Discover Avalanche

Workshop

Novembre 2025

Glossario

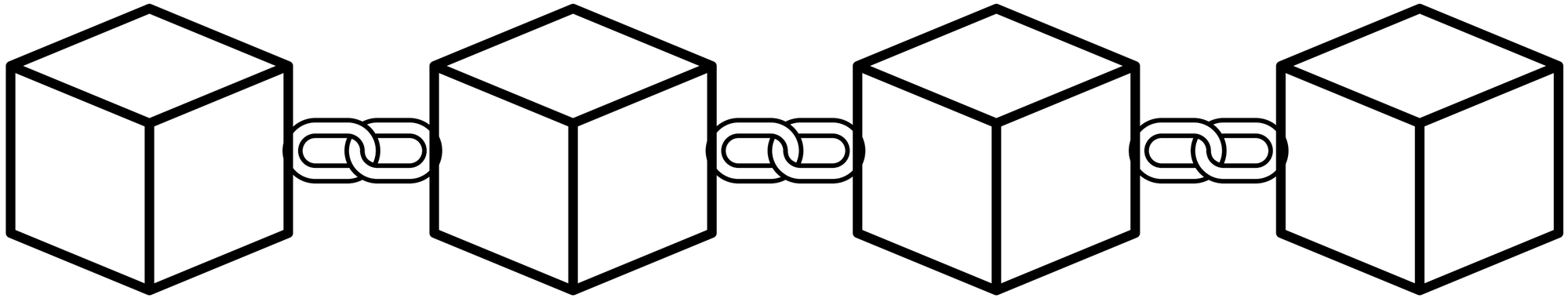
- Blockchain
- come funziona alla base
- Funzioni hash
- Firme digitali
- Bitcoin
- Avalanche

Introduzione

Cosa sono le blockchain

Una **blockchain è un registro condiviso, immutabile e decentralizzato**

- Non è memorizzato in un singolo posto ma in più server chiamati nodi che ne possiedono una copia identica
- Immutabile → i dati non si possono modificare senza che l'intera rete se ne accorga
- Decentralizzato → senza autorità centrale



