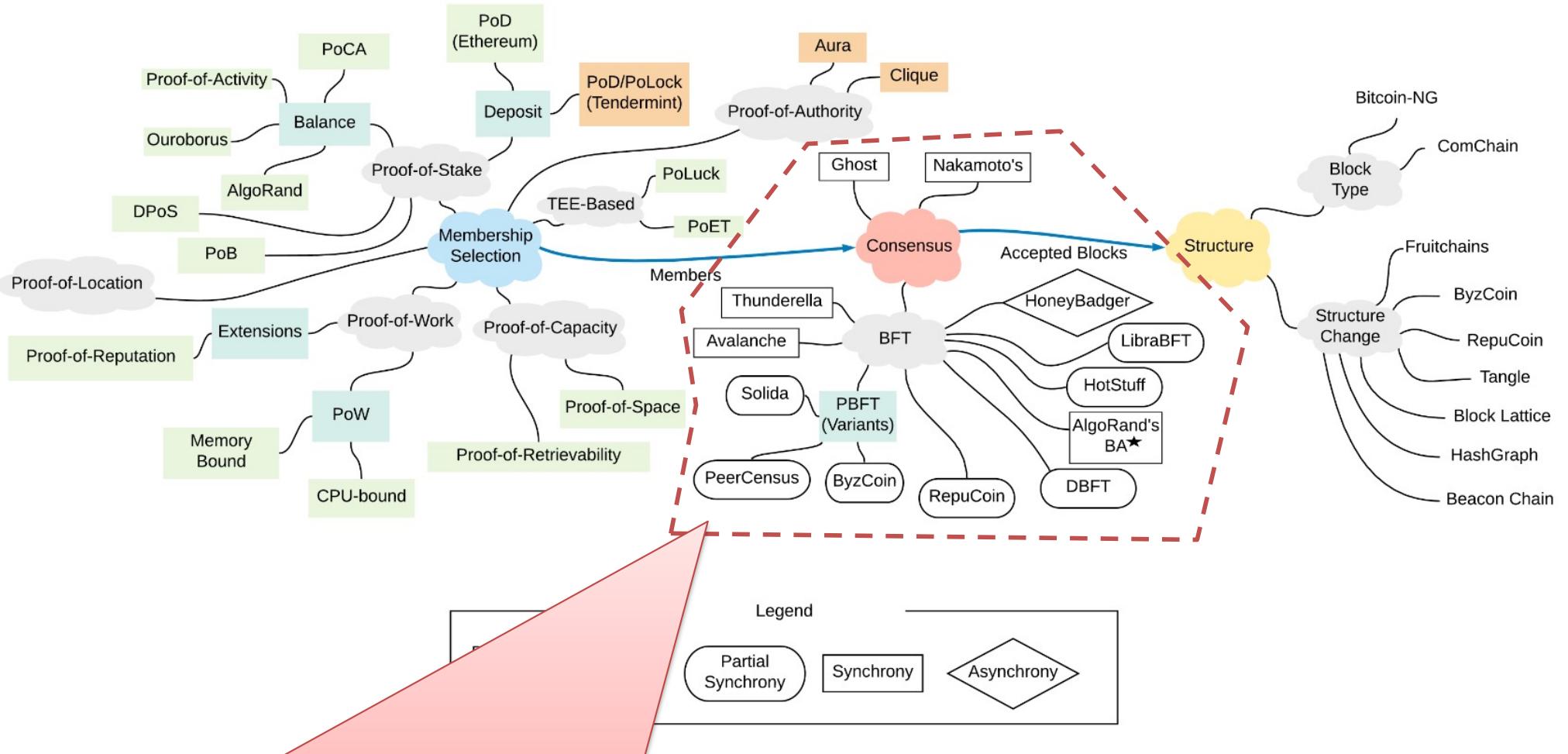
Nei meccanismi di consenso delle blockchain possiamo distinguere tre elementi chiave.

::: Il Consenso nelle Blockchain (4/5)



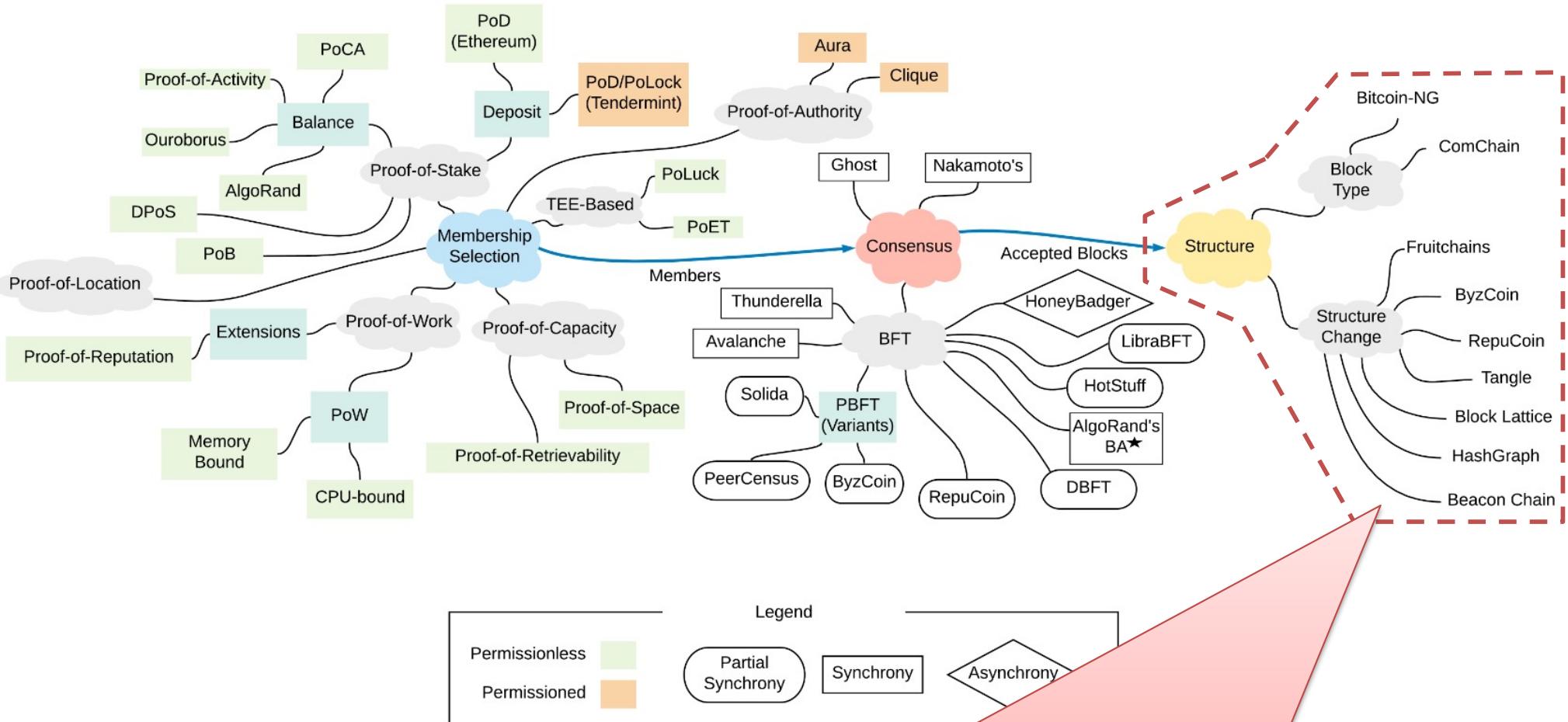
Come vengono selezionati i partecipanti al consenso.

::: Il Consenso nelle Blockchain (4/5)



Come viene realizzato il processo di raggiungimento del consenso.

::: Il Consenso nelle Blockchain (4/5)



Come sono strutturati i dati nella blockchain.

::: Il Consenso nelle Blockchain (5/5)

Il protocollo di consenso si compone di 5 elementi:

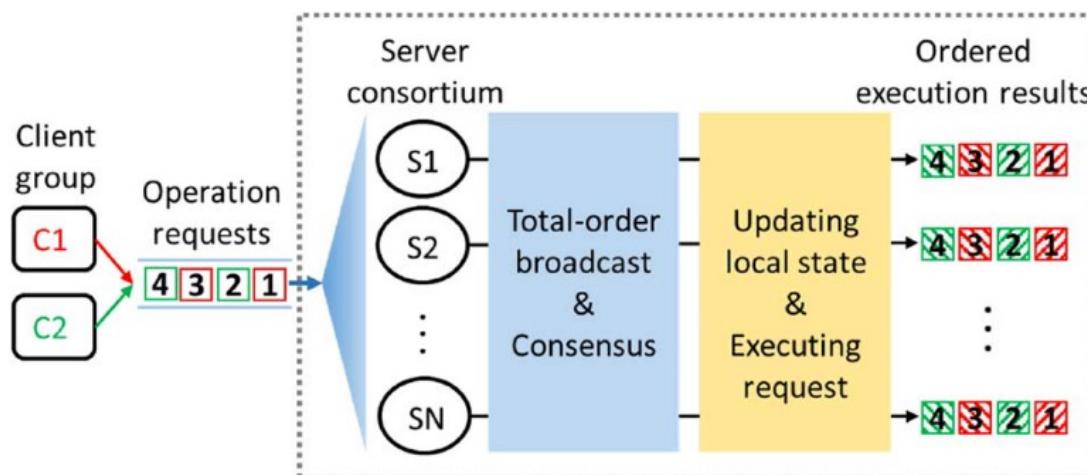
1. La proposta di un blocco: generazione di blocchi e associazione di prove;
2. La propagazione di informazioni: disseminazione di blocchi e transazioni nella rete;
3. La validazione di blocchi: controllo dei blocchi per la generazione di prove e verifica della validità delle transazioni;
4. La finalizzazione dei blocchi: raggiungimento del consenso sull'accettazione di blocchi validati;
5. Il meccanismo di incentivazione: promozione di partecipanti onesti e creazione di token di rete.

Il consenso su blockchain si ispira al meccanismo di State Machine Replication (SMR).

::: State Machine Replication (1/3)

SMR stabilisce i seguenti requisiti:

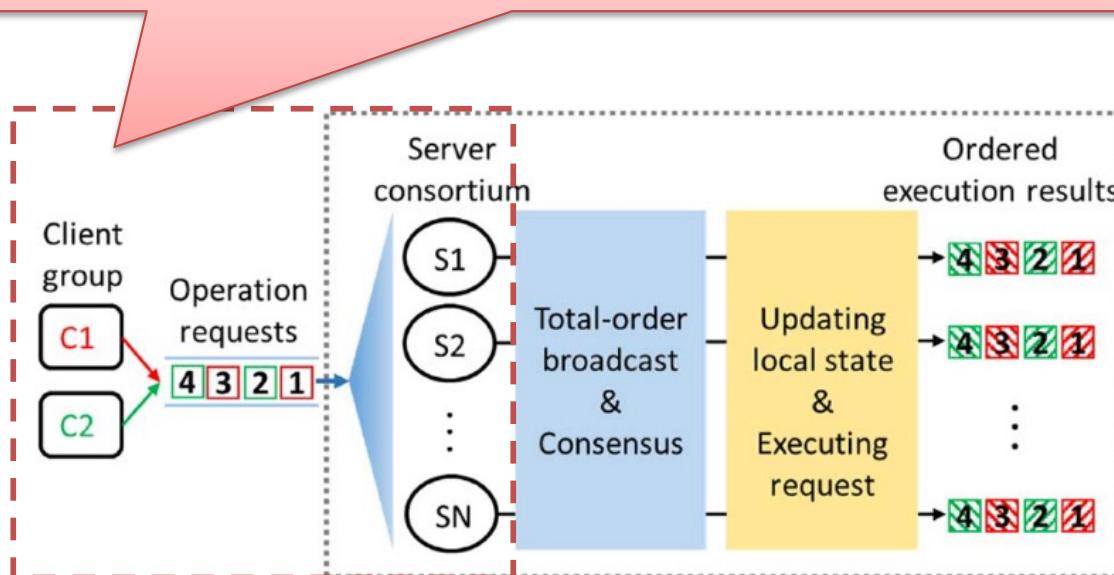
1. Tutti i server iniziano con lo stesso stato iniziale;
2. Total-order broadcast: tutti i server ricevono la stessa sequenza di richieste secondo l'ordine di generazione dai client;
3. Tutti i server che ricevono la stessa richiesta emetteranno gli stessi risultati di esecuzione e termineranno nello stesso stato.



::: State Machine Replication (1/3)

SMR stabilisce i seguenti requisiti:

1. Tutti i server iniziano con lo stesso stato iniziale;
2. Total-order broadcast: tutti i server ricevono la stessa sequenza di richieste secondo l'ordine di generazione dai client;
3. Un consorzio di N server accetta le richieste di stessi operazioni dai client.

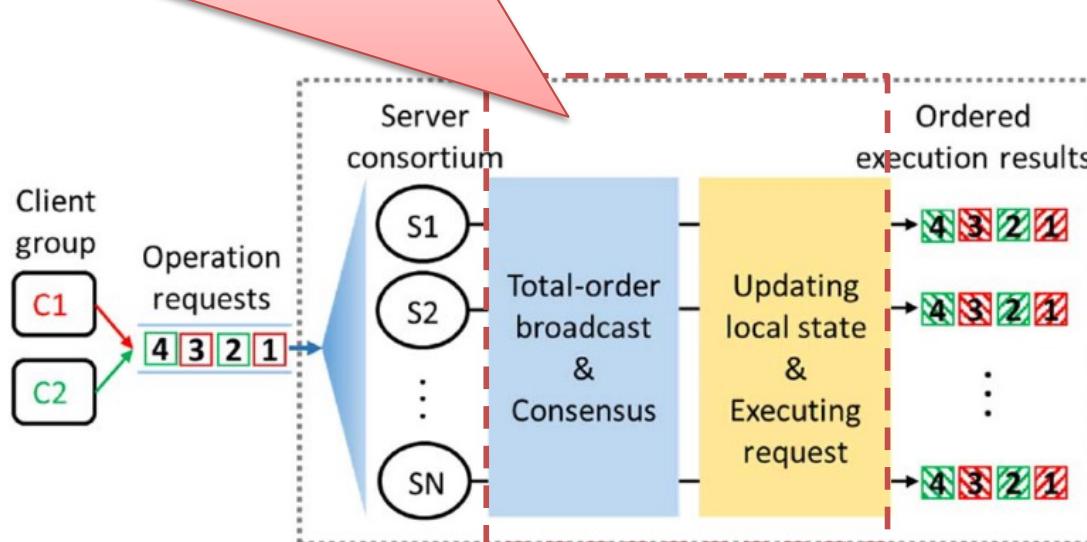


::: State Machine Replication (1/3)

SMR stabilisce i seguenti requisiti:

1. Tutti i server iniziano con lo stesso stato iniziale;
2. Total-order broadcast: tutti i server ricevono la stessa sequenza di richieste secondo l'ordine di generazione dai client;
3. Tutti i server confermano il proprio stato prima di raggiungere il consenso ed eseguire le richieste

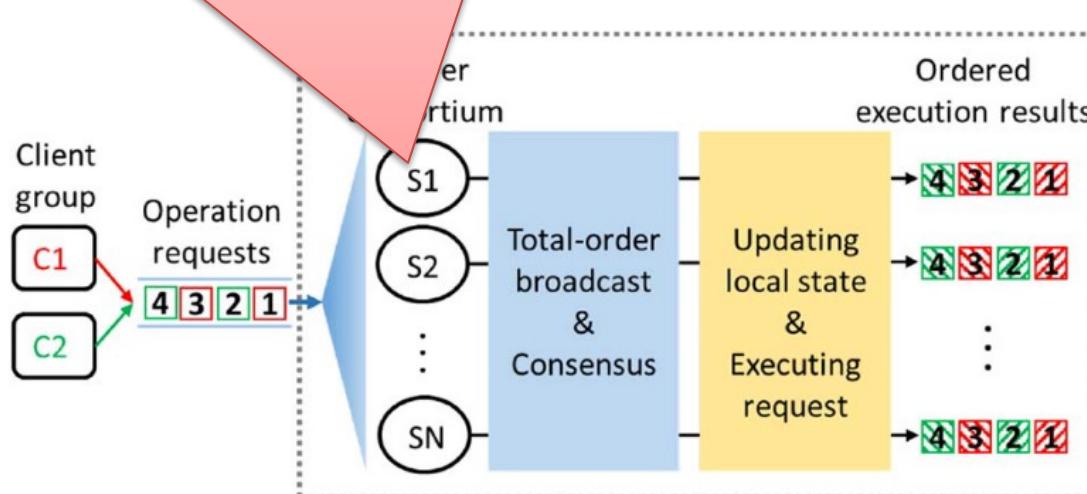
no gli
stato.



::: State Machine Replication (1/3)

SMR stabilisce i seguenti requisiti:

1. Tutti i server iniziano con lo stesso stato iniziale;
2. Total-order broadcast: tutti i server ricevono la stessa sequenza di richieste. SMR è spesso realizzato in maniera leader-based, con un server primario (ad es. S1) che riceve le richieste dai client, le ordina e le invia alle altre parti del sistema. I server ricevono le stesse richieste e aggiornano il proprio stato in modo che corrisponda a quello del leader.
3. Tutti i server ricevono le stesse richieste e aggiornano il proprio stato locale in modo che corrisponda a quello del leader.



::: State Machine Replication (2/3)

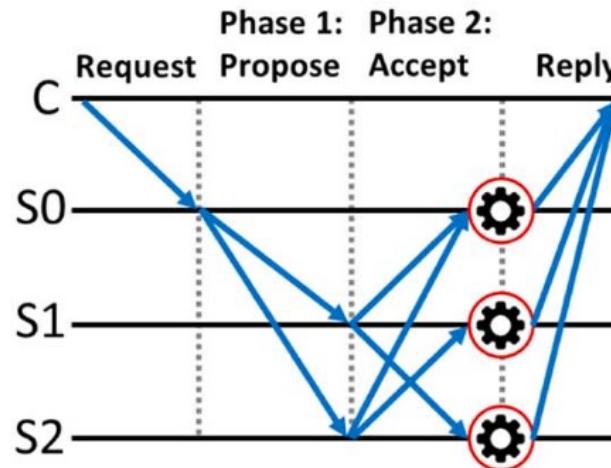
Sussiste una corrispondenza tra gli elementi del consenso nelle blockchain e quelli in SMR:

1. La proposta di blocchi corrisponde alle richieste di operazioni da parte dei client in SMR e il leader che inizia il consenso;
2. La propagazione delle informazioni corrisponde al reliable broadcast delle richieste di operazioni;
3. La validazione dei blocchi corrisponde alla verifica delle firme e l'esecuzione delle operazioni richieste;
4. La finalizzazione dei blocchi corrisponde al raggiungimento del consenso da parte dei server sullo stato corrente;
5. Il meccanismo di incentivo non trova una corrispondenza. Questo perché SMR presuppone un gruppo ben definito di partecipanti che si presuppongono onesti.

::: State Machine Replication (3/3)

L'algoritmo di consenso di Paxos è uno schema SMR progettato per garantire il consenso tollerante a guasti di tipo crash.

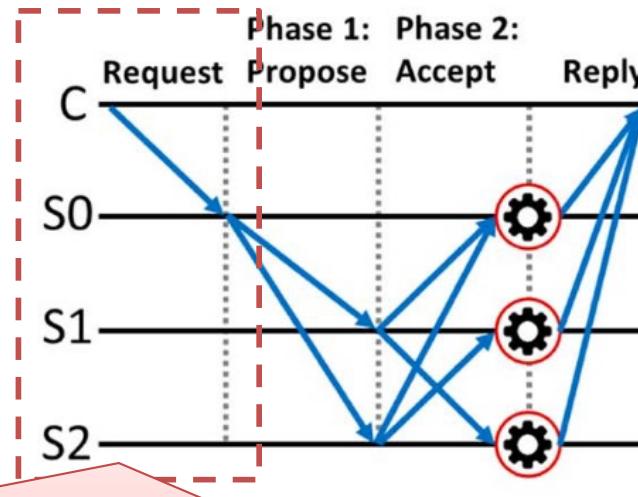
- Un proposer suggerisce un valore all'inizio e lo scopo del sistema è di far sì che gli acceptor concordano su un singolo valore e i learners apprendano tale valore dagli acceptor.



::: State Machine Replication (3/3)

L'algoritmo di consenso di Paxos è uno schema SMR progettato per garantire il consenso tollerante a guasti di tipo crash.

- Un proposer suggerisce un valore all'inizio e lo scopo del sistema è di far sì che gli acceptor concordano su un singolo valore e i learners apprendano tale valore dagli acceptor.

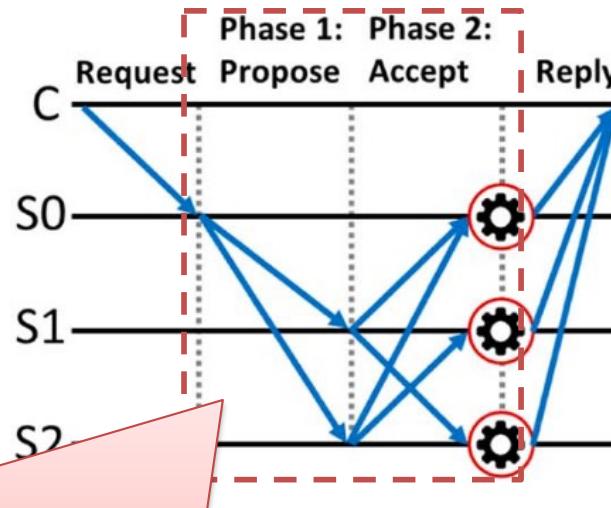


Il client è un learner e il leader tra i server è un proposer, mentre le repliche sono degli acceptors. Il client richiede il consenso su un singolo valore.

::: State Machine Replication (3/3)

L'algoritmo di consenso di Paxos è uno schema SMR progettato per garantire il consenso tollerante a guasti di tipo crash.

- Un proposer suggerisce un valore all'inizio e lo scopo del sistema è di far sì che gli acceptor concordano su un singolo valore e i learners apprendano tale valore dagli acceptor.

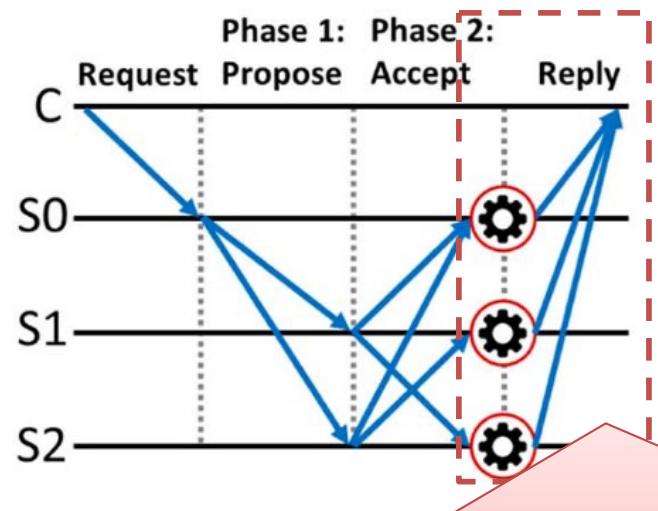


Il proposer propaga la richiesta agli acceptors, che si scambiano informazioni sui propri stati. Dopo essersi aggiornati allo stesso stato, tutti i server eseguono la richiesta, che la inviano al client.

::: State Machine Replication (3/3)

L'algoritmo di consenso di Paxos è uno schema SMR progettato per garantire il consenso tollerante a guasti di tipo crash.

- Un proposer suggerisce un valore all'inizio e lo scopo del sistema è di far sì che gli acceptor concordano su un singolo valore e i learners apprendano tale valore dagli acceptor.

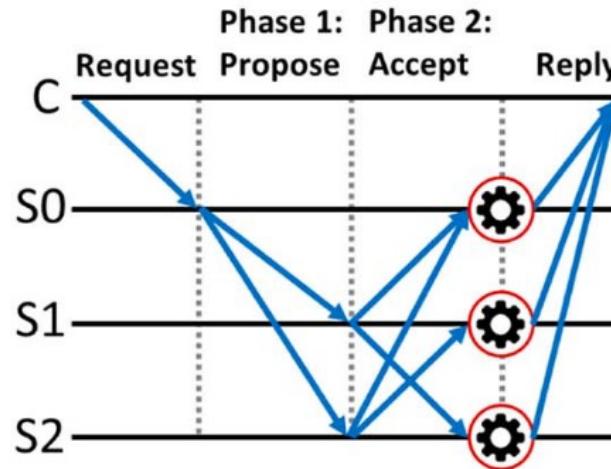


Il client riceve i risultati dagli acceptor e formula l'esito a maggioranza. Quando il leader è indisponibile per crash, le repliche ne eleggono uno nuovo.

::: State Machine Replication (3/3)

L'algoritmo di consenso di Paxos è uno schema SMR progettato per garantire il consenso tollerante a guasti di tipo crash.

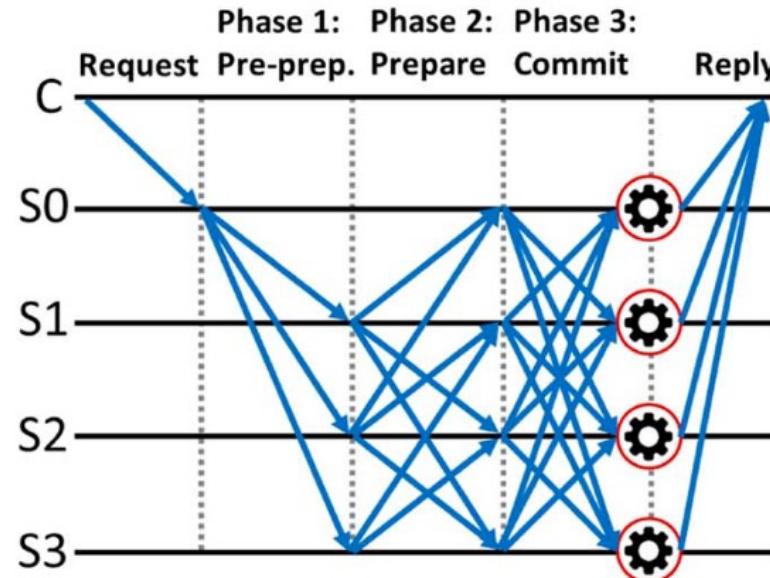
- Un proposer suggerisce un valore all'inizio e lo scopo del sistema è di far sì che gli acceptor concordano su un singolo valore e i learners apprendano tale valore dagli acceptor.



- Questo algoritmo tollera f crash dei server il cui numero N è maggiore o uguale di $2f + 1$, ma non tollera guasti bizantini.

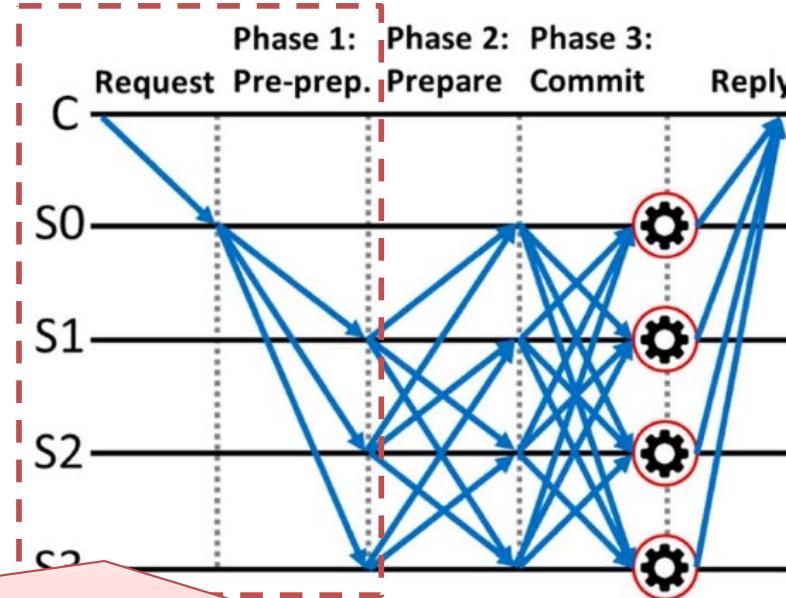
::: PBFT (1/4)

L'algoritmo di Practical Byzantine Fault Tolerance (PBFT) è uno schema SMR che tollera i guasti bizantini ed è diventato sinonimo di tolleranza ai guasti bizantini (BFT) nel contesto delle blockchain. Si compone di tre fasi:



::: PBFT (1/4)

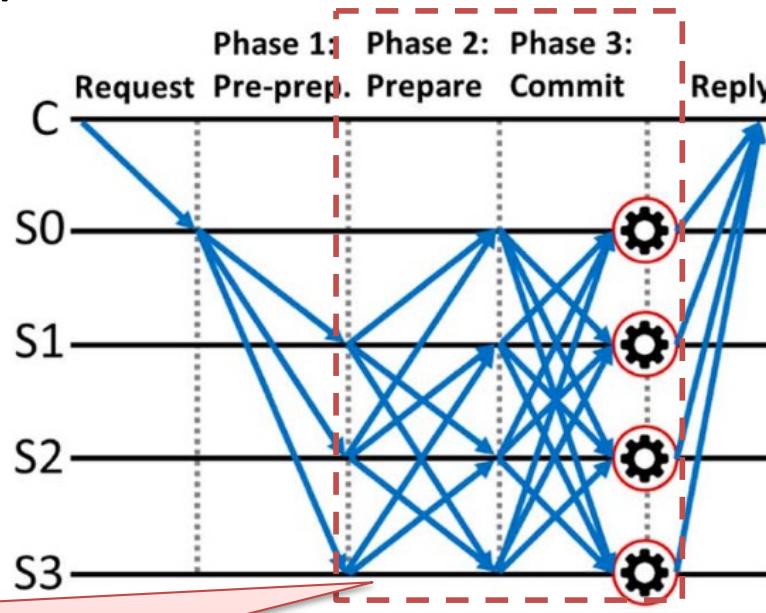
L'algoritmo di Practical Byzantine Fault Tolerance (PBFT) è uno schema SMR che tollera i guasti bizantini ed è diventato sinonimo di tolleranza ai guasti bizantini (BFT) nel contesto delle blockchain. Si compone di tre fasi:



Tutti i risultati inviati al client devono essere uguali, altrimenti il client decide a maggioranza. Il client invia una richiesta al nodo primario (leader), che la trasmette a tutti i nodi secondari (backup) assegnando un numero di sequenza.

::: PBFT (1/4)

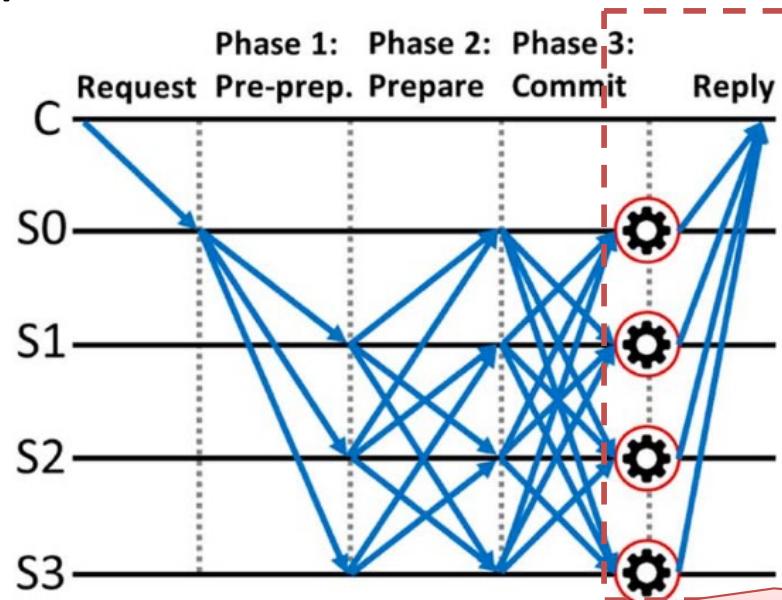
L'algoritmo di Practical Byzantine Fault Tolerance (PBFT) è uno schema SMR che tollera i guasti bizantini ed è diventato sinonimo di tolleranza ai guasti bizantini (BFT) nel contesto delle blockchain. Si compone di tre fasi:



I nodi secondari e il leader si accordano sull'ordinamento delle richieste, quando l'ordinamento è stato approvato, la richiesta viene eseguita e un risultato restituito al client.

::: PBFT (1/4)

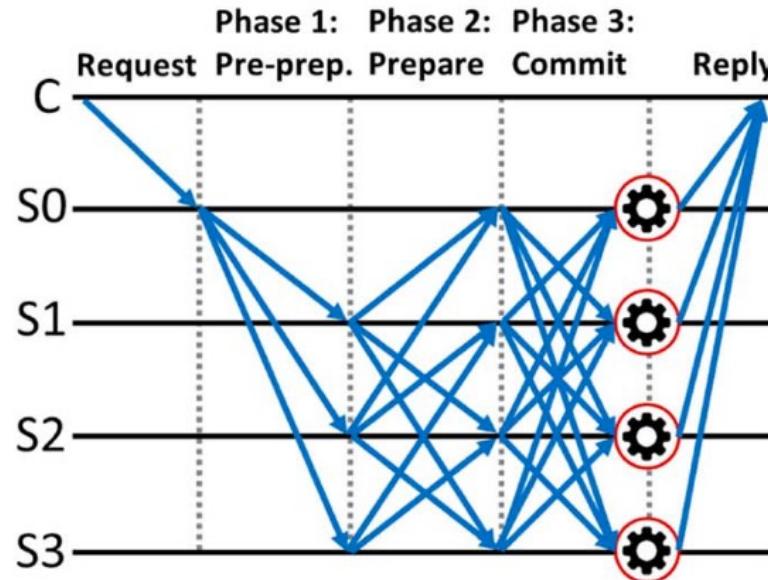
L'algoritmo di Practical Byzantine Fault Tolerance (PBFT) è uno schema SMR che tollera i guasti bizantini ed è diventato sinonimo di tolleranza ai guasti bizantini (BFT) nel contesto delle blockchain. Si compone di tre fasi:



Se il client riceve $f + 1$ risposte identiche, si raggiunge consenso.

::: PBFT (1/4)

L'algoritmo di Practical Byzantine Fault Tolerance (PBFT) è uno schema SMR che tollera i guasti bizantini ed è diventato sinonimo di tolleranza ai guasti bizantini (BFT) nel contesto delle blockchain. Si compone di tre fasi:



Il leader è scelto per ogni round dell'algoritmo o vista. I nodi sono ordinati in base al proprio identificativo, ed indicando il numero della vista con v , il leader è il nodo con identificativo $i = v \bmod N$.

::: PBFT (2/4)

Quando una replica nota un comportamento errato da parte del leader, se ne può richiedere la sostituzione con il meccanismo della view change e l'elezione di un nuovo leader.

L'algoritmo consente di tollerare f guasti anche di natura bizantina, con N maggiore o uguale di $3f + 1$, ovvero i nodi bizantini non riescono a far deviare il consenso raggiunto.

La primitiva di comunicazione alla base delle implementazioni PBFT nel contesto delle blockchain è il gossiping, impiegato per la propagazione di messaggi di blocchi o transazioni.

A differenza della formulazione classica, nelle reti blockchain si ha la terminazione probabilistica: Per ogni nodo onesto, ogni nuovo blocco è sia scartato o accettato nella sua blockchain locale. Un blocco accettato può essere ancora scartato ma con probabilità decrescente in maniera esponenziale con la crescita della dimensione della catena.

::: PBFT (3/4)

Il principale problema con la PBFT è che richiede che i nodi verifichino la validità dei messaggi degli altri e che il numero di nodi attivi in un dato momento sia sempre noto, e ciò lo rende applicabile in contesti permissioned.

Un altro svantaggio è che i leader vengono sostituiti solo quando le view change vengono attivate dalla rete. L'opportunità di diventare un leader è quindi unfair e mancano pochi incentivi per entrare a far parte della rete. Blockchain elegge i leader in base alla difficoltà del lavoro svolto, che genera incentivi, anche se spreca potenza di calcolo.

La sicurezza di PBFT si basa sul voto in tre fasi con MAC (Message Authentication Code) per la verifica dei messaggi. Sebbene non consumi molte risorse di elaborazione, crea inevitabilmente problemi di scalabilità: in PBFT è impossibile espandersi oltre i 1000 nodi.

::: PBFT (4/4)

PBFT garantisce fortemente il requisito di Safety:

- un fork è quasi impossibile e non viene garantita la terminazione immediata.

Al contrario, la blockchain con Nakamoto è più focalizzata sulla liveliness che sulla safety:

- i fork si verificano abbastanza frequentemente (consenso multiplo) e affinché un blocco sia sicuro la sua catena deve essere più lunga di un certo numero di blocchi.

A tale scopo si è formulato un diverso algoritmo di consenso, quello di Nakamoto. Lo scopo è di evitare il consenso in gruppi chiusi, e di non punire i singoli nodi in un gruppo aperto per essersi comportati in modo malevolo, ma bisogna disincentivare i nodi a replicare comportamenti malevoli.

::: PBFT (4/4)

Il processo di mining dei blocchi (ovvero calcolo della PoW) è stocastico, per cui è impossibile sapere con certezza chi troverà la soluzione, anche se al crescere della difficoltà i nodi capaci di portare avanti il processo diminuiscono. Questo scoraggia tutti gli agenti non disposti ad investire risorse economiche a partecipare al gioco.

Con il crescere dell'uso, e quindi del valore, la difficoltà a minare bitcoin aumenta, disincentivando ulteriormente chiunque voglia attaccare la rete. Inoltre, il crescente valore costringe gli agenti onesti ad investire di più nella sicurezza dei propri nodi e, di conseguenza, della rete in generale.

Le regole di validazione dei blocchi assicurano che nessun agente onesto accetti blocchi con informazioni scorrette.

A tale scopo, si è scelto un diverso algoritmo di consenso, quello di Nakamoto. L'obiettivo è di evitare il consenso in gruppi chiusi, e di non punire i singoli nodi in un gruppo aperto per essersi comportati in modo malevolo, ma bisogna disincentivare i nodi a replicare comportamenti malevoli.

::: Il Consenso Nakamoto (1/7)

Rispetto al consenso di Nakamoto implementato nel contesto della BitCoin, è possibile trovare un parallelismo con le 5 componenti del consenso nelle blockchain:

1. La generazione di blocchi richiede una Proof-of-Work mediante la risoluzione di un puzzle crittografico con un determinato grado di difficoltà tale da mantenere un intervallo di generazione e un grado di protezione adatti;
2. Il gossiping viene impiegato per la distribuzione dei blocchi/transazioni appena ricevuto o localmente generati;
3. Un blocco o transazione deve essere validata prima di essere inviata in broadcast agli altri o collegata alla coda di una catena locale. La validità si realizza evitando la double-spending o controllando la PoW allegata al blocco.

::: Il Consenso Nakamoto (2/7)

4. La catena più lunga rappresenta il raggiungimento del consenso in caso di disaccordo (che ha causato la fork).
5. Chi ha generato un blocco accettato con successo può ottenere un reward o premio. Sottomettere una nuova transazione ha un costo monetario.

L'impiego di intensive PoW è necessario per evitare e tollerare attacchi Sybil, a causa della natura permissionless e pseudonima di alcune reti blockchain. Un attaccante può ottenere facilmente delle nuove identità, ma la risoluzione di una PoW implica il consumo di risorse di hash, che può essere difficilmente falsata.

Le reward per i blocchi e il costo delle transazioni serve ad incentivare i nodi a partecipare onestamente.

::: Il Consenso Nakamoto (3/7)

Nelle classiche formulazioni del consenso distribuito, la caratteristica di tolleranza ai guasti è espressa in termini di numero di nodi non corretti che si possono tollerare. Nel caso del consenso di Nakamoto è caratterizzata in termini di percentuale di potenza di hashing avversaria tollerabile.

- Se la rete si sincronizza più velocemente del tasso di proposta di blocco basato su PoW, una maggioranza onesta può garantire il consenso su una parte stabile in continua crescita della blockchain.
- Fintanto che meno del 50% della potenza di hashing totale è controllata in modo malizioso, i blocchi prodotti da miners onesti vengono propagati tempestivamente, la catena principale è della maggioranza onesta che eventualmente supera qualsiasi ramo malizioso.

::: Il Consenso Nakamoto (4/7)

Dal punto di vista del consenso distribuito classico, il consenso di Nakamoto elude abilmente il vincolo fondamentale di BFT pari a 1/3 adottando finalità probabilistiche.

- Nel consenso del BFT classico se più di 1/3 della popolazione è malizioso, i nodi onesti finiranno per decidere valori contrastanti, portando al fallimento del consenso.
- Nel consenso di Nakamoto, tuttavia, le decisioni contrastanti sono consentite temporaneamente sotto forma di fork della blockchain, a condizione che alla fine verranno eliminate dal continuo sforzo della maggioranza onesta.

Il consenso di Nakamoto soffre di alcune limitazioni, primo tra tutti un basso throughput delle transazioni.

::: Il Consenso Nakamoto (5/7)

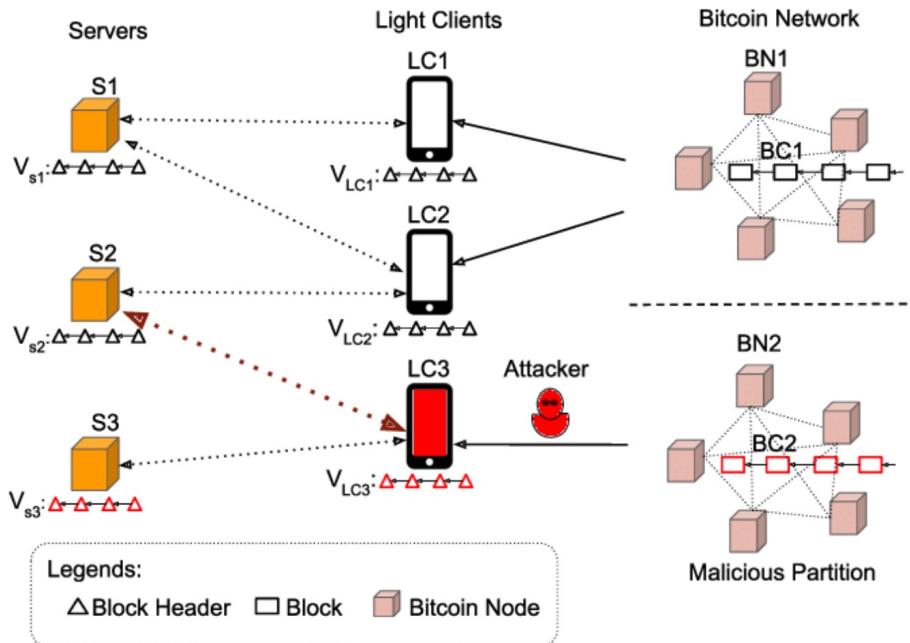
Si può dimostrare che l'intervallo di 10 minuti tra la generazione di blocchi garantisce che ogni nuovo blocco sia sufficientemente propagato prima che venga inserito nel sistema un nuovo blocco.

- Ridurre l'intervallo tra i blocchi aumenta il throughput delle transazioni, ma lascia i nuovi blocchi non sufficientemente propagati e provoca più incidenti di fork, minando la sicurezza della catena principale.

Aumentare la dimensione del blocco (attualmente 1 MB) ha lo stesso effetto, poiché dimensioni maggiori portano a ritardi di trasmissione più elevati e una propagazione insufficiente.

Inoltre, il meccanismo PoW di Nakamoto causa enorme consumo di energia: una transazione Bitcoin in media consuma 431 KWh di elettricità.

::: Il Consenso Nakamoto (6/7)



Attacco Eclisse: Se un potente attacco riesce a dominare la comunicazione in entrata/uscita tra un miner vittima e la rete principale, allora la vittima non sarà più in grado di contribuire all'estensione della catena principale.

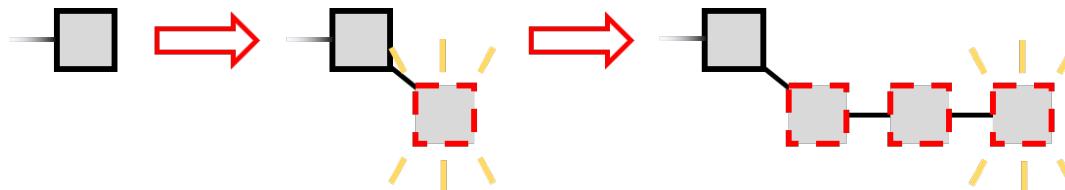
Se il potere di hashing dell'attaccante è α , allora un attacco double-spending è possibile se $\alpha + \varepsilon > 50\%$, con ε percentuali di miners eclissati.

L'attacco Eclisse è un exploit della debole connettività di una rete peer-to-peer senza autorizzazione basata su Internet. Per risolvere bisogna aumentare la connettività e la diversità geografica delle connessioni peer-to-peer.

::: Il Consenso Nakamoto (7/7)

Selfish Mining: Se un gruppo di miners maligni trattiene i blocchi appena estratti e li pubblicizza strategicamente per interrompere la propagazione dei blocchi estratti da miner onesti, può parzialmente annullare il lavoro di miner onesti e amplificare il loro potere di mining.

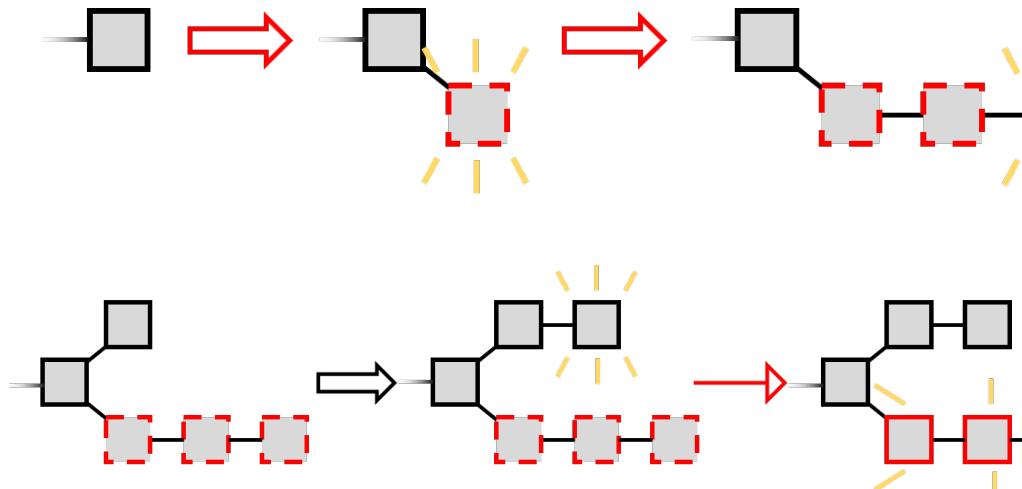
::: Il Consenso Nakamoto (7/7)



I blocchi non sono pubblicizzati ma tenuti segreti tra i miners selfish.

Selfish Mining: Se un gruppo di miners maligni trattiene i blocchi appena estratti e li pubblica strategicamente per interrompere la propagazione dei blocchi estratti da miner onesti, può parzialmente annullare il lavoro di miner onesti e amplificare il loro potere di mining.

::.. Il Consenso Nakamoto (7/7)

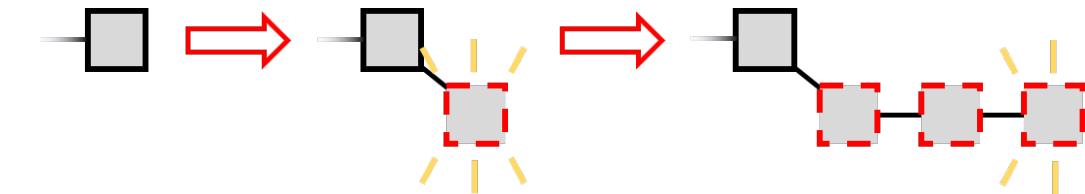


Gli altri miners estendono la catena con blocchi validi. Il miner egoista continua a estendere il suo ramo segreto fino a quando la catena pubblica è un passo indietro. Quindi la pubblica.

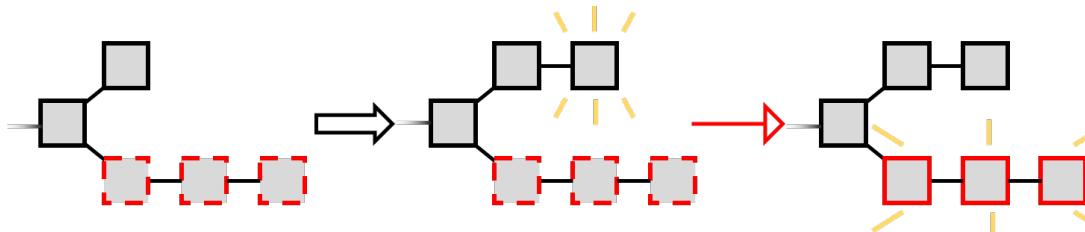
Selfish Mining: Se un gruppo di miners maligni trattiene i blocchi appena estratti e li pubblicizza strategicamente per interrompere la propagazione dei blocchi estratti da miner onesti, può parzialmente annullare il lavoro di miner onesti e amplificare il loro potere di mining.

Poiché la catena segreta è più lunga, le altre parti la considerano la catena principale, quindi ora tutti stanno seguendo i blocchi del minatore egoista. I blocchi generati dagli altri minatori vengono così eliminate.

::: Il Consenso Nakamoto (7/7)

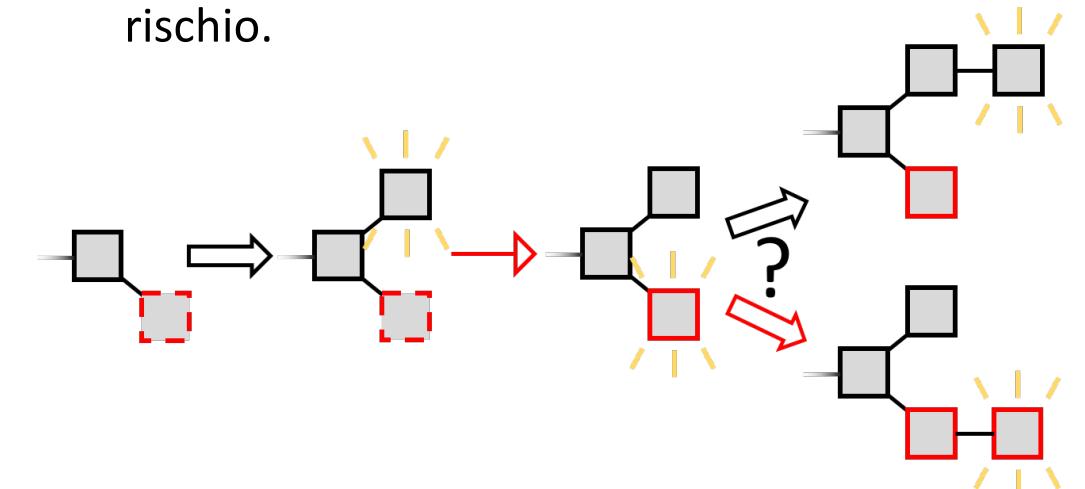


Selfish Mining: Se un gruppo di miners maligni trattiene i blocchi appena estratti e li pubblicizza strategicamente per interrompere la propagazione dei blocchi estratti da miner onesti, può parzialmente annullare il lavoro di miner onesti e amplificare il loro potere di mining.

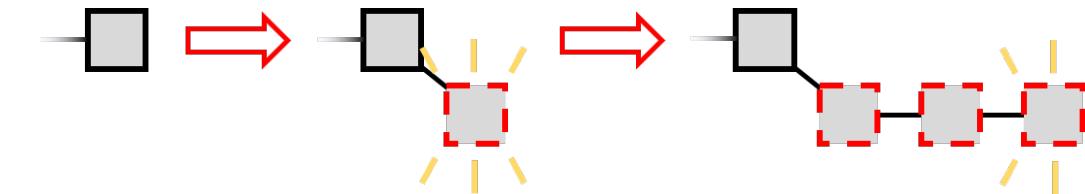


Quando forma per la prima volta la sua catena segreta, il minatore egoista corre un rischio.

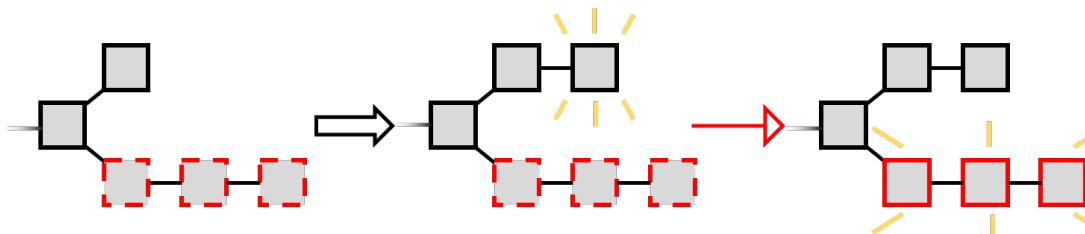
Se ha generato il primo blocco segreto e poi un altro miner ha generato un blocco, non può pubblicare il suo blocco segreto e avere la catena più lunga: si ha una corsa tra due rami di lunghezza uno.



::: Il Consenso Nakamoto (7/7)

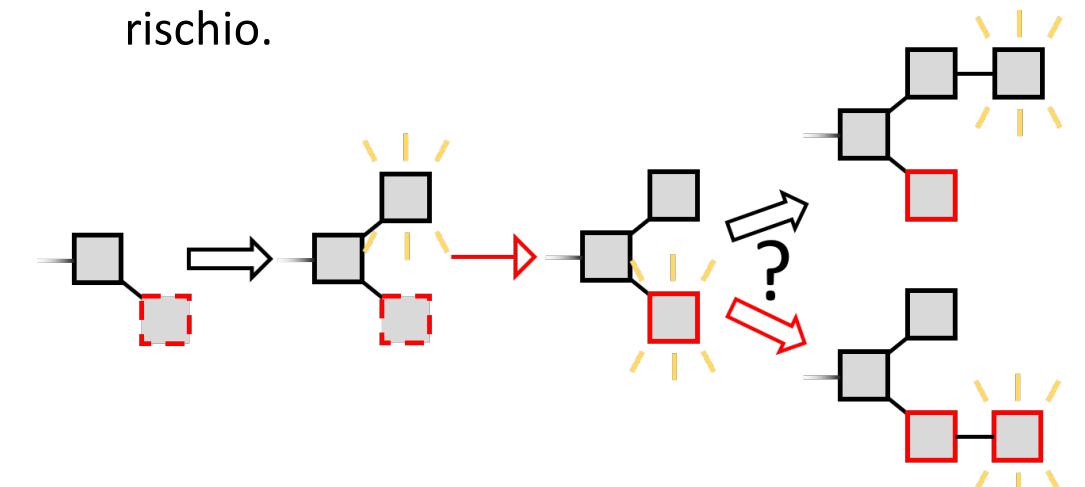


Selfish Mining: Se un gruppo di miners maligni trattiene i blocchi appena estratti e li pubblicizza strategicamente per interrompere la propagazione dei blocchi estratti da miner onesti, può parzialmente annullare il lavoro di miner onesti e amplificare il loro potere di mining.



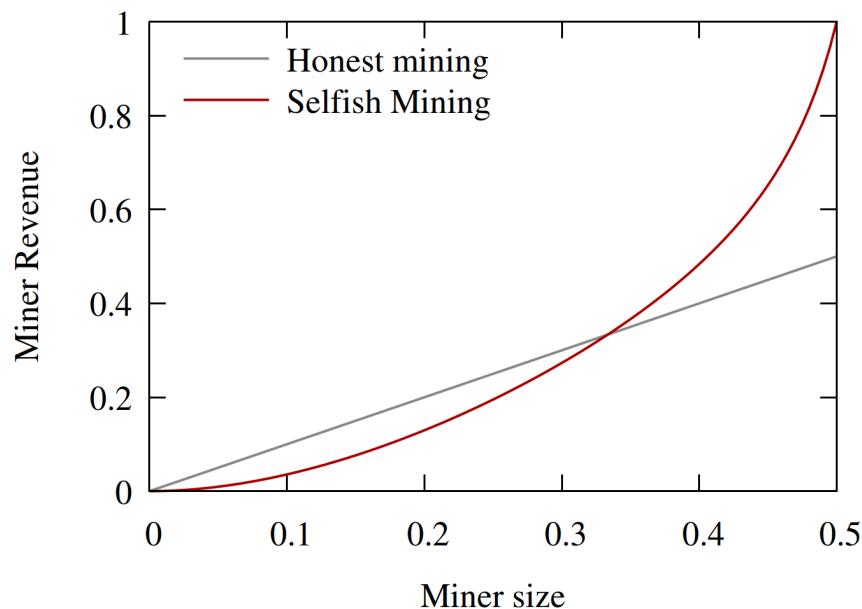
Quando forma per la prima volta la sua catena segreta, il minatore egoista corre un rischio.

Il miner egoista cercherà di estendere il proprio ramo come tutti gli altri miner. Se vince, pubblica la catena più lunga, e l'attacco riparte. Se vincono gli altri, il miner egoista è in svantaggio.



::: Il Consenso Nakamoto (7/7)

A prima vista potrebbe sembrare che l'attacco non funzioni: il miner di minoranza perderà più gare che vince. Tuttavia, un'attenta analisi mostra che non è così in generale.



Selfish Mining: Se un gruppo di miners maligni trattiene i blocchi appena estratti e li pubblicizza strategicamente per interrompere la propagazione dei blocchi estratti da miner onesti, può parzialmente annullare il lavoro di miner onesti e amplificare il loro potere di estrazione.

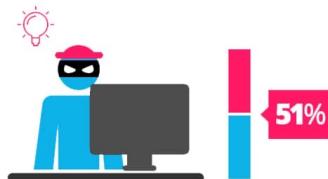
Un insieme di miner selfish più grande di $1/3$ della potenza di mining aumenterebbe le sue entrate deviando dal protocollo prescritto ed eseguendo Selfish Mining.

::: Proof-of-Stake (1/4)

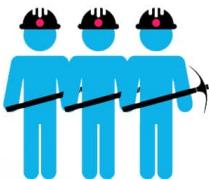
Proof of Work



To add each block to the chain, miners must compete to solve a difficult puzzle using their computers processing power.



In order to add a malicious block, you'd have to have a computer more powerful than 51% of the network.



The first miner to solve the puzzle is given a reward for their work.

vs.

Proof of Stake



There is no competition as the block creator is chosen by an algorithm based on the user's stake.



In order to add a malicious block, you'd have to own 51% of all the cryptocurrency on the network.



There is no reward for making a block, so the block creator takes a transaction fee.

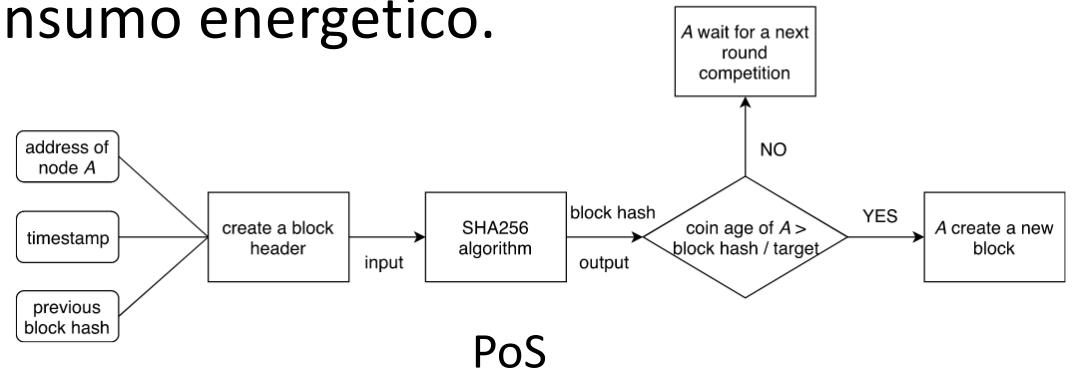
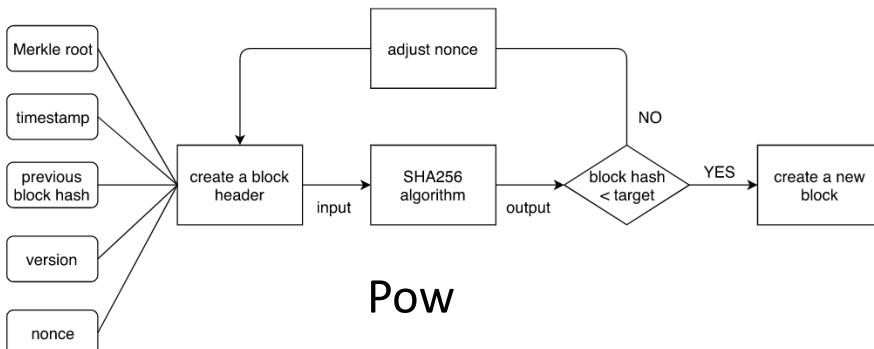
Proof-of-Stake (PoS) è un'alternativa efficiente dal punto di vista energetico al PoW.

Uno Stake o puntata si riferisce alle monete o token posseduti da un partecipante che possono essere investiti nel processo di consenso.

Rispetto a PoW la cui possibilità di proporre un blocco è proporzionale alla sua potenza di calcolo, la possibilità di proporre un blocco per PoS è proporzionale al valore del suo stake.

::: Proof-of-Stake (2/4)

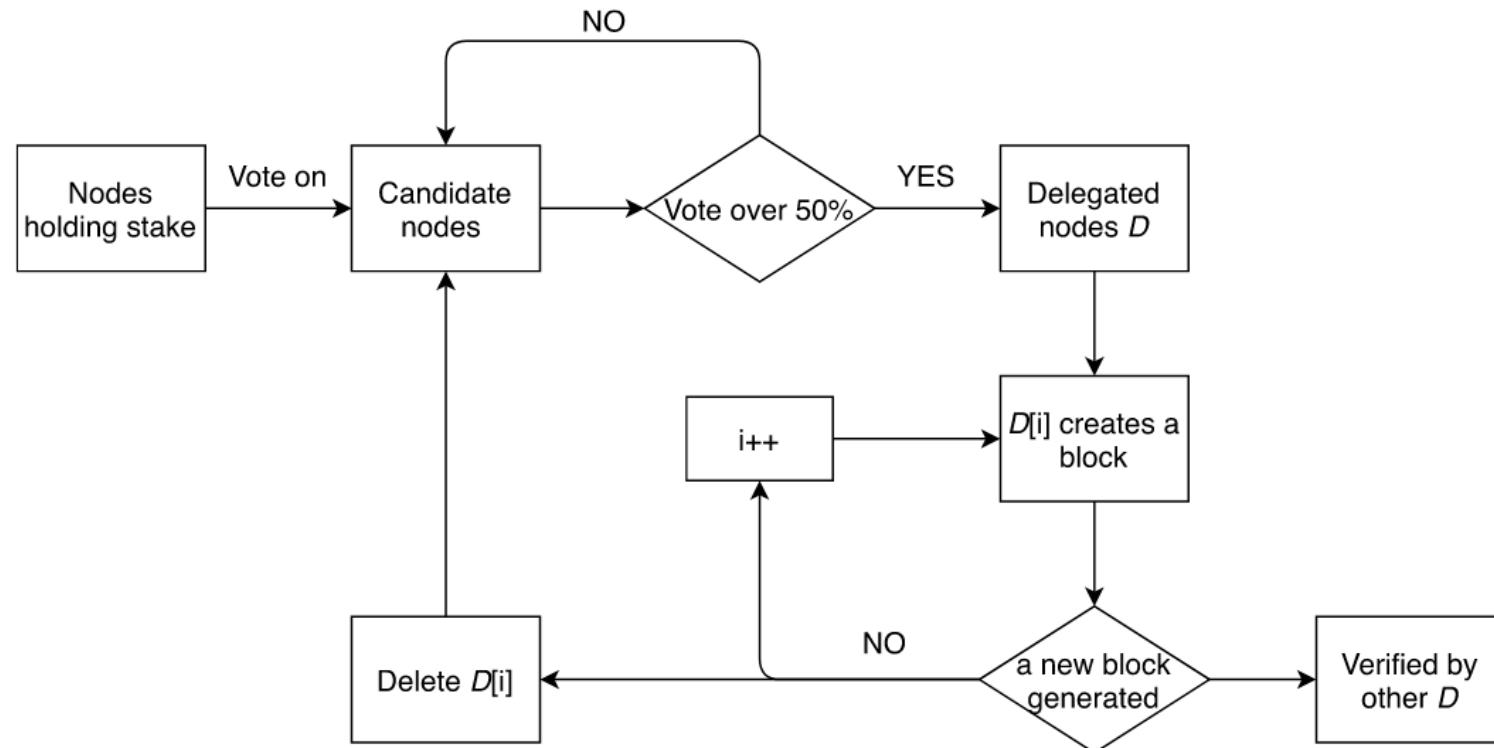
PoS non si basa sull'hashing dispendioso per generare blocchi. Poiché la difficoltà dell'hashing puzzle diminuisce con il valore dello stake del miner, il numero atteso di tentativi di hashing per un minter per risolvere il puzzle può essere significativamente ridotto se il suo valore di stake è alto. Pertanto, PoS evita la competizione di hashing brute-force che si verificherebbe se fosse stato usato PoW ottenendo così una significativa riduzione del consumo energetico.



Questo implica l'assenza del premio per chi inserisce il blocco giusto nella blockchain, e del mining, in quanto non vengono create nuove unità di criptovaluta con la creazione di ogni blocco. I validatori sono ricompensati con una commissione per le transazioni validate.

::: Proof-of-Stake (3/4)

Proof of Stake delegato (DPoS): consente ai nodi che detengono lo stake maggiore di votare per eleggere i verificatori di blocchi. Questo fa sì che i detentori di stake concedano il diritto di creare blocchi ai delegati che sostengono invece di creare blocchi stessi, riducendo così il loro consumo di potenza computazionale a 0.



::: Proof-of-Stake (4/4)

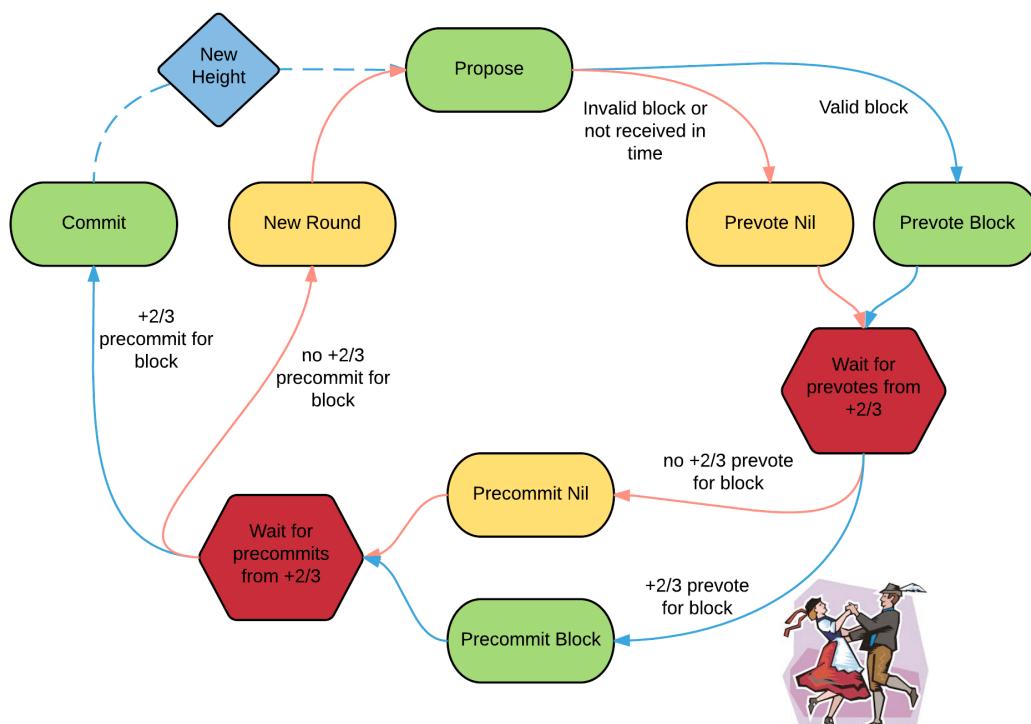
PoW avvantaggia i miners che hanno investito maggiormente in hardware, PoS dà una influenza sproporzionata a coloro che posseggono un numero importante di criptovalute, con il rischio di un accentramento di ricchezza nelle mani di pochi, seguendo il dogma secondo cui «Rich people get richer».

Un altro problema è “nothing at stake” per il quale nel caso di una fork del network i validatori saranno incentivati ad operare su entrambe le catene, risultando eventualmente in problemi di double-spending — questo problema è meno evidente in un sistema di DPoS.

PoS e DPoS sono stati applicati nel contesto di classici algoritmi BFT, come PBFT, al fine di consentirne l'applicazione in contesti open e permissionless.

::: Tendermint

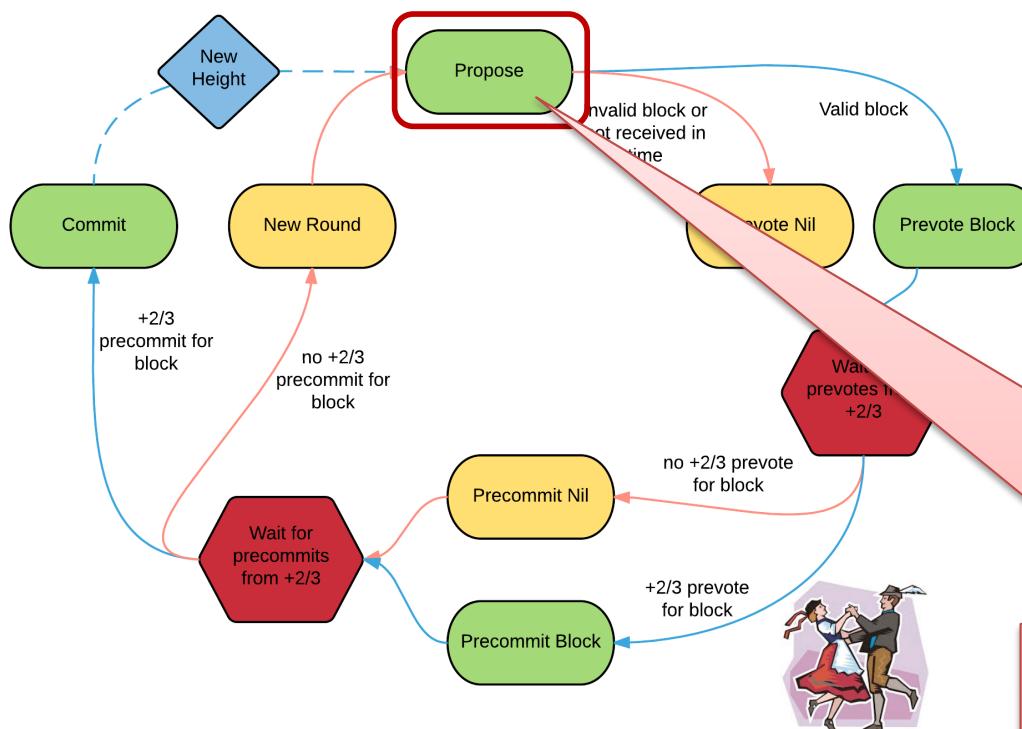
È un algoritmo di consenso ispirato a PBFT che sfrutta la PoS per il consenso in un ambiente permissioned, dove il gruppo dei validatori è un gruppo chiuso e precostituito.



Ogni nuovo blocco viene validato o meno in una iterazione dell'algoritmo, che si compone di molti round per poter giungere a consenso. Si tratta di un consenso di tipo CP, dove la consistenza viene fortemente garantita.

::: Tendermint

È un algoritmo di consenso ispirato a PBFT che sfrutta la PoS per il consenso in un ambiente permissioned, dove il gruppo dei validatori è un gruppo chiuso e precostituito.

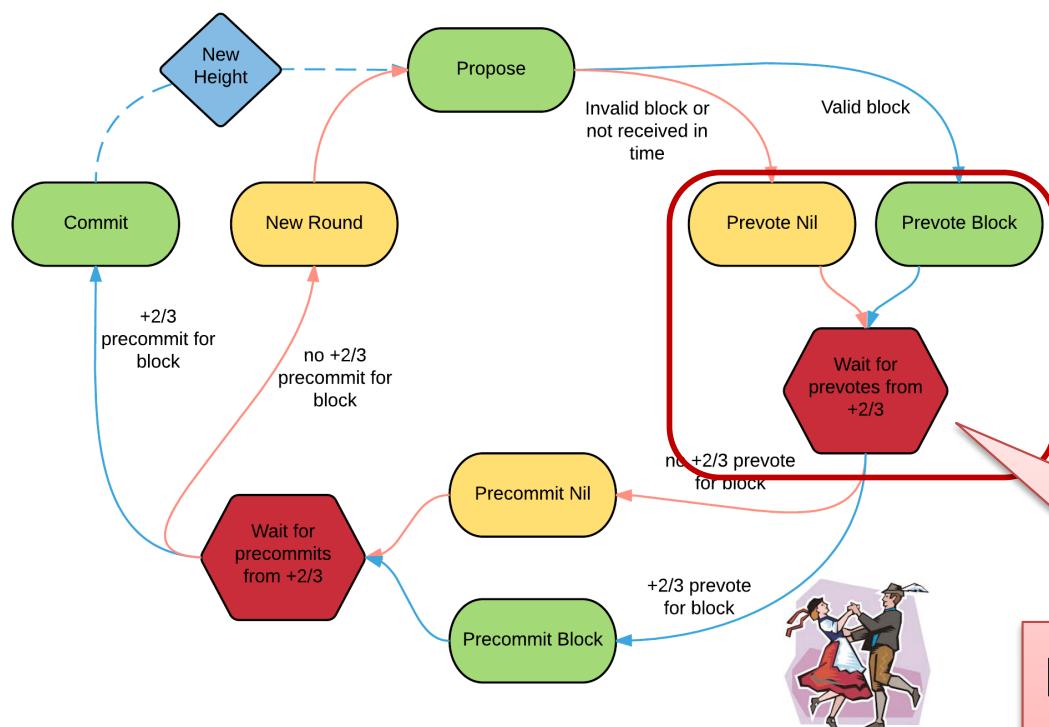


Ogni nuovo blocco viene validato o meno in una iterazione dell'algoritmo, che si compone di molti round per poter giungere a consenso. Si tratta di un consenso di tipo CP, cioè la consistenza viene fortemente garantita.

Inizialmente, viene scelto a caso un validatore che propone un nuovo blocco.

::: Tendermint

È un algoritmo di consenso ispirato a PBFT che sfrutta la PoS per il consenso in un ambiente permissioned, dove il gruppo dei validatori è un gruppo chiuso e precostituito.

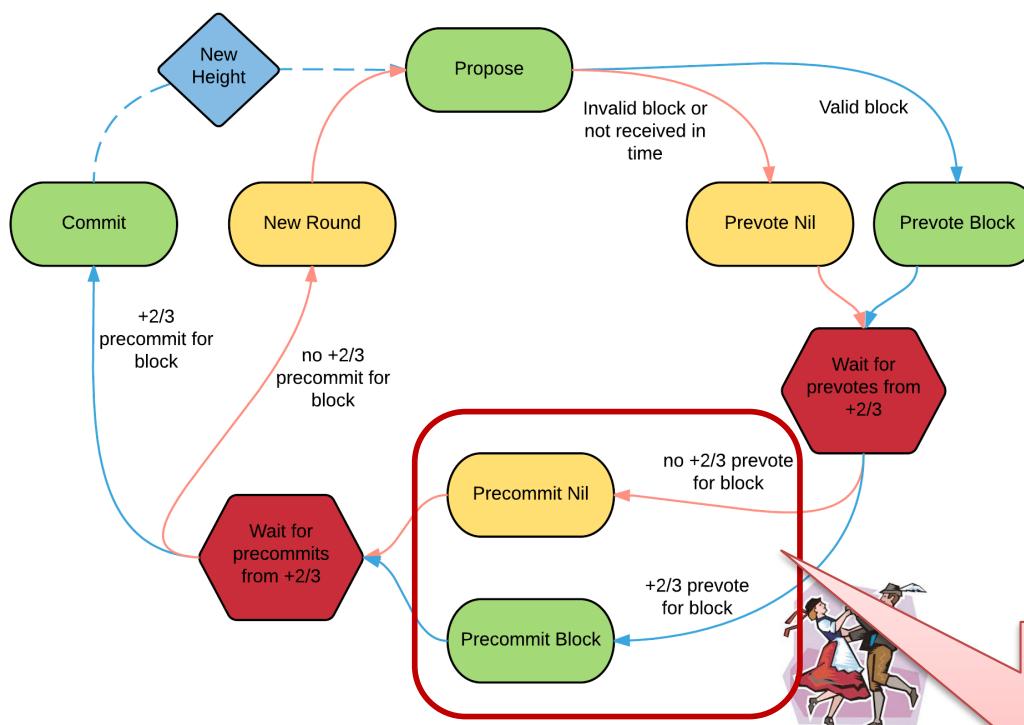


Ogni nuovo blocco viene validato o meno in una iterazione dell'algoritmo, che si compone di molti round per poter giungere a consenso. Si tratta di un consenso di tipo CP, dove la consistenza viene formalmente garantita.

Il nuovo blocco viene distribuito agli altri validatori che ne verificano la validità, e ne votano l'accettazione.

::: Tendermint

È un algoritmo di consenso ispirato a PBFT che sfrutta la PoS per il consenso in un ambiente permissioned, dove il gruppo dei validatori è un gruppo chiuso e precostituito.

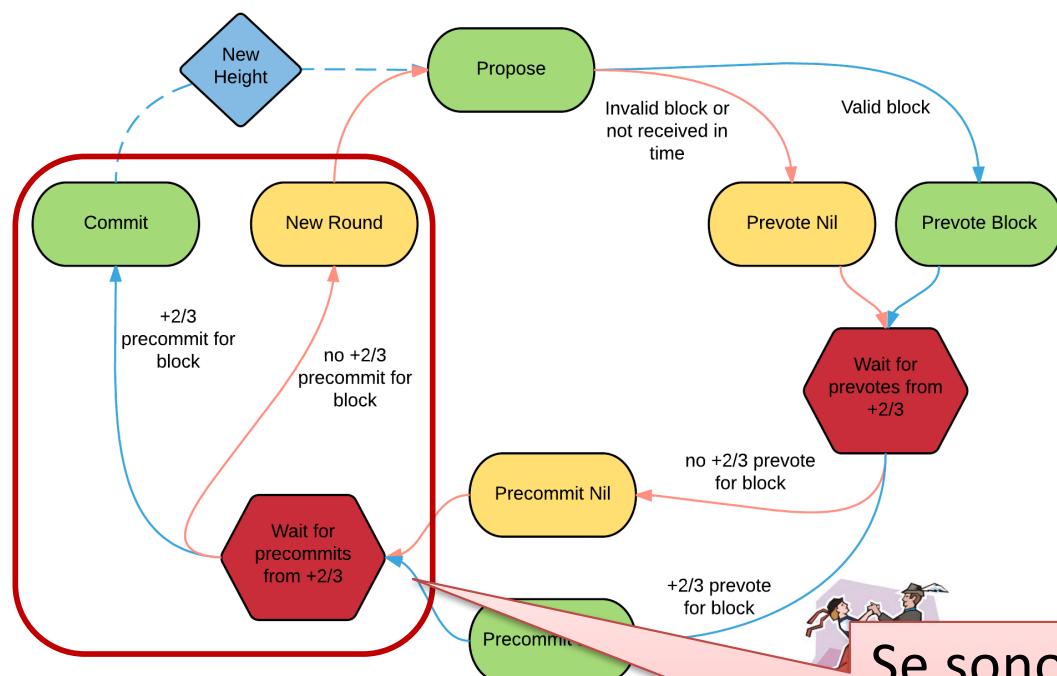


Ogni nuovo blocco viene validato o meno in una iterazione dell'algoritmo, che si compone di molti round per poter giungere a consenso. Si tratta di un consenso di tipo CP, dove la consistenza viene fortemente garantita.

Se sono arrivati i 2/3 ed oltre voti allora si manifesta l'intenzione di accettare il blocco, altrimenti di rigettarlo.

::: Tendermint

È un algoritmo di consenso ispirato a PBFT che sfrutta la PoS per il consenso in un ambiente permissioned, dove il gruppo dei validatori è un gruppo chiuso e precostituito.

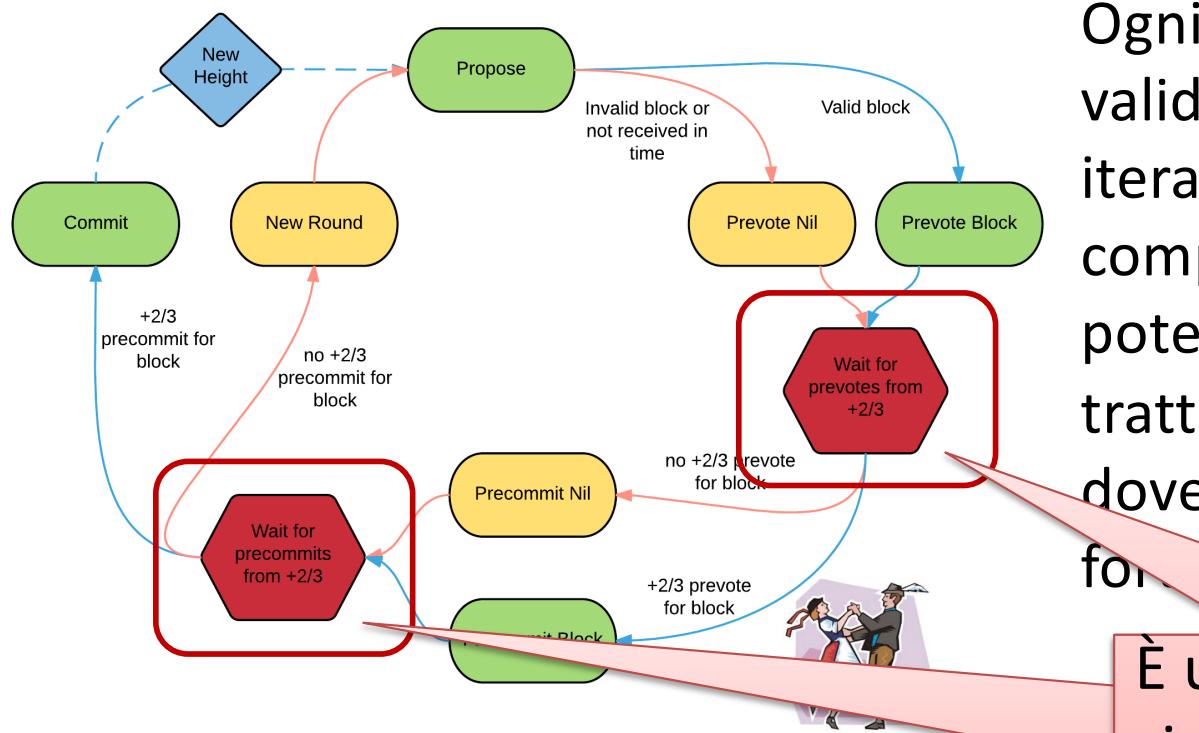


Ogni nuovo blocco viene validato o meno in una iterazione dell'algoritmo, che si compone di molti round per poter giungere a consenso. Si tratta di un consenso di tipo CP, dove la consistenza viene fortemente garantita.

Se sono arrivati i 2/3 dei voti ed oltre allora si conferma la decisione e si passa ad un'altra iterazione con un nuovo validatore, altrimenti si realizza un nuovo round.

::: Tendermint

È un algoritmo di consenso ispirato a PBFT che sfrutta la PoS per il consenso in un ambiente permissioned, dove il gruppo dei validatori è un gruppo chiuso e precostituito.

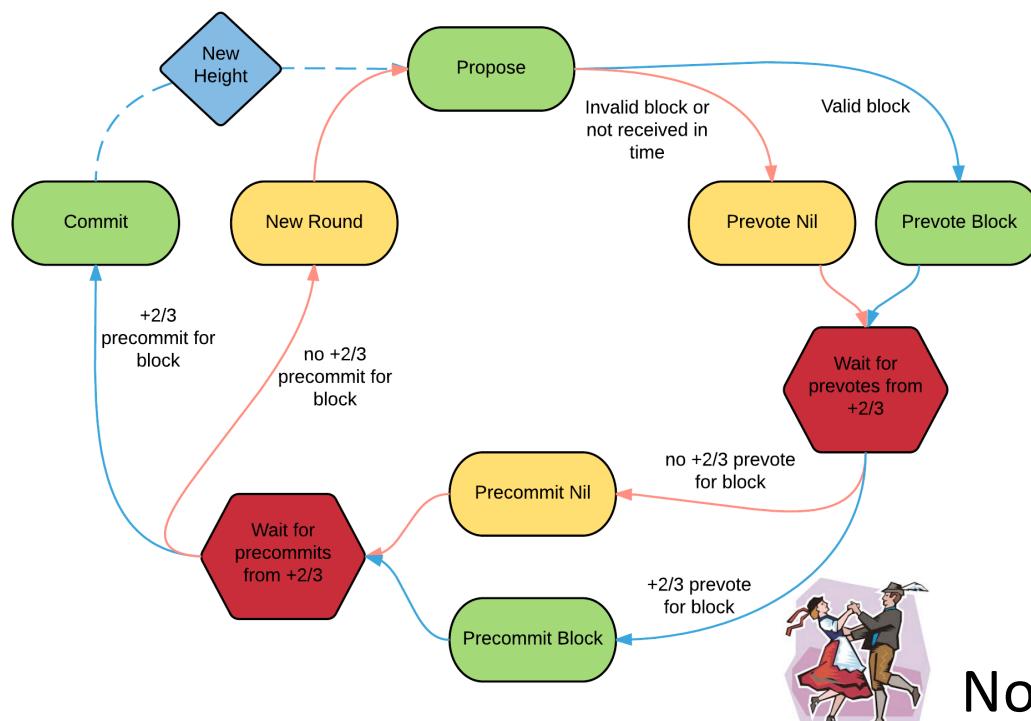


Ogni nuovo blocco viene validato o meno in una iterazione dell'algoritmo, che si compone di molti round per poter giungere a consenso. Si tratta di un consenso di tipo CP, dove la consistenza viene formalmente garantita.

È un algoritmo parzialmente sincrono, quindi si smette di aspettare dopo che è trascorso un timeout.

::: Tendermint

È un algoritmo di consenso ispirato a PBFT che sfrutta la PoS per il consenso in un ambiente permissioned, dove il gruppo dei validatori è un gruppo chiuso e precostituito.



Ogni nuovo blocco viene validato o meno in una iterazione dell'algoritmo, che si compone di molti round per poter giungere a consenso. Si tratta di un consenso di tipo CP, dove la consistenza viene fortemente garantita.

Non tutti i validatori avranno lo stesso "peso", ma la PoS viene usata per dare peso maggiore a chi ha uno stake maggiore. La condizione di $2/3$ non è sul numero di votanti ma sulla quantità di criptovaluta totale nel sistema.



Scalabilità nelle blockchain

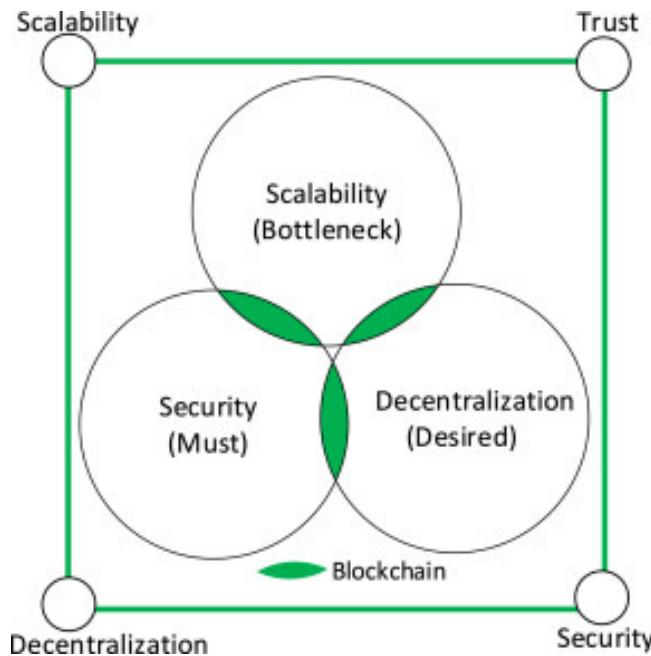
::: Blockchain Quadrilemma/Trilemma

Nonostante il successo e la forza, la scalabilità è la sfida principale che ostacola la piena adozione della blockchain in alcune aree.

- Le blockchain hanno basse prestazioni di throughput e latenza rispetto ai sistemi non blockchain. Ad esempio, il throughput delle blockchain di Bitcoin ed Ethereum è rispettivamente di 3-4 e 15 transazioni al secondo (TPS). In confronto, Visa e PayPal raggiungono rispettivamente 1667 e 193 TPS.
- Oltre ai suoi enormi dati di archiviazione, anche le prestazioni di lettura dei server blockchain sono basse rispetto a quelle dei server non blockchain come YouTube e Google.

Ci sono diversi sforzi e proposte per migliorare la scalabilità della blockchain. Tuttavia, è difficile risolvere i problemi di scalabilità della blockchain senza compromettere la sicurezza, la decentralizzazione o la fiducia della blockchain. C'è sempre un compromesso tra sicurezza, scalabilità, decentralizzazione e fiducia nella blockchain.

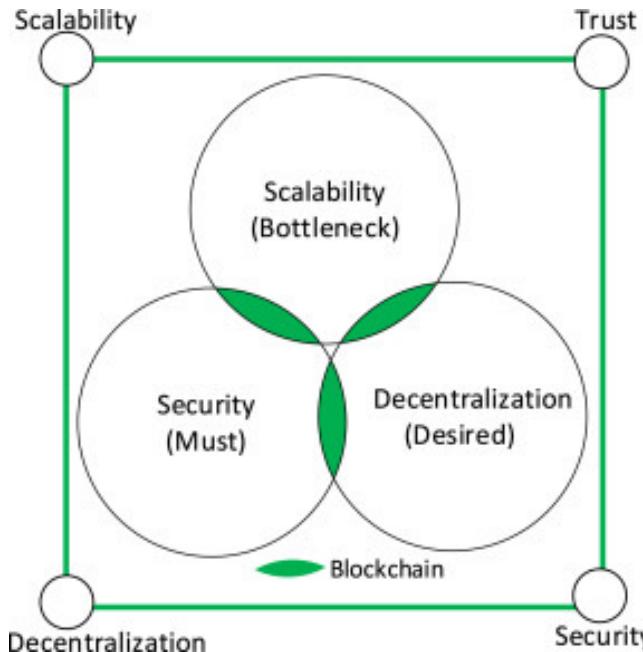
::: Blockchain Quadrilemma/Trilemma



Il Trilemma della blockchain (simile al teorema CAP) significa che la scalabilità, la decentralizzazione e la sicurezza della blockchain non possono coesistere perfettamente allo stesso tempo senza compromettere una di esse. D'altro canto, la fiducia è molto critica per la scalabilità della blockchain. Tuttavia, c'è anche un compromesso tra fiducia e decentralizzazione.

Le blockchain con parti fideate possono adottare un consenso, comunicazioni e calcoli meno complessi per ottenere una maggiore scalabilità. Quindi, il trilemma della scalabilità della blockchain viene spesso esteso al quadrilemma con l'aggiunta della fiducia.

::: Blockchain Quadrilemma/Trilemma



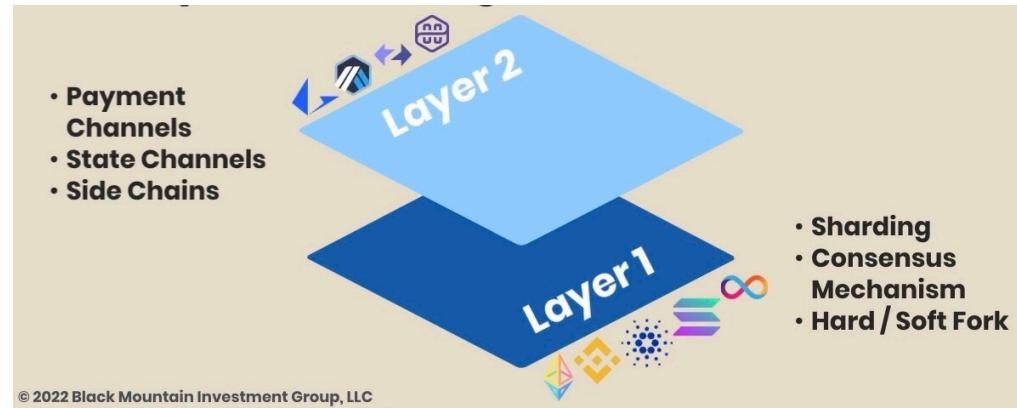
Il Quadrilemma della scalabilità della blockchain è il compromesso che esiste tra scalabilità, decentralizzazione, sicurezza e fiducia negli attuali sistemi blockchain in aggiunta al trilemma della blockchain.

È molto difficile ottenere queste quattro proprietà contemporaneamente nell'attuale blockchain.

Ad esempio, sicurezza e scalabilità sono ottenute in blockchain private e consorziali che hanno parti completamente affidabili ma sono completamente o parzialmente centralizzate. Scalabilità e decentralizzazione sono ottenute in blockchain basate su DAG che sono meno sicure e con meno fiducia. D'altro canto, le blockchain pubbliche hanno una buona sicurezza e decentralizzazione ma la loro scalabilità è debole.

::: Soluzioni di Efficientamento

Le soluzioni di efficientamento della scalabilità della blockchain si dividono in due classi:

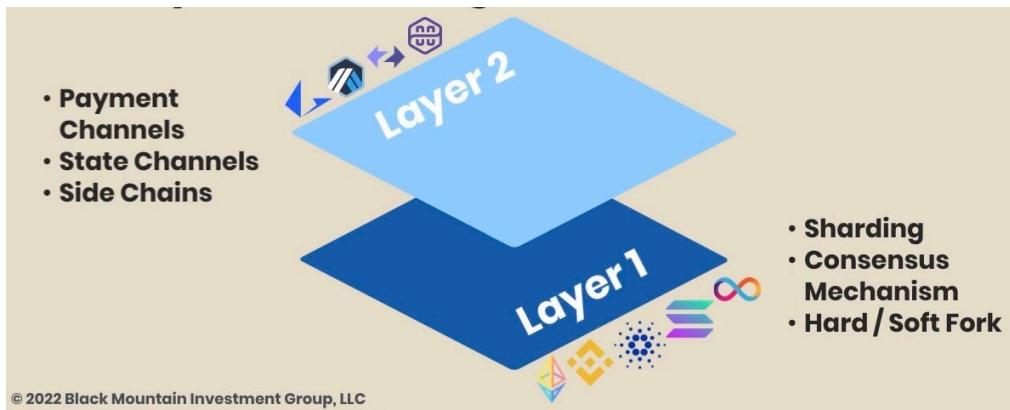


- Layer 1 include soluzioni di ottimizzazione nella blockchain
- Layer 2 sono soluzioni in cui si hanno soluzioni parallele alla catena principale, creando un secondo livello di transazioni che non sono archiviate sulla blockchain principale.

Le soluzioni Offchain si riferiscono a metodi e tecniche utilizzati per elaborare transazioni o calcoli al di fuori della rete blockchain principale. Queste soluzioni mirano a migliorare l'efficienza, la scalabilità e le prestazioni dei sistemi blockchain riducendo il carico sulla catena principale.

::: Soluzioni di Efficientamento

Le soluzioni di efficientamento della scalabilità della blockchain si dividono in due classi:



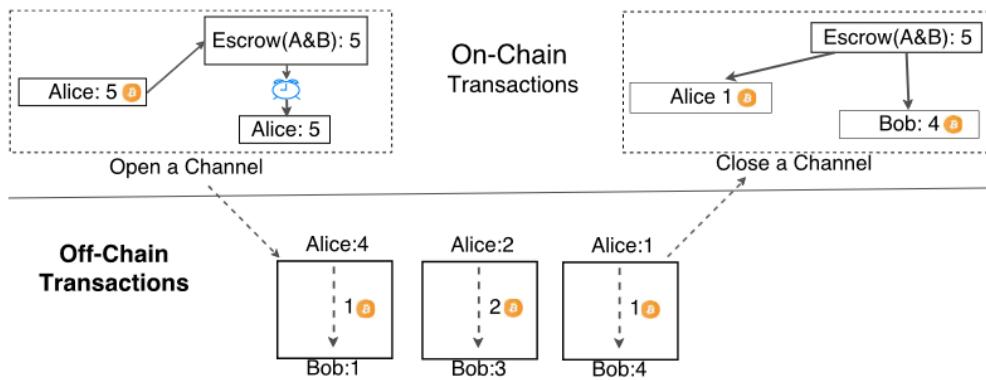
- Layer 1 include soluzioni di ottimizzazione nella blockchain
- Layer 2 sono soluzioni in cui si hanno soluzioni parallele alla catena principale, creando un secondo livello di transazioni che non sono archiviate sulla blockchain principale.

Una sidechain è una blockchain indipendente che è interoperabile con una blockchain principale. Consente la sperimentazione, funzionalità aggiuntive o diversi meccanismi di consenso senza influenzare la blockchain principale. Il peg bidirezionale assicura che gli asset possano essere spostati tra la catena principale e la sidechain.

::: OffChain

Tipi di soluzioni Offchain

- Calcolo/Transazioni Offchain: calcoli complessi o transazioni vengono eseguiti fuori dalla blockchain e solo i risultati, un riepilogo o stato finale vengono registrati on-chain.
- Archiviazione Offchain: i dati vengono archiviati fuori dalla blockchain per ridurre i costi di caricamento e archiviazione dei dati on-chain.
- Un canale di transazione consente di eseguire più pagamenti P2P diretti tra due parti senza impegnare ogni transazione nel registro condiviso Bitcoin (ad esempio, on-chain).

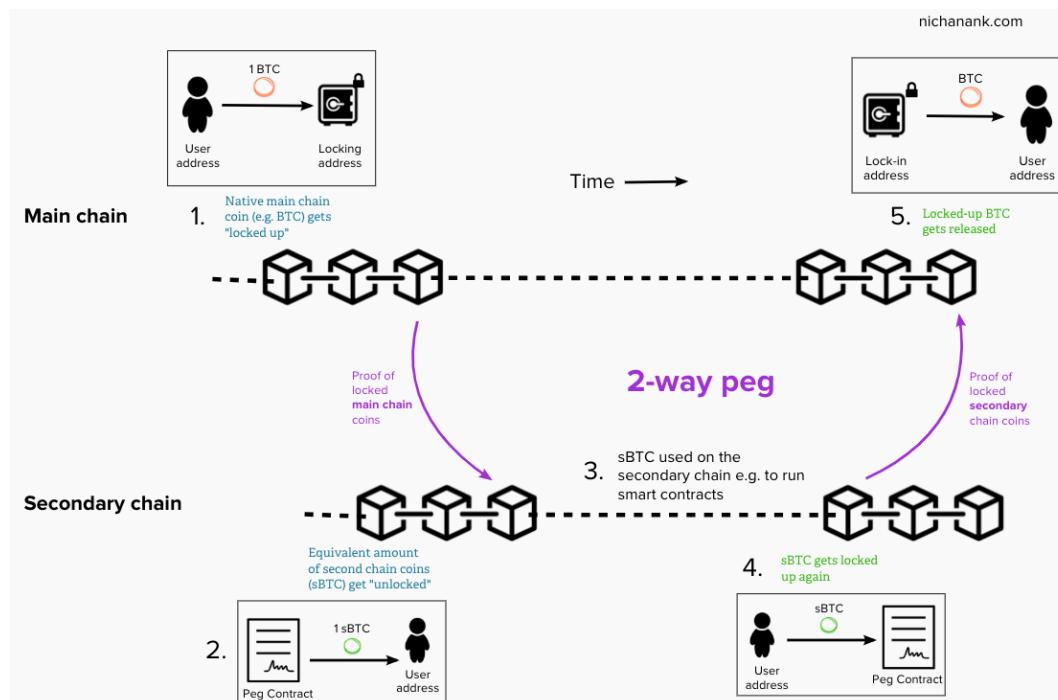


È possibile creare un canale off-chain unidirezionale verso un fornitore di addebito e può effettuare pagamenti fino a quando la capacità di questo canale non viene raggiunta senza pagare alcuna commissione di transazione Bitcoin.

::: SideChain

Come funzionano le sidechain?

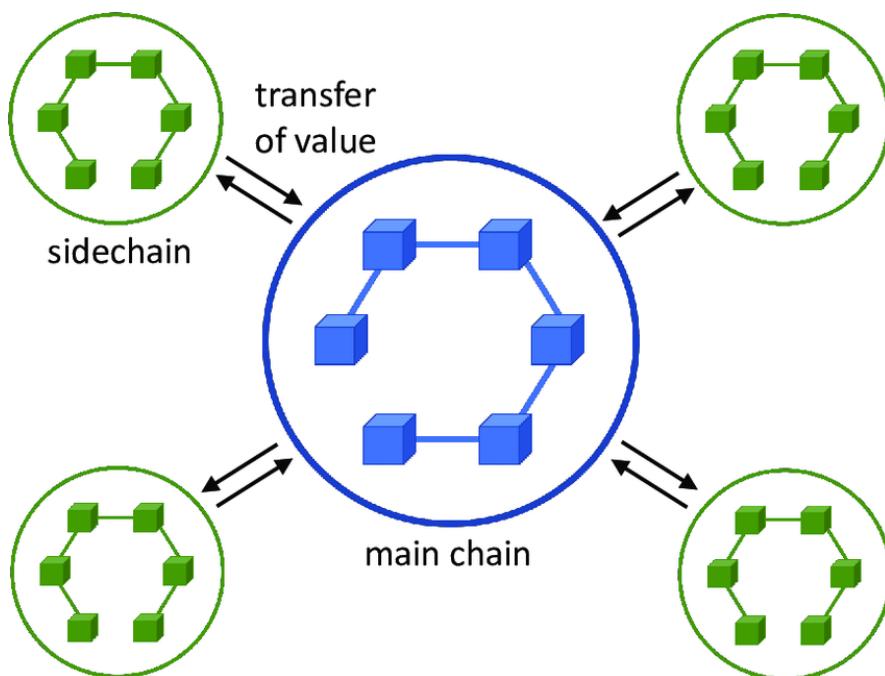
- Peg bidirezionale: è un meccanismo che blocca gli asset sulla catena principale e rilascia asset equivalenti sulla sidechain. Quando gli asset vengono spostati di nuovo sulla catena principale, gli asset della sidechain vengono bruciati o distrutti e gli asset originali vengono sbloccati.



::: SideChain

Come funzionano le sidechain?

- Funzionamento indipendente: le sidechain possono avere i propri algoritmi di consenso, regole e protocolli, consentendo operazioni e sperimentazioni personalizzate. Possono elaborare transazioni ed eseguire contratti intelligenti senza influire sulla catena principale.



- Sincronizzazione periodica: le sidechain comunicano periodicamente con la blockchain principale per sincronizzare lo stato e garantire la coerenza.

::: SideChain

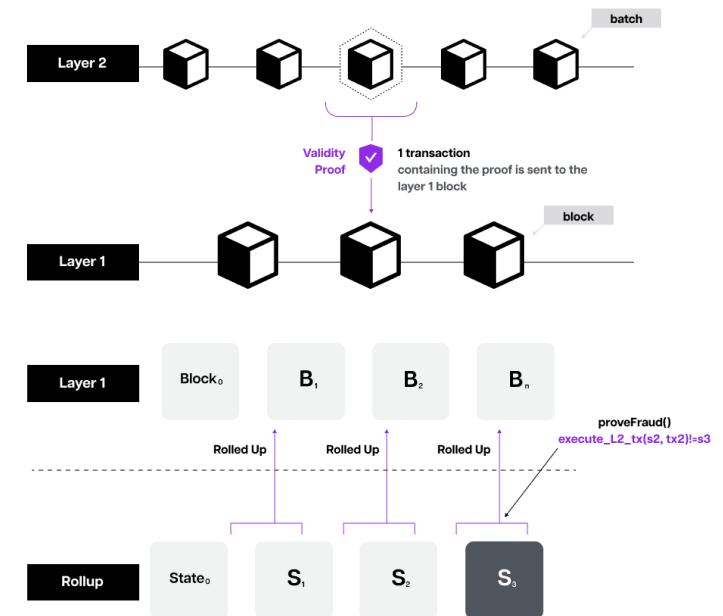
Tipi di sidechain

- Sidechain federate: queste sidechain sono gestite da un consorzio o federazione di entità affidabili anziché da una rete decentralizzata. La federazione controlla il consenso (con approcci federati e decentralizzati) e la governance della sidechain.
- Sidechain autorizzate: le sidechain autorizzate limitano l'accesso a un set pre-approvato di partecipanti o entità. Operano in un ambiente controllato con diritti di accesso specifici. Sidechain pubbliche: le sidechain pubbliche sono aperte a chiunque voglia unirsi e partecipare, in modo simile alla blockchain principale. Traggono vantaggio dalla natura decentralizzata della catena principale, ma operano in modo indipendente.
- Rollup: i rollup sono un tipo di sidechain che esegue transazioni al di fuori della catena principale, ma pubblica periodicamente riepiloghi di queste transazioni sulla blockchain principale. Sono disponibili in due tipi principali: rollup ottimistici e ZK-rollup.

::: SideChain

- Rollup ottimistici: i rollup ottimistici presuppongono che le transazioni siano valide di default e le verificano solo in caso di controversia. Ciò riduce la quantità di calcolo richiesta ma anche introduce un meccanismo per la risoluzione delle controversie.
- Rollup a conoscenza zero (ZK): questi utilizzano prove crittografiche chiamate zk-SNARK o zk-STARK per dimostrare che le transazioni sono valide senza rivelare i dati effettivi. Ciò consente una verifica più rapida.

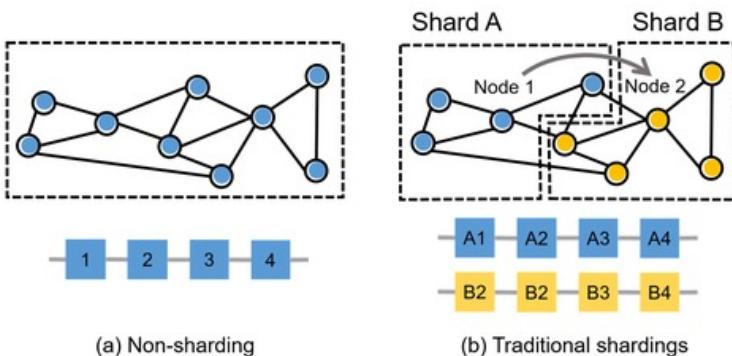
	Optimistic Rollup	ZK-Rollup
Realistic Throughput Cap	~500 TPS	>2000 TPS
Fund Withdrawal Period	1 - 2 weeks	A few minutes to hours
Privacy	Hard and expensive	Easy and cheap
EVM Compatability	Easier - Solidity code with minor changes	Harder - Solidity code must be adjusted
Layer-2 Computation Costs	Lower - Not very hardware intensive	Higher - expensive and hardware intensive
Transaction Validation	1 honest validator is required at all times	ZKP setup/audit required once



::: Sharding

Lo Sharding consente di elaborare molte più transazioni di quanto non sia possibile con i tradizionali meccanismi di consenso, riducendo le commissioni di transazione nel processo, poiché la concorrenza per lo spazio nel blocco successivo sarà ridotta.

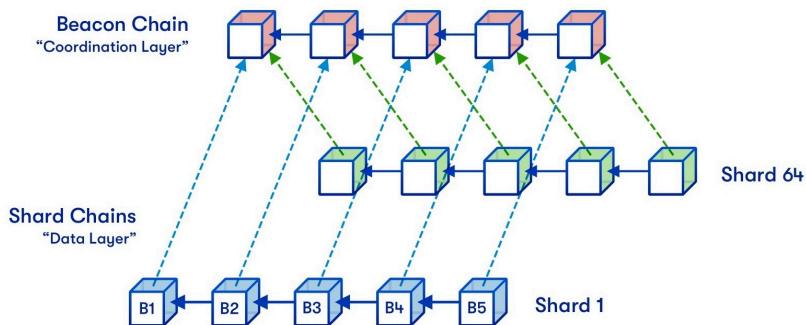
- Nel consenso tradizionale, tutti i nodi della rete devono verificare, scaricare e archiviare ogni transazione prima di elaborarne di nuove.
- Con lo sharding, si suddivide un blocco in porzioni o shard gestiti da blockchain semi-indipendenti, per condividere il carico. I validatori devono solo archiviare ed elaborare le transazioni sullo shard che stanno convalidando, non sull'intera rete.



::: Sharding

Lo Sharding consente di elaborare molte più transazioni di quanto non sia possibile con i tradizionali meccanismi di consenso, riducendo le commissioni di transazione nel processo, poiché la concorrenza per lo spazio nel blocco successivo sarà ridotta.

- Nel consenso tradizionale, tutti i nodi della rete devono verificare, scaricare e archiviare ogni transazione prima di elaborarne di nuove.
- Con lo sharding, si suddivide un blocco in porzioni o shard gestiti da blockchain semi-indipendenti, per condividere il carico. I validatori devono solo archiviare ed elaborare le transazioni sullo shard che stanno convalidando, non sull'intera rete.



Per preservare la sicurezza, ogni catena di shard invia a intervalli regolari un record di transazioni alla catena principale (Beacon Chain).

::: Sharding

Sebbene lo sharding prometta molti vantaggi, introduce nuovi problemi:

- Meno nodi validano ogni shard e le attività dannose, come gli attacchi del 51%, diventano più facili.
- Il codice è più complesso, aumentando il rischio di vulnerabilità della sicurezza del consenso e/o degli smart contracts.
- I membri del comitato possono colludere per inviare transazioni dannose alla catena principale.

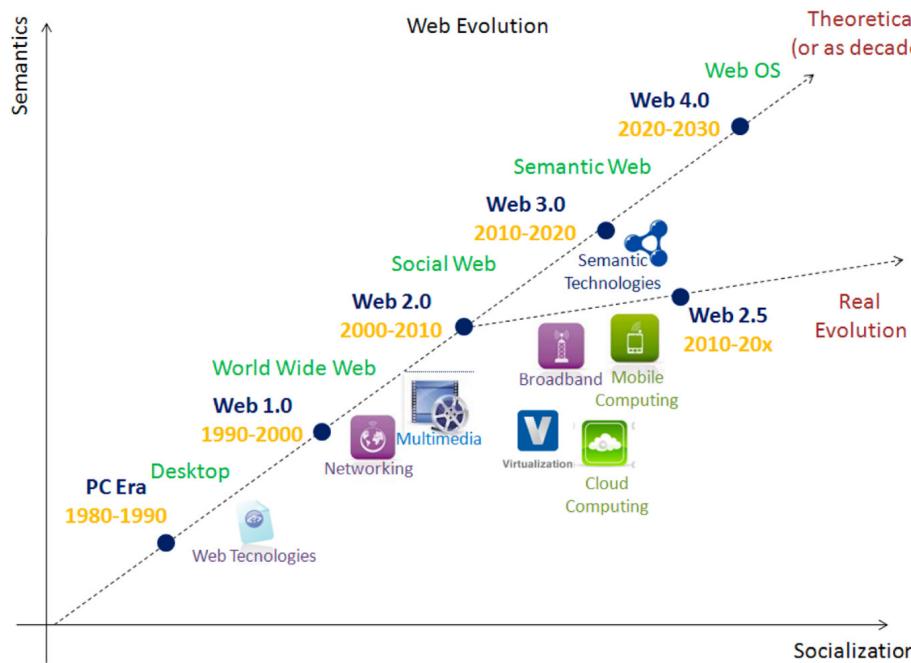
Meccanismi di protezione saranno integrati:

- Le prove di frode possono essere utilizzate per dimostrare la non-validità delle transazioni e punire attività disoneste.
- La partecipazione casuale ai comitati di ogni shard server per prevenire la collusione. Se i validatori non sanno per quale sotto-catena andranno a svolgere il consenso, diventa più difficile coordinare un attacco al sistema. In questo scenario, un attacco del 51% su una catena di shard diventa impossibile da eseguire.

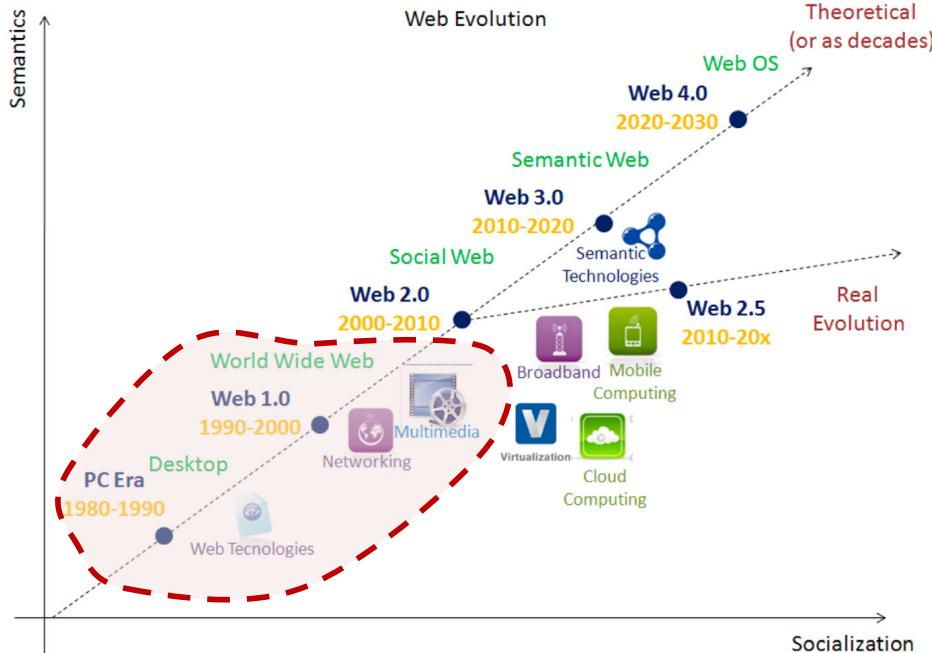


Web 3.0

::: Evoluzione Web (1/8)

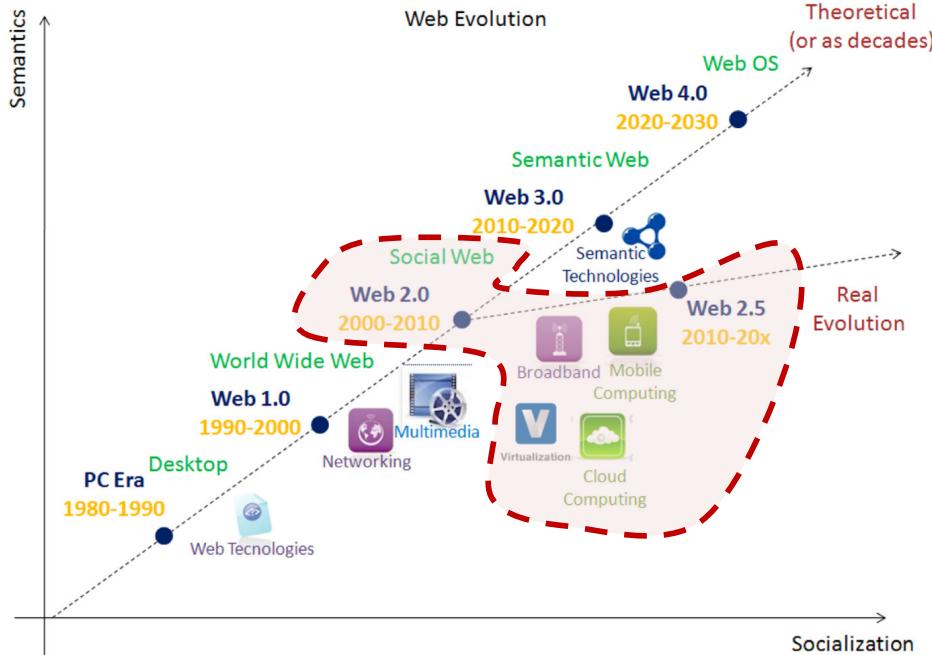


::: Evoluzione Web (1/8)



La prima versione di Internet, o Web 1.0, è stata progettata specificamente per le aziende piuttosto che per gli individui. Si trattava di una rete di distribuzione di contenuti (CDN) che consentiva agli utenti di visualizzare dati statici sui siti Web senza avere la possibilità di esprimere i propri pensieri, opinioni o osservazioni.

::: Evoluzione Web (1/8)



La prima versione di Internet, o Web 1.0, è stata progettata specificamente per le aziende piuttosto che per gli individui. Si trattava di una rete di distribuzione di contenuti (CDN) che consentiva agli utenti di visualizzare dati statici sui siti Web senza avere la possibilità di esprimere i propri pensieri, opinioni o osservazioni.

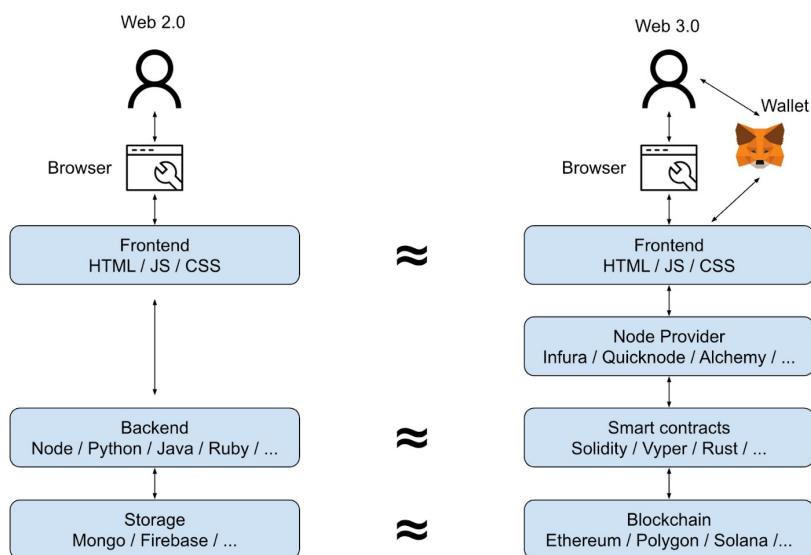
La prossima generazione di Internet, attualmente in uso in tutto il mondo, è il Web 2.0 che ha reso estremamente semplice per gli utenti raccogliere, generare e distribuire enormi quantità di dati sulle reti globali. I canali dei social media e altre applicazioni di streaming video sono tutti esempi di piattaforme Web 2.0.

::: Evoluzione Web (2/8)



Web 3.0 aspira a fornire un'interfaccia affidabile e basata sui dati che soddisfi ogni utente.

La principale distinzione tra Web 2.0 e Web 3.0 è il decentramento: gli utenti saranno proprietari dei propri contenuti e avranno il controllo completo sull'utilizzo di Internet. Il Web 2.0 si concentra sulla lettura e scrittura di contenuti, mentre il Web 3.0 sulla creazione di contenuti (Semantic Web) e sulla decentralizzazione delle applicazioni secondo un approccio P2P e non client-server..

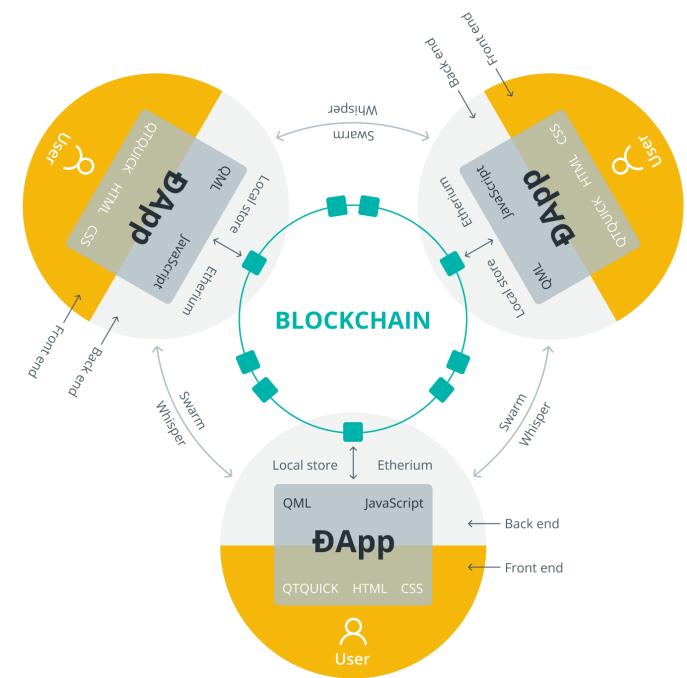
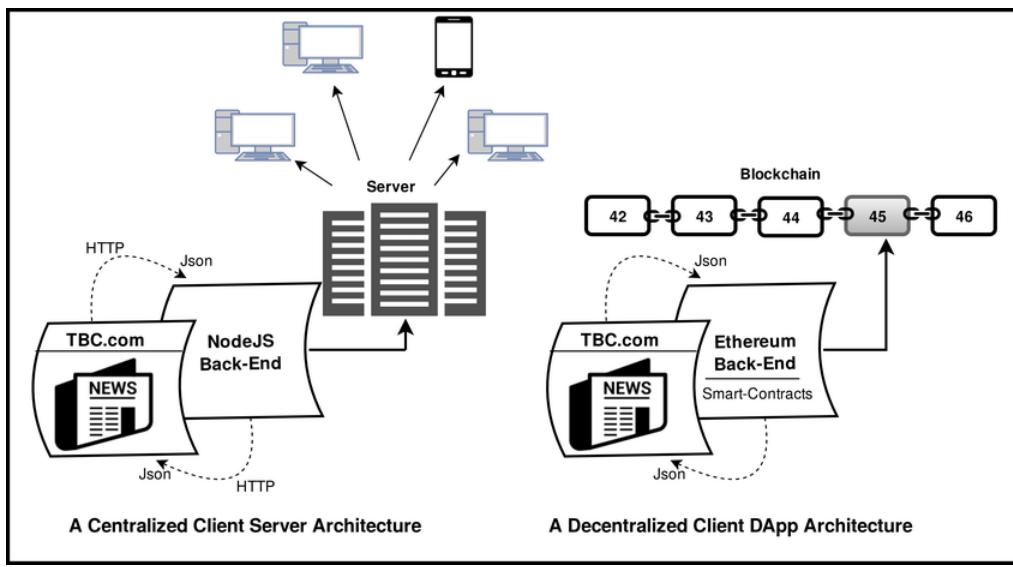


È possibile trovare una corrispondenza tra gli elementi nelle applicazioni Web 2.0 e quelle Web 3.0, dove la blockchain e gli smart contract sono il back-end.

Nel front-end troviamo i wallet che conservano i claim di accesso dei provider di blockchain.

::: Evoluzione Web (3/8)

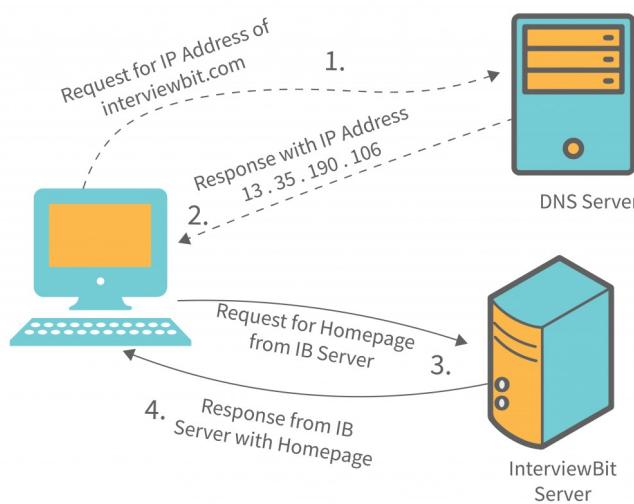
Le DApp sono applicazioni simili alle app tradizionali, con la differenza fondamentale che al posto di poggiarsi su server centralizzati sfruttano le piattaforme blockchain e il loro network distribuito. Un’ulteriore peculiarità è che una app tradizionale richiede i dati dell’utente per poter accedere ai suoi servizi, nelle DApp l’utente utilizza il proprio “account” blockchain, ossia le proprie chiavi crittografiche, evitando così di esplicitare i propri dati personali.



::: Evoluzione Web (4/8)

Nello sviluppo web tradizionale ci ritroviamo davanti un'architettura client-server, dove il client è una app o front-end che comunica con il server (back-end) mentre il server memorizza i dati ed espone dei servizi disponibili al client.

- Il client fornisce un'interfaccia per gli utenti così da richiedere servizi al server e visualizzare i risultati che esso restituisce;
- Il server attende le richieste dal client e risponde a queste, e le interazioni avvengono mediante il protocollo HTTP.

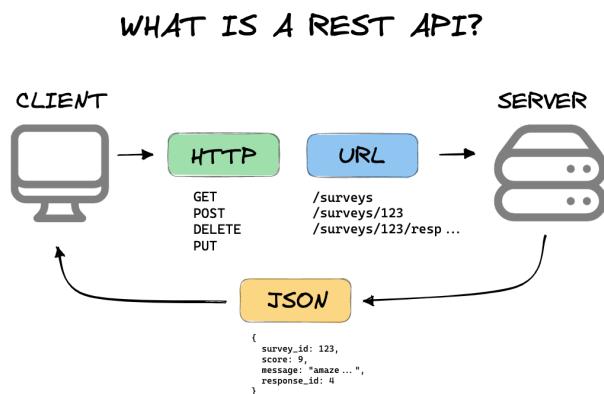


Quest'architettura è originata dall'idea della «separation of concerns»: separando l'interfaccia utente dalla memorizzazione dei dati viene migliorata la portabilità e la scalabilità, così che ogni componente può evolvere indipendentemente dagli altri. Inoltre, si riduce la complessità, si migliora l'effettività e si aumenta l'efficienza.

::: Evoluzione Web (5/8)

HTTP è il protocollo di riferimento per la comunicazione nel WWW e fornisce una sintassi/semantica che disciplina come le interazioni tra client e server avvengono attraverso vari metodi di richiesta.

SAFE METHODS	GET HEAD	HTTP/1.1 MUST IMPLEMENT THIS METHOD
NO ACTION ON SERVER		INSPECT RESOURCE HEADERS
MESSAGE WITH BODY	PUT POST PATCH TRACE OPTIONS DELETE	DEPOSIT DATA ON SERVER – INVERSE OF GET SEND INPUT DATA FOR PROCESSING PARTIALLY MODIFY A RESOURCE ECHO BACK RECEIVED MESSAGE SERVER CAPABILITIES DELETE A RESOURCE – NOT GUARANTEED
SEND DATA TO SERVER		

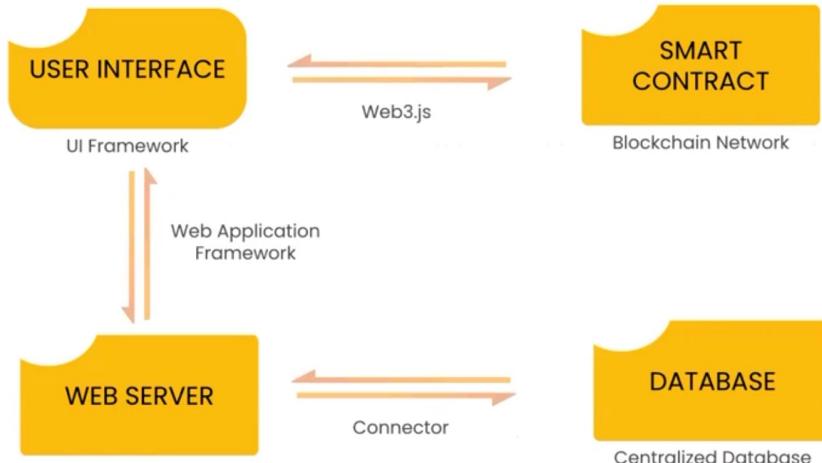


REST (Representational State Transfer) definisce un insieme di vincoli secondo cui il server processa e risponde alle richieste del client. Rappresenta un protocollo per lo scambio di ogni messaggio usando HTTP per il loro trasporto e JSON come modello dati.

Vengono identificate delle risorse che il servizio web offre mediante protocollo HTTP, i cui principali quattro metodi indicano quali operazioni il client potrà richiedere al server. Ogni richiesta verrà inviata ad un particolare indirizzo web e tale URL sarà univoco per la risorsa su cui richiedere l'azione e per l'operazione che si richiede su di essa.

::: Evoluzione Web (6/8)

Un'architettura di riferimento per una Dapp è leggermente diversa:



L'interfaccia utente dialoga con il web server, che interagisce con il database. – Generica architettura WEB.

La novità è rappresentata dalla possibilità per l'interfaccia di dialogare direttamente con uno smart contract.

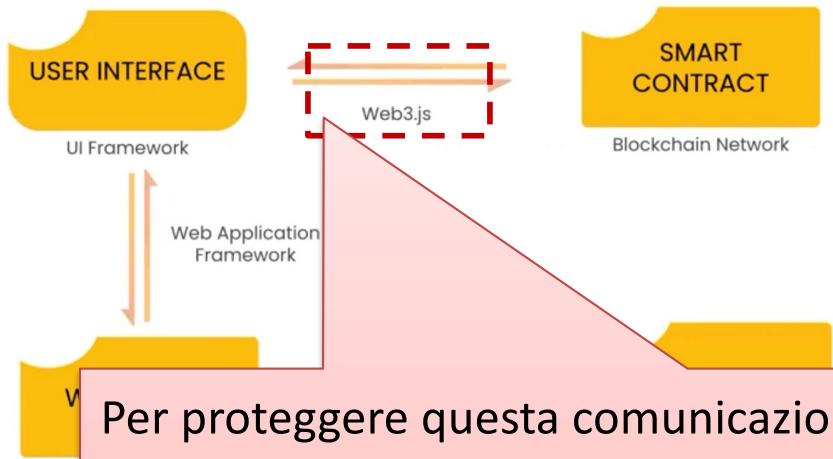
Tale novità è dovuta per motivi di sicurezza/privacy, per non inviare dati sensibili verso il web server che funge da intermediario, ma il client vuole dialogare direttamente con la blockchain.

Nel Web 3.0 la dipendenza con la blockchain è iniettata direttamente nello user interface, senza alcun brokering del web server.

Web3.js è il mezzo per l'interazione con gli smart contracts, ed è possibile usarlo solo nel browser e non in un ambiente Node. Ogni sincronizzazione tra back-end e smart contract avviene tramite il client.

::: Evoluzione Web (6/8)

Un'architettura di riferimento per una Dapp è leggermente diversa:

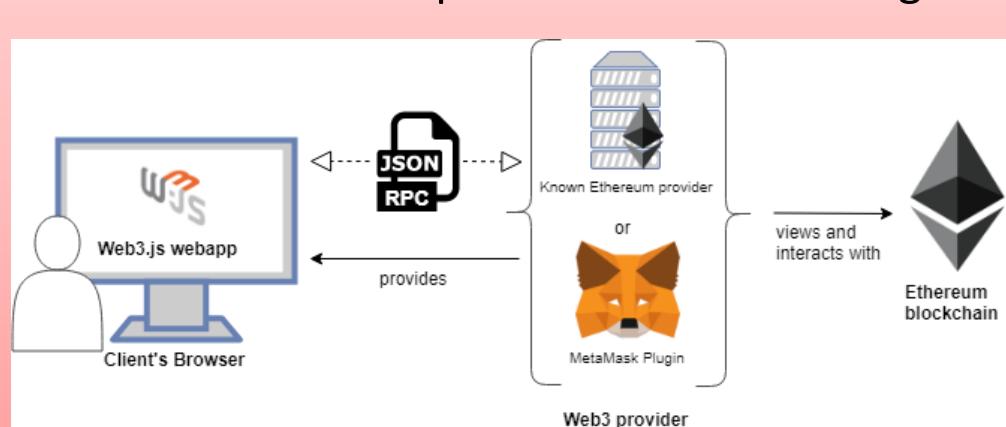


L'interfaccia utente dialoga con il web server, che interagisce con il database. – Generica architettura WEB.

La novità è rappresentata dalla possibilità per l'interfaccia di dialogare con il smart contract.

non inviare dati al server, ma il client

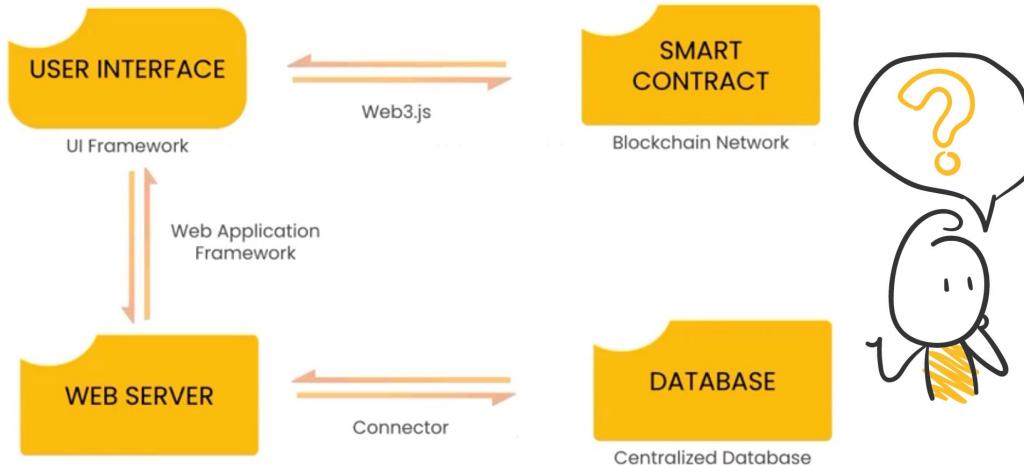
comunica direttamente con il server.



Web3.js è il mezzo per l'interazione con gli smart contracts, ed è possibile usarlo solo nel browser e non in un ambiente Node. Ogni sincronizzazione tra back-end e smart contract avviene tramite il client.

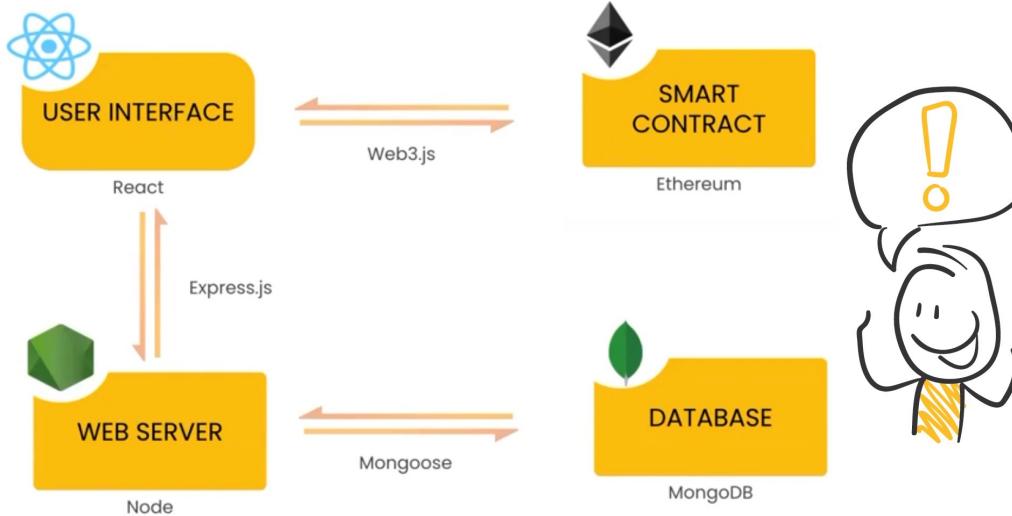
::: Evoluzione Web (7/8)

Esiste una implementazione di questa architettura?



::: Evoluzione Web (7/8)

Esiste una implementazione di questa architettura?



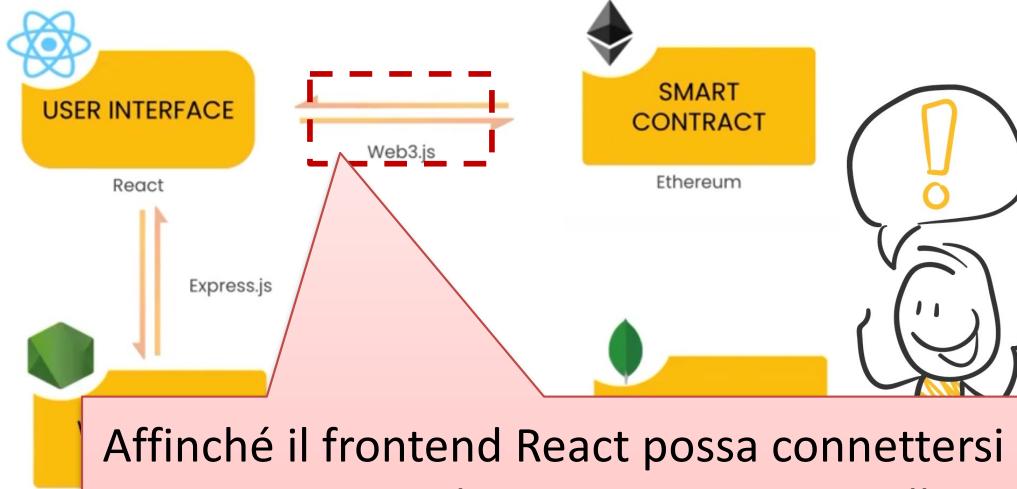
È possibile impiegare lo stack MERN con Ethereum.

MERN è uno stack Javascript per lo sviluppo di applicazioni web:

- MongoDB è un database distribuito NoSQL;
- Express è un framework Node.js che aiuta lo sviluppo di applicazioni web e APIs;
- React è una libreria Javascript per interfacce utente;
- Node.js è un ambiente per eseguire Javascript su un server come back-end.

::: Evoluzione Web (7/8)

Esiste una implementazione di questa architettura?



Affinché il frontend React possa connettersi e comunicare con il nostro contratto intelligente, ha bisogno dell'ABI e dell'indirizzo del contratto.

```
var Contract = require('web3-eth-contract');

// set provider for all later instances to use
Contract.setProvider('ws://localhost:8546');

var contract = new Contract(jsonInterface, address);

contract.methods.someFunc().send({from: ....})
.on('receipt', function(){
  ...
});
```

È possibile impiegare lo stack MERN con Ethereum.

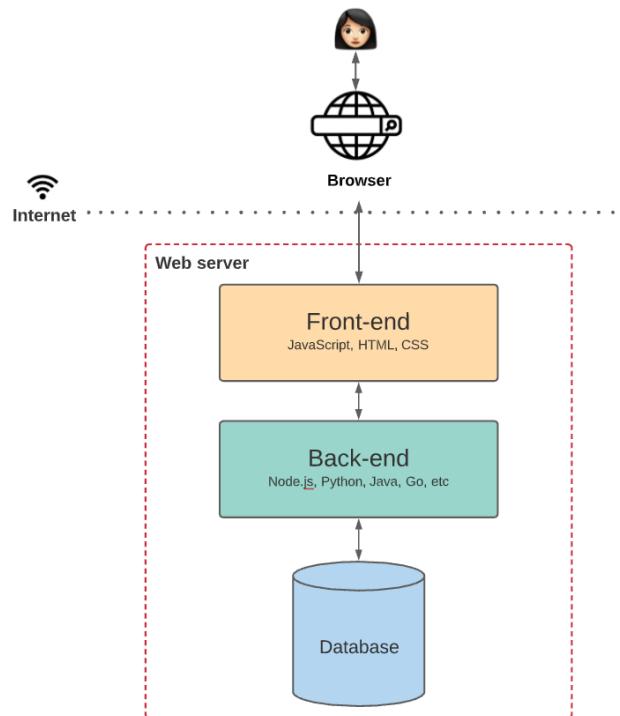


o di applicazioni web:
MySQL;
che aiuta lo sviluppo di
interfacce utente;

- Node.js è un ambiente per eseguire Javascript su un server come back-end.

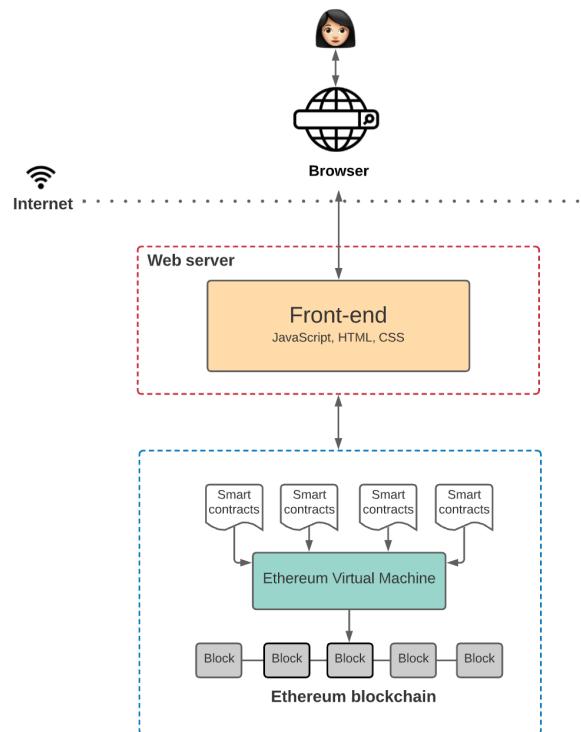
::: Evoluzione Web (8/8)

Esiste anche un'altra possibile evoluzione architetturale. Ripartiamo dallo stack protocolare del Web 2.0, ed introduciamo gli smart contract per la logica di back-end, e la blockchain come database.



::: Evoluzione Web (8/8)

Esiste anche un'altra possibile evoluzione architetturale. Ripartiamo dallo stack protocolare del Web 2.0, ed introduciamo gli smart contract per la logica di back-end, e la blockchain come database.

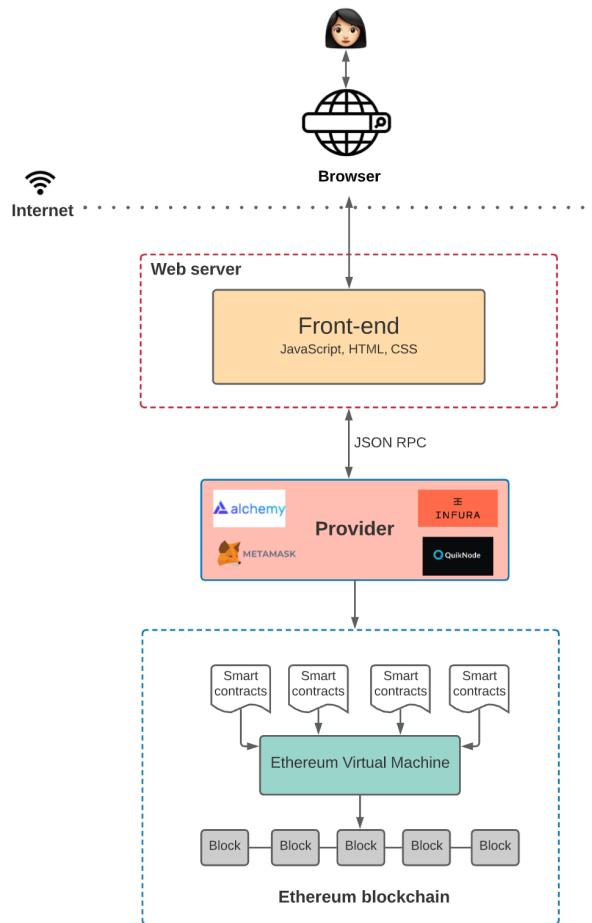


Come comunica il front-end con la blockchain e gli smart contracts? Ovviamente impiegando web3.js, ma dove è in esecuzione la blockchain?

Si utilizza i nodi forniti da servizi di terze parti come Infura, Alchemy e Quicknode, detti provider.

::: Evoluzione Web (8/8)

Esiste anche un'altra possibile evoluzione architetturale. Ripartiamo dallo stack protocollare del Web 2.0, ed introduciamo gli smart contract per la logica di back-end, e la blockchain come database.

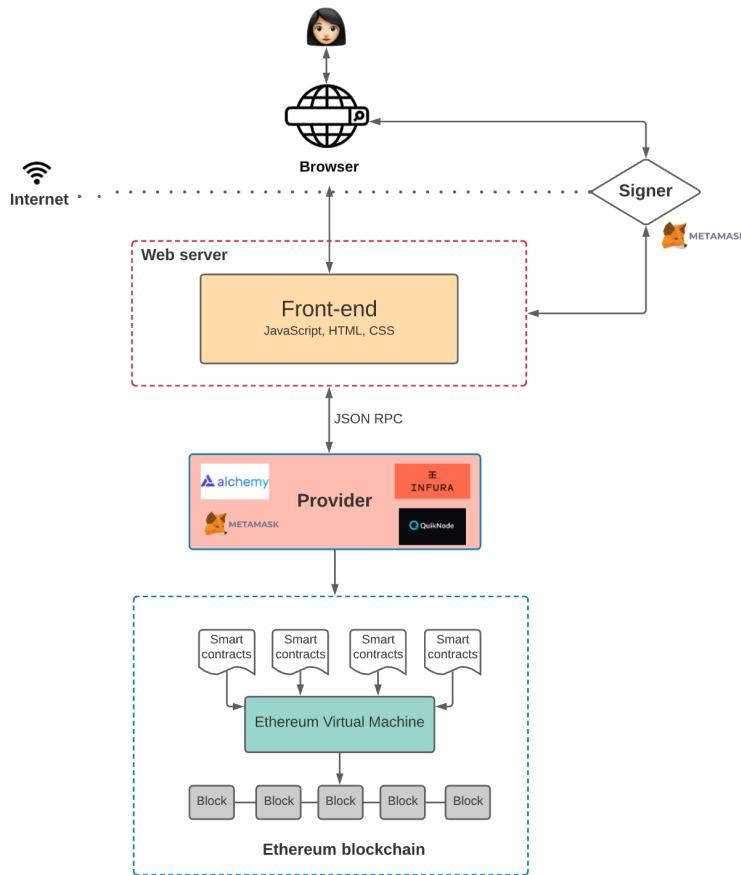


Un provider implementa una specifica JSON-RPC e ciò garantisce che esista un insieme uniforme di metodi quando le applicazioni frontend desiderano interagire con la blockchain.

Tramite un provider è possibile leggere lo stato della blockchain, ma per scrivere bisogna firmare la transazione di richiesta.

::: Evoluzione Web (8/8)

Esiste anche un'altra possibile evoluzione architetturale. Ripartiamo dallo stack protocollare del Web 2.0, ed introduciamo gli smart contract per la logica di back-end, e la blockchain come database.

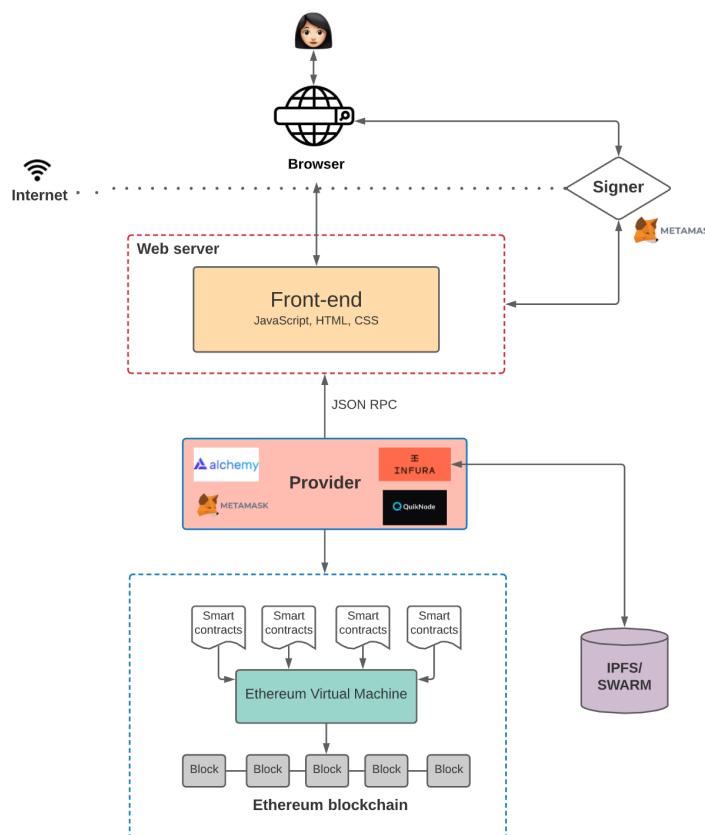


Metamask è uno strumento che consente alle applicazioni di gestire facilmente le chiavi e la firma delle transazioni. Metamask memorizza le chiavi private di un utente nel browser e ogni volta che il frontend ha bisogno che l'utente firmi una transazione, chiama Metamask.

Metamask fornisce anche una connessione alla blockchain (come "provider"). In questo modo, Metamask è sia fornitore che firmatario.

::: Evoluzione Web (8/8)

Esiste anche un'altra possibile evoluzione architetturale. Ripartiamo dallo stack protocollare del Web 2.0, ed introduciamo gli smart contract per la logica di back-end, e la blockchain come database.

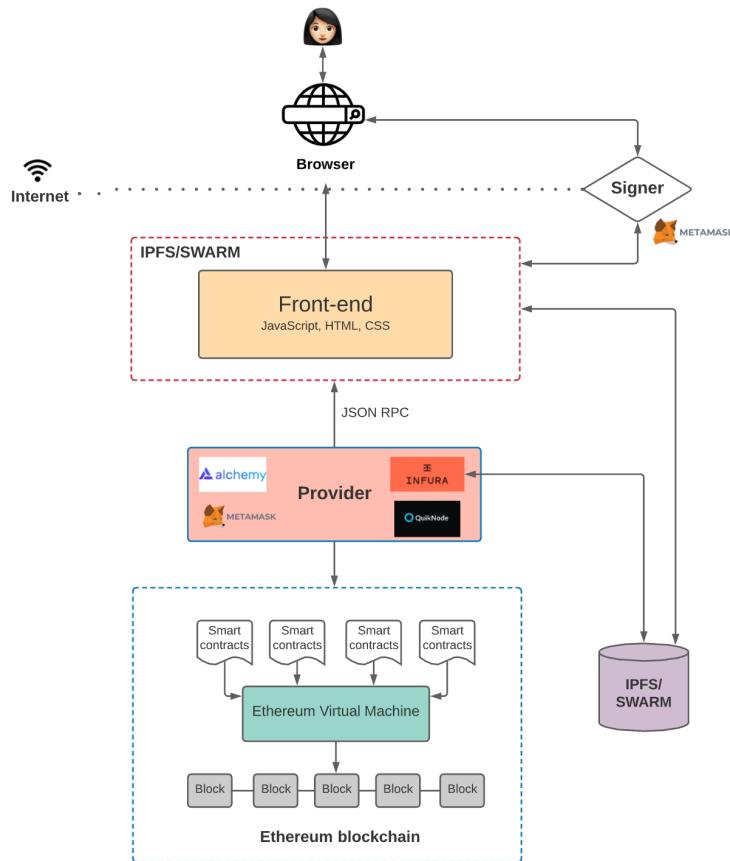


Archiviare tutto sulla blockchain diventa molto costoso, molto lento. Un modo per combattere questo problema è utilizzare una soluzione di archiviazione fuori catena decentralizzata, come IPFS o Swarm.

IPFS è un file system distribuito per l'archiviazione e l'accesso ai dati. Pertanto, anziché archiviare i dati in un database centralizzato, il sistema IPFS distribuisce e archivia i dati in una rete peer-to-peer. Ciò consente di recuperarlo facilmente quando ne hai bisogno.

::: Evoluzione Web (8/8)

Esiste anche un'altra possibile evoluzione architetturale. Ripartiamo dallo stack protocollare del Web 2.0, ed introduciamo gli smart contract per la logica di back-end, e la blockchain come database.

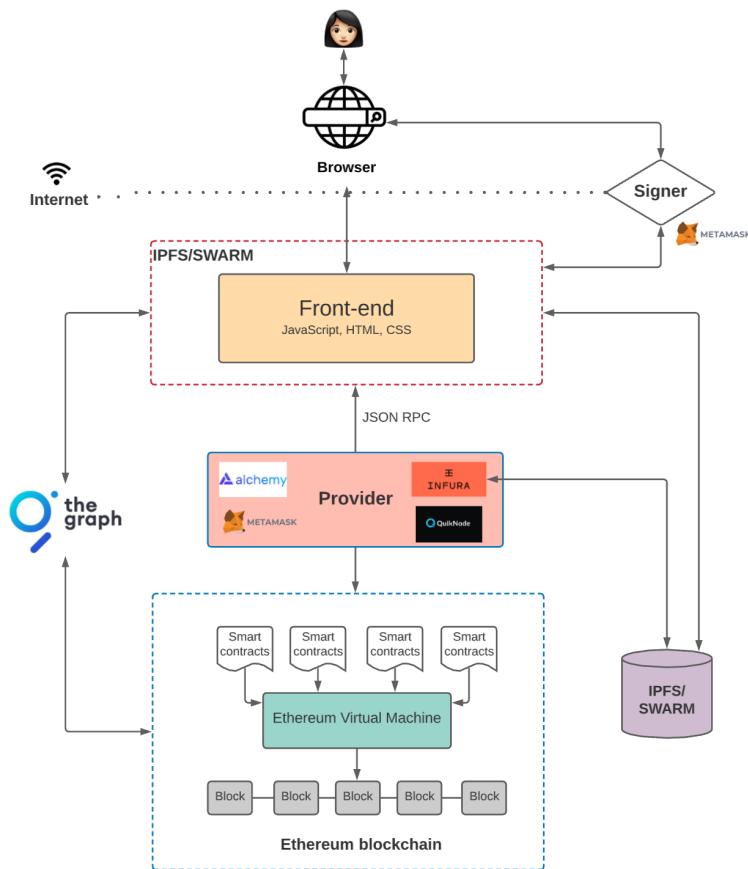


Il codice frontend non è memorizzato sulla blockchain, ma lo si potrebbe ospitare su AWS, come faremmo normalmente nel Web 2.0, ma ciò crea un punto di centralizzazione per la DApp.

Per creare un'app veramente decentralizzata, si può scegliere di ospitare il frontend su una soluzione di archiviazione decentralizzata, come IPFS o Swarm.

::: Evoluzione Web (8/8)

Esiste anche un'altra possibile evoluzione architetturale. Ripartiamo dallo stack protocollare del Web 2.0, ed introduciamo gli smart contract per la logica di back-end, e la blockchain come database.



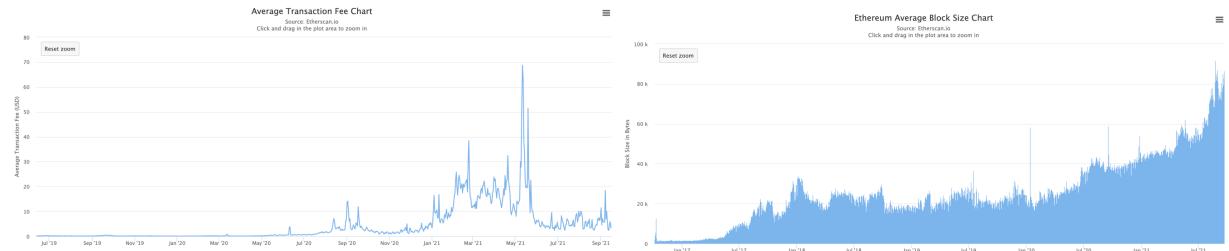
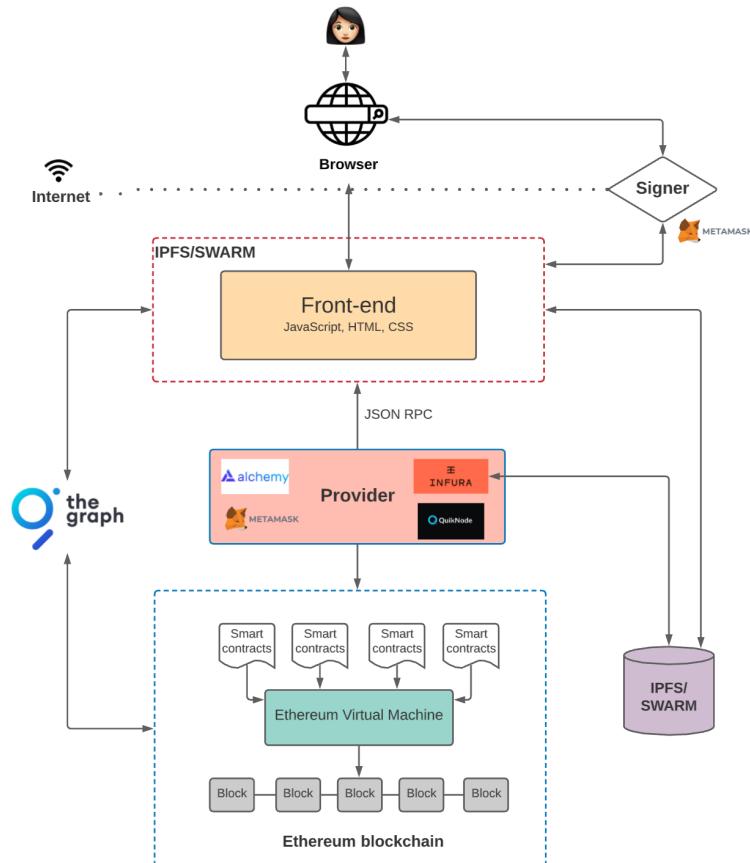
Come leggere dati degli smart contracts sulla blockchain? È possibile utilizzare la libreria Web3.js per eseguire query e ascoltare eventi di smart contract così da eseguire azioni specifiche.

Se un evento non è stato originariamente incluso? L'utilizzo dei callback per gestire le varie logiche dell'interfaccia utente diventa molto complesso.

The Graph è una soluzione di indicizzazione fuori catena che semplifica l'interrogazione dei dati sulla blockchain di Ethereum.

::: Evoluzione Web (8/8)

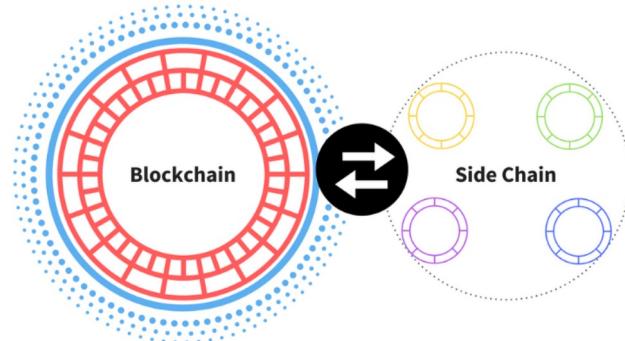
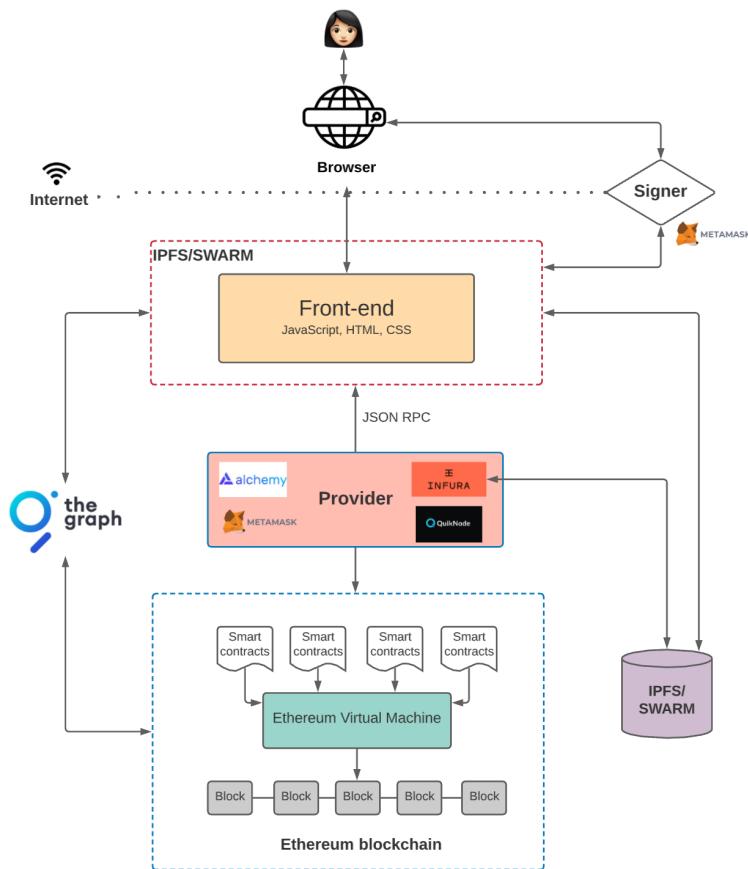
Esiste anche un'altra possibile evoluzione architetturale. Ripartiamo dallo stack protocollare del Web 2.0, ed introduciamo gli smart contract per la logica di back-end, e la blockchain come database.



Costruire una DApp su Ethereum con commissioni elevate sul gas e blocchi interi porta a una pessima user experience.

::: Evoluzione Web (8/8)

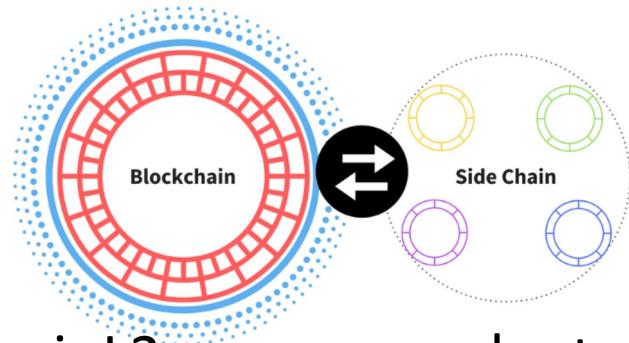
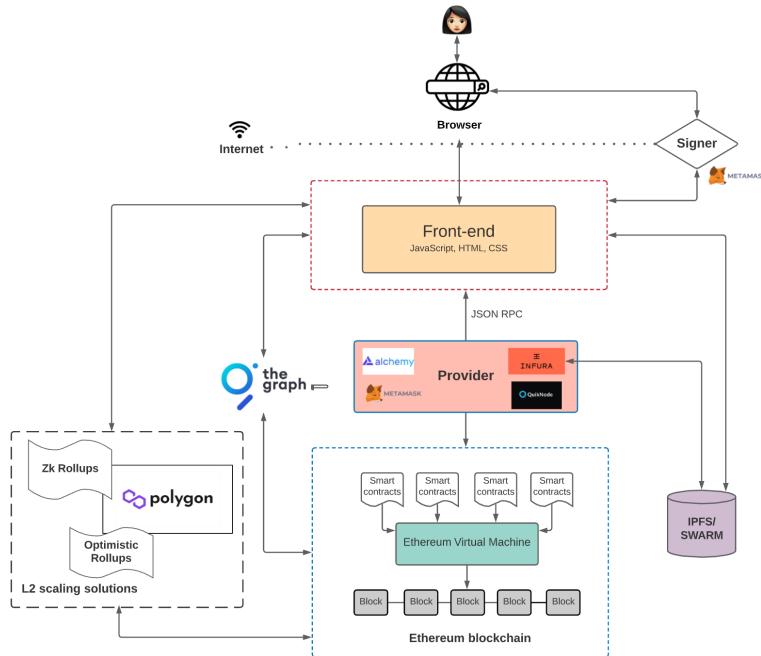
Esiste anche un'altra possibile evoluzione architetturale. Ripartiamo dallo stack protocollare del Web 2.0, ed introduciamo gli smart contract per la logica di back-end, e la blockchain come database.



Una soluzione di ridimensionamento popolare è Polygon. Invece di eseguire transazioni sulla blockchain principale, Polygon dispone di “sidechain” che elaborano ed eseguono transazioni. Di tanto in tanto, la sidechain invia un'aggregazione dei suoi blocchi recenti alla catena primaria.

::: Evoluzione Web (8/8)

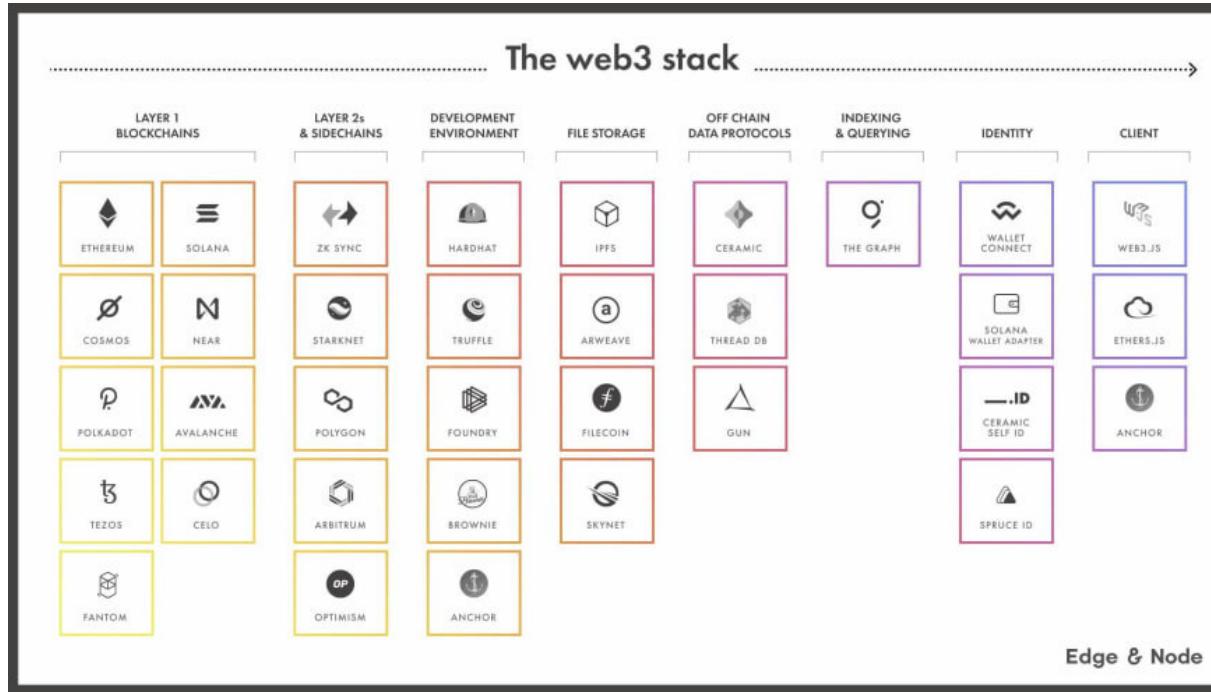
Esiste anche un'altra possibile evoluzione architetturale. Ripartiamo dallo stack protocollare del Web 2.0, ed introduciamo gli smart contract per la logica di back-end, e la blockchain come database.



Le soluzioni L2 eseguono le transazioni (ovvero la parte lenta) fuori catena, con solo i dati delle transazioni archiviati in catena. Questo consente di non dobbiamo eseguire ogni singola transazione sulla catena. Le transazioni diventano più veloci ed economiche, mantenendo la possibilità di comunicare con la blockchain principale se necessario.

::: Evoluzione Web (8/8)

Esiste anche un'altra possibile evoluzione architetturale. Ripartiamo dallo stack protocollare del Web 2.0, ed introduciamo gli smart contract per la logica di back-end, e la blockchain come database.



Lo stack web 3.0 è così composto la veri componenti

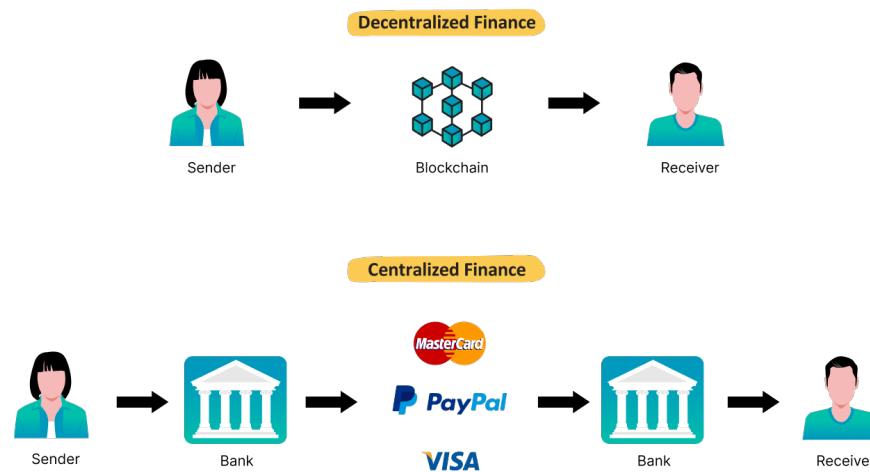


Finanza Dencentralizzata

::: Finanza Decentralizzata (1/4)

Le DApp rappresentano un nuovo paradigma nello sviluppo del software, rivoluzionando le applicazioni tradizionali in molti settori. Sono diventati particolarmente popolari nella finanza, dove hanno dato vita a un nuovo settore chiamato Finanza Decentralizzata (DeFi):

- un'alternativa globale open source a ogni servizio finanziario utilizzabile oggi, come per esempio i conti correnti, prestiti, trading e prodotti assicurativi che sono accessibili da qualunque angolo del mondo, a patto che si abbia uno smartphone e una connessione a Internet.



::: Finanza Decentralizzata (1/4)

Le DApp rappresentano un nuovo paradigma nello sviluppo del software, rivoluzionando le applicazioni tradizionali in molti settori. Sono diventati particolarmente popolari nella finanza, dove hanno dato vita a un nuovo settore chiamato Finanza Decentralizzata (DeFi):

- un'alternativa globale open source a ogni servizio finanziario utilizzabile oggi, come per esempio i conti correnti, prestiti, trading e prodotti assicurativi che sono accessibili da qualunque angolo del mondo, a patto che si abbia uno smartphone e una connessione a Internet.

Alcune delle piattaforme DeFi più conosciute sono gli exchange decentralizzati (DEX), mercati di concessione e contrazione di prestiti e asset fisicamente tokenizzati, come l'oro. Addirittura, le piattaforme DeFi si sono espansse per includere altri servizi finanziari come derivati, network per pagamenti e assicurazioni.

::: Finanza Decentralizzata (2/4)

Exchange decentralizzati (DEX) è simile alle piattaforme di trading online, come eToro, eccetto che funziona tramite smart contract e utilizza la blockchain. Invece degli asset tradizionali, consentono di scambiare criptovalute e altri token. Vantaggi:

- Maggiore sicurezza: il rischio di perdere asset per via di attacchi hacker esterni o frodi interne è nettamente minore.
- Minor spazio all'errore umano: invece di affidarsi esclusivamente agli umani per gestire le operazioni, la DeFi si appoggia su regole scritte e codificate (smart contract) con minime interazioni umane.
- Un elevato livello di fiducia da parte degli utenti, dato che ognuno ha la possibilità di apprendere il codice utilizzato o regole.
- Alta trasparenza e privacy: tutte le attività transazionali sono pubbliche e visibili a chiunque, ma la privacy è protetta, dato che ogni transazione non è connessa direttamente ad alcuna identità reale.

::: Finanza Decentralizzata (3/4)

- Accesso ovunque, in qualunque momento: DeFi è pensata per potervi accedere globalmente, senza alcuna limitazione o confini internazionali e questo significa che è ben più di una semplice banca decentralizzata.
- Zero burocrazia: la DeFi non richiede alcuna autorizzazione, poiché non ci sono intermediari ma è possibile interagire direttamente con gli smart contract dal proprio portafoglio di criptovalute.
- Esperienza utente flessibile: gli smart contract sono una forma di API open source, la cui interfaccia è alterabile in autonomia.
- Modulare: le nuove DApp possono essere liberamente create o personalizzate combinando perfettamente altri prodotti digitali come fossero pezzi di costruzioni Lego.

I potenziali utilizzi della DeFi sono enormi, tuttavia rimane una tecnologia emergente che si colloca ancora ad uno stadio embrionale.

Al di fuori degli exchange decentralizzati, i servizi finanziari principali della DeFi riguardano il prestito di denaro decentralizzato.

::: Finanza Decentralizzata (4/4)

Consideriamo quali sono i svantaggi nell'utilizzo della DeFi:

- Maggior lentezza nelle transazioni: le blockchain decentralizzate sono attualmente più lente rispetto alle controparti centralizzate, e ciò impatta sulle DApp costruite su di esse.
- Niente reti di sicurezza: nonostante alcuni vedano in ciò un punto di forza della DeFi e altri ne vedano il fattore dominante per l'adozione di massa, la finanza decentralizzata trasferisce interamente la responsabilità dagli intermediari agli utenti: se si trasferisce dei fondi ad un indirizzo sbagliato o si dimenticano le chiavi di sicurezza del proprio portafoglio digitale, i fondi non saranno più accessibili. Non c'è nessun servizio clienti bancario da consultare per annullare transazioni o per riottenere l'accesso al profilo.
- Curva di apprendimento scoscesa: la tecnologia blockchain è già di per sé complessa e gli step successivi per utilizzare i servizi DeFi possono essere parecchio scoraggianti per i principianti, i quali potrebbero commettere un errore e perdere tutti i fondi: per sempre.

::: Decentr. Autonomous Orga. (1/5)

Le DApp stanno rivoluzionando anche il modo di fare impresa e hanno dato vista alle Decentralized Autonomous Organization (DAO): un'organizzazione governata da codice e programmi informatici che ha la capacità di funzionare in modo autonomo, senza bisogno di un'autorità centrale.

- Attraverso l'uso di smart contract, una DAO può lavorare con informazioni esterne ed eseguire comandi basati su di esse – tutto ciò senza alcun intervento umano.
- Una DAO viene solitamente operata da una comunità di parti interessate, incentivate attraverso un qualche tipo di meccanismo di token: qualsiasi risorsa trasferibile digitalmente tra due persone la cui rappresentazione digitale è memorizzata su una blockchain.
- Le regole e i registri delle transazioni di una DAO vengono archiviati in modo trasparente sulla blockchain. In una DAO le decisioni vengono prese attraverso proposte e votazioni a maggioranza per l'esecuzione delle proposte.

::: Decentr. Autonomous Orga. (2/5)

Le organizzazioni tradizionali funzionano secondo una struttura gerarchica e diversi livelli di burocrazia, le DAO non hanno alcuna gerarchia, ma utilizzano meccanismi economici per allineare gli interessi dell'organizzazione con gli interessi dei suoi membri, solitamente attraverso l'uso di teoria dei giochi.

- I membri di una DAO non sono vincolati da nessun contratto formale, ma hanno un obiettivo comune e da incentivi del network connessi alle regole di consenso. Queste regole sono completamente trasparenti e scritte nel software open-source che governa l'organizzazione. Dato che le DAO operano senza frontiere, potrebbero essere soggette a diverse giurisdizioni legali.

Una volta che la DAO viene rilasciata, non può essere controllata da un singolo partecipante, ma solo da una comunità di partecipanti. Se le regole di governance definite nel protocollo sono ben progettate, dovrebbero indirizzare le parti interessate verso il risultato più vantaggioso per il network.

::: Decentr. Autonomous Orga. (3/5)

Le DAO affrontano un dilemma in economia chiamato il problema principale-agente. Quando una persona o un'entità (l'“agente”) ha la capacità di prendere decisioni e intraprendere azioni per conto di un'altra persona o entità (il “principale”), si possono verificare delle problematiche.

- Se l'agente è motivato ad agire nel suo proprio interesse, potrebbe ignorare gli interessi del principale. Questa situazione permette all'agente di assumere rischi per conto del principale.
- Potrebbero esserci delle asimmetrie informative tra il principale e l'agente. Il principale potrebbe non scoprire mai che sta venendo sfruttato e non ha modo di assicurarsi che l'agente stia agendo nel suo interesse.

Consentendo un maggiore grado di trasparenza, reso possibile dalle blockchain, i modelli di incentivi ben progettati alla base delle DAO possono eliminare parti di questo problema.

::: Decentr. Autonomous Orga. (4/5)

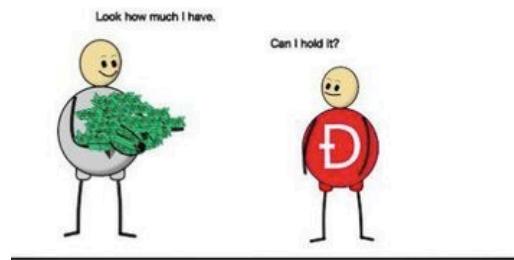
Seppure molto primitiva, il network di Bitcoin potrebbe essere considerato il primo esempio di DAO.

- Opera in modo decentralizzato ed è coordinato da un protocollo di consenso senza gerarchia tra i partecipanti.
- Il protocollo di Bitcoin definisce le regole dell'organizzazione, mentre i bitcoin come moneta forniscono agli utenti un incentivo per proteggere il network.
- L'obiettivo comune nel caso di Bitcoin è conservare e trasferire valore senza un'entità centrale che coordini il sistema.

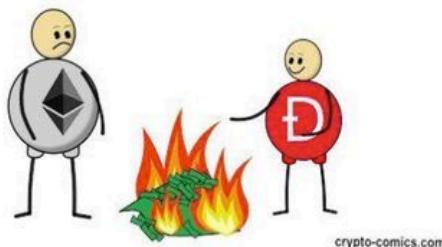
Uno dei primi esempi di DAO «classici» è stato “The DAO”: un insieme di smart contract complessi in esecuzione sulla blockchain di Ethereum che avrebbero dovuto agire come un venture fund autonomo. Gestito dai suoi membri attraverso un sistema di voto basato sulla blockchain, i membri della DAO potevano acquistare token DAO con Ethereum, e poi utilizzarli per partecipare al processo decisionale dell'organizzazione e al finanziamento di progetti.

::: Decentr. Autonomous Orga. (5/5)

I token DAO sono stati venduti in una Initial Coin Offering (ICO). Tuttavia, poco dopo il lancio, circa un terzo dei fondi è stato rubato in uno dei più grandi hack nella storia delle criptovalute. Un bug nei contratti intelligenti del portafoglio di DAO ne consentirebbe il drenaggio. Mentre i programmatore tentavano di correggere il bug, un utente malintenzionato ha sfruttato la vulnerabilità e ha iniziato a sottrarre fondi a The DAO.



Name: **TheDAO Hack**
Type: **Reentrancy**
Year: **2016**
Loss: **\$70M**
Countermeasure: **Ethereum rollback via hard fork**



crypto-comics.com

::: Decentr. Autonomous Orga. (5/5)

I token DAO sono stati venduti in una Initial Coin Offering (ICO). Tuttavia, poco dopo il lancio, circa un terzo dei fondi è stato rubato in uno dei più grandi hack nella storia delle criptovalute. Un bug nei contratti intelligenti del portafoglio di DAO ne consentirebbe il drenaggio. Mentre i programmatore tentavano di correggere il bug, un utente malintenzionato ha sfruttato la vulnerabilità e ha iniziato a sottrarre fondi a The DAO.

Il risultato di questo evento è stata la divisione di Ethereum in due chain in seguito a un hard fork. In una, le transazioni fraudolente sono state di fatto invertite, come se l'hack non fosse mai accaduto. Questa chain è quella che oggi chiamiamo la blockchain di Ethereum. L'altra chain, rispettando il principio “code is law”, ha lasciato le transazioni fraudolente intoccate e ha mantenuto l'immutabilità. Questa blockchain è conosciuta oggi come Ethereum Classic.

::: Problemi DAO (1/3)

- Legali - Il contesto normativo che circonda le DAO è del tutto incerto, e potrebbe essere un ostacolo all'adozione delle DAO.
 - Poiché dall'esecuzione degli smart contract possono scaturire una serie di conseguenze di natura patrimoniale, che di fatto sfuggono al controllo del singolo membro della DAO.
 - Se una DAO rimanesse come una sorta di “aggregato di fatto”, non regolamentato, tra una collettività di soggetti che, in concreto, svolgono una certa attività che comporta il sistematico compimento di atti di disposizione di natura patrimoniale, il rischio è che quest’aggregato assuma le connotazioni di una società di fatto. Ogni membro della DAO rischia di diventare responsabile per gli atti compiuti da altri membri e di rispondere, illimitatamente, con tutto il suo patrimonio.
 - Nessuno dei modelli societari vigenti in Italia sembra adattarsi allo scopo:



::: Problemi DAO (1/3)

- Legali - Il contesto normativo che circonda le DAO è del tutto incerto, e potrebbe essere un ostacolo all'adozione delle DAO.
 - Poiché dall'esecuzione degli smart contract possono scaturire una serie di conseguenze di natura patrimoniale, che di fatto sfuggono al controllo del singolo membro della DAO.
 - Se una DAO rimanesse come una sorta di “aggregato di fatto”, non regolamentato, tra una collettività di soggetti che, in concreto, svolgono una certa attività che comporta il sistematico compimento di atti di disposizione di natura patrimoniale, il rischio è che quest’aggregato assuma le connotazioni di una società di fatto. Ogni membro della DAO rischia di diventare responsabile per gli atti compiuti da altri membri e di rispondere, illimitatamente, con tutto il suo patrimonio.
 - Nessuno dei modelli societari vigenti in Italia sembra adattarsi allo scopo:
 - La società per azioni (S.p.A.) è una società di capitali dotata di personalità giuridica e una autonomia patrimoniale perfetta, nella quale le partecipazioni dei soci sono rappresentate da titoli trasferibili o azioni, e nella quale la gestione è delegata a un organo di gestione come un consiglio di amministrazione (CDA).
 - Una S.p.A. che postula un rigoroso sistema centralizzato di governance, amministrazione e controllo, con relativa vigilanza, è in totale antitesi con la natura decentralizzata di una DAO.

::: Problemi DAO (1/3)

- Legali - Il contesto normativo che circonda le DAO è del tutto incerto, e potrebbe essere un ostacolo all'adozione delle DAO.
 - Poiché dall'esecuzione degli smart contract possono scaturire una serie di conseguenze di natura patrimoniale, che di fatto sfuggono al controllo del singolo membro della DAO.
 - Se una DAO rimanesse come una sorta di “aggregato di fatto”, non regolamentato, tra una collettività di soggetti che, in concreto, svolgono una certa attività che comporta il sistematico compimento di atti di disposizione di natura patrimoniale, il rischio è che quest’aggregato assuma le connotazioni di una società di fatto. Ogni membro della DAO rischia di diventare responsabile per gli atti compiuti da altri membri e di rispondere, illimitatamente, con tutto il suo patrimonio.
 - Nessuno dei modelli societari vigenti in Italia sembra adattarsi allo scopo:
 - La società a responsabilità limitata (S.r.l.), nell'ordinamento italiano, è una società di capitali che, come tale, è dotata di personalità giuridica e risponde delle obbligazioni sociali solamente nei limiti delle quote versate da ciascun socio.
 - Una S.r.l. in certa misura non pone il problema del ruolo di amministrazione attiva dei singoli soci, ma la richiesta delle modalità (forma scritta, atto pubblico notarile) di circolazione delle quote è inconciliabili con la fluidità e l'informalità dei passaggi di mano dei token.

::: Problemi DAO (2/3)

- Il Regolamento europeo sui c.d. MiCA – Markets in Crypto Asset è stata approvata con modifiche dal Parlamento europeo il 20 Aprile 2023, ed è attualmente in una fase di "trialogo" tra Commissione europea, Parlamento europeo e Consiglio. "a rule-based organisational system that is not controlled by any central authority and whose rules are entirely routed in its algorithm (...)" . Questa nuova forma identifica i governance token, per attribuire ai membri diritti amministrativi (come il diritto di voto) e, solitamente, patrimoniali (come il diritto alla partecipazione ai frutti economici dei digital asset gestiti).
- Come realizzare una DAO legalmente riconosciuta?

::: Problemi DAO (2/3)

- Il Regolamento europeo sui c.d. MiCA – Markets in Crypto Asset è stata approvata con modifiche dal Parlamento europeo il 20 Aprile 2023, ed è attualmente in una fase di "trialogo" tra Commissione europea, Parlamento europeo e Consiglio. "a rule-based organisational system that is not controlled by any central authority and whose rules are entirely routed in its algorithm (...)" . Questa nuova forma identifica i governance token, per attribuire ai membri diritti amministrativi (come il diritto di voto) e, solitamente, patrimoniali (come il diritto alla partecipazione ai frutti economici dei digital asset gestiti).
- Come realizzare una DAO legalmente riconosciuta? Si adotta un processo a tre step:
 1. Lo scorso luglio, lo Stato del Wyoming ha approvato definitivamente il disegno di legge (denominato Wyoming Decentralized Autonomous Organization Supplement), che introduce formalmente - per la prima volta - la possibilità di costituire una DAO nella forma di una limited liability company, riconoscendo in tal modo ai nuovi sistemi di governance decentralizzati una propria personalità giuridica.



::: Problemi DAO (2/3)

- Il Regolamento europeo sui c.d. MiCA – Markets in Crypto Asset è stata approvata con modifiche dal Parlamento europeo il 20 Aprile 2023, ed è attualmente in una fase di "trialogo" tra Commissione europea, Parlamento europeo e Consiglio. "a rule-based organisational system that is not controlled by any central authority and whose rules are entirely routed in its algorithm (...)" . Questa nuova forma identifica i governance token, per attribuire ai membri diritti amministrativi (come il diritto di voto) e, solitamente, patrimoniali (come il diritto alla partecipazione ai frutti economici dei digital asset gestiti).
- Come realizzare una DAO legalmente riconosciuta? Si adotta un processo a tre step:
 2. Il Trattato di Alleanza, Commercio e Navigazione stipulato tra la Repubblica Federale di Germania e gli Stati Uniti d'America, il 29 ottobre 1954, attualmente vigente, - tra l'altro - consente il pieno riconoscimento in Germania di LLC di diritto statunitense.



::: Problemi DAO (2/3)

- Il Regolamento europeo sui c.d. MiCA – Markets in Crypto Asset è stata approvata con modifiche dal Parlamento europeo il 20 Aprile 2023, ed è attualmente in una fase di "trialogo" tra Commissione europea, Parlamento europeo e Consiglio. "a rule-based organisational system that is not controlled by any central authority and whose rules are entirely routed in its algorithm (...)" . Questa nuova forma identifica i governance token, per attribuire ai membri diritti amministrativi (come il diritto di voto) e, solitamente, patrimoniali (come il diritto alla partecipazione ai frutti economici dei digital asset gestiti).
- Come realizzare una DAO legalmente riconosciuta? Si adotta un processo a tre step:
 3. Trovato il proprio riconoscimento giuridico, un'entità costituita sottoforma di DAO LLC. potrà, in forza del principio della libertà di stabilimento, essere riconosciuta in tutti gli Stati membri dell'Unione Europea. Gli artt. 49-54 del Trattato sul Funzionamento dell'Unione Europea (TFUE), dettano precise disposizioni in materia di libera circolazione delle persone, dei servizi e dei capitali, volte all'attuazione del del c.d. mercato unico dell'Unione.



::: Problemi DAO (2/3)

- Il Regolamento europeo sui c.d. MiCA – Markets in Crypto Asset è stata approvata con modifiche dal Parlamento europeo il 20 Aprile 2023, ed è attualmente in una fase di "trialogo" tra Commissione europea, Parlamento europeo e Consiglio. "a rule-based organisational system that is not controlled by any central authority and whose rules are entirely routed in its algorithm (...)" . Questa nuova forma identifica i governance token, per attribuire ai membri diritti amministrativi (come il diritto di voto) e, solitamente, patrimoniali (come il diritto alla partecipazione ai frutti economici dei digital asset gestiti).
- Come realizzare una DAO legalmente riconosciuta? Le DAO potrebbero trovare riconoscimento legale sottoforma di fondazioni o, semplicemente, conservare una struttura totalmente decentralizzata, configurandosi giuridicamente come delle general partnership.

Ad ogni modo, attribuire alla DAO una veste giuridica potrebbe rappresentare un grande incentivo per l'espansione del modello, ad esempio offrendo maggiori garanzie sulla responsabilità patrimoniale dei membri dell'organizzazione o imponendo presidi anti-ricilaggio alla comunità decentralizzata degli utenti.

::: Problemi DAO (3/3)

- Attacchi coordinati – La decentralizzazione, immutabilità, trustlessness nelle DAO portano con sé degli svantaggi intrinseci in termini di prestazioni e sicurezza.
- Punti di centralizzazione - In alcuni casi, l'autonomia completa o la decentralizzazione potrebbero non essere né possibili né ragionevoli.

::: Regolamento MiCA

Al centro del regolamento MiCA c'è l'introduzione di requisiti chiari per le cripto-attività che non rientrano nella legislazione esistente sui servizi finanziari, con l'obiettivo di mitigare i rischi per gli investitori, prevenire l'abuso di mercato e combattere la criminalità finanziaria (secondo le definizioni all'art. 3):

- i token collegati ad attività (cd: ART)
- i token di moneta elettronica (cd: EMT), e
- altre forme di cripto-attività come utility token.

Si esclude esplicitamente dal suo ambito di applicazione le cripto-attività uniche e non fungibili (NFT).

Oltre a regolare l'emissione e l'offerta di tali cripto-attività, il regolamento stabilisce anche obblighi precisi per i fornitori di servizi ad esse correlati, inclusi, ma non limitati a, l'operatività di piattaforme di scambio, la custodia sicura di cripto-attività per conto terzi, e la consulenza in materia, enfatizzando ulteriormente la sicurezza e la trasparenza nel crescente ecosistema delle cripto-attività.

::: Regolamento MICA

MICA disciplina l'offerta e la commercializzazione al pubblico di cripto-attività, in relazione ai quali gli emittenti sono tenuti a redigere, con obblighi di informativa, uno specifico documento (c.d. White Paper), a notificarlo alle autorità competenti (specificando la lista di Stati membri verso i quali si prevede di rivolgere l'offerta) e a pubblicarlo nel proprio sito internet, dove dovrà rimanere liberamente accessibile e disponibile per tutta la durata dell'offerta.

Sono previsti requisiti per la prestazione di servizi specifici, quali la custodia e amministrazione di cripto-attività per conto di terzi (i c.d. wallet che, in particolare, devono adottare una politica per garantire la custodia o il controllo di tali cripto-attività o dei mezzi di accesso alle cripto-attività, quali le chiavi crittografiche), le piattaforme di negoziazione delle cripto-attività, lo scambio di cripto-attività con una moneta fiduciaria o altre cripto-attività, l'esecuzione di ordini, il collocamento, la ricezione e trasmissione di ordini per conto di terzi e la consulenza sulle criptoattività.

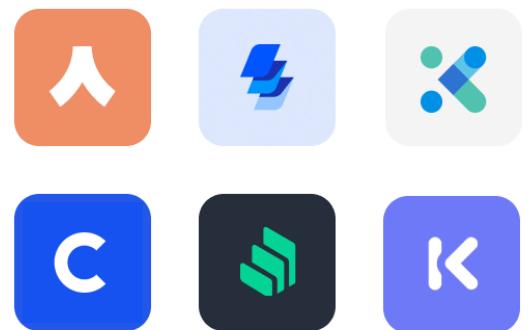
::: Esempio DAO in DeFi

MakerDAO (<https://makerdao.com/it/>) è una DAO con una DApp, che favorisce le funzioni di prestito.



A growing ecosystem

Over 400 apps and services have integrated Dai, including wallets, DeFi platforms, games and more.



::: Esempio DAO in DeFi

MakerDAO (<https://makerdao.com/it/>) è una DAO con una DApp, che favorisce le funzioni di prestito.

- Funziona attraverso l'apertura di un'obbligazione di debito con garanzia collaterale (“Collateralized debt obligation” - CDO), in cui l'utente vincola criptovaluta per agire poi in qualità di controparte collaterale così che si possa prendere a prestito fondi sotto forma del token chiamato DAI, che è uno stable coin con valore uguale al dollaro statunitense ad un tasso di cambio di 1:1 .
- Il prestito viene ripagato con un tasso di interesse annuo, chiamato “commissione di stabilità”, che agisce come protezione nei confronti di un'eccessiva inflazione data dalla quantità totale di DAI disponibile.

MakerDAO emette DAI e ne garantisce la stabilità del valore.

::: Esempio DAO in DeFi

MakerDAO (<https://makerdao.com/it/>) è una DAO con una DApp, che favorisce le funzioni di prestito.

MakerDAO emette DAI e ne garantisce la stabilità del valore.

- Per esempio, se la fornitura di DAI aumenta troppo e causa la diminuzione del valore al di sotto di 1 dollaro, MakerDAO aumenterà la commissione di stabilità (tasso di interesse) per incoraggiare coloro che hanno debiti contratti a ripagarli.
- Al tempo stesso, il token DAI agisce come una versione digitalizzata del dollaro statunitense. Mentre le criptovalute usate come controparte collaterale sono ancora soggetti alla volatilità insita nel mercato delle criptovalute, DAI rimane ad un tasso di cambio di 1:1 rispetto al dollaro statunitense.
- Ciò genera una naturale domanda di DAI, perché è sia digitale sia stabile, il che lo rende una moneta digitale ideale per pagamenti online, transazioni in-game, prestiti e molto altro ancora.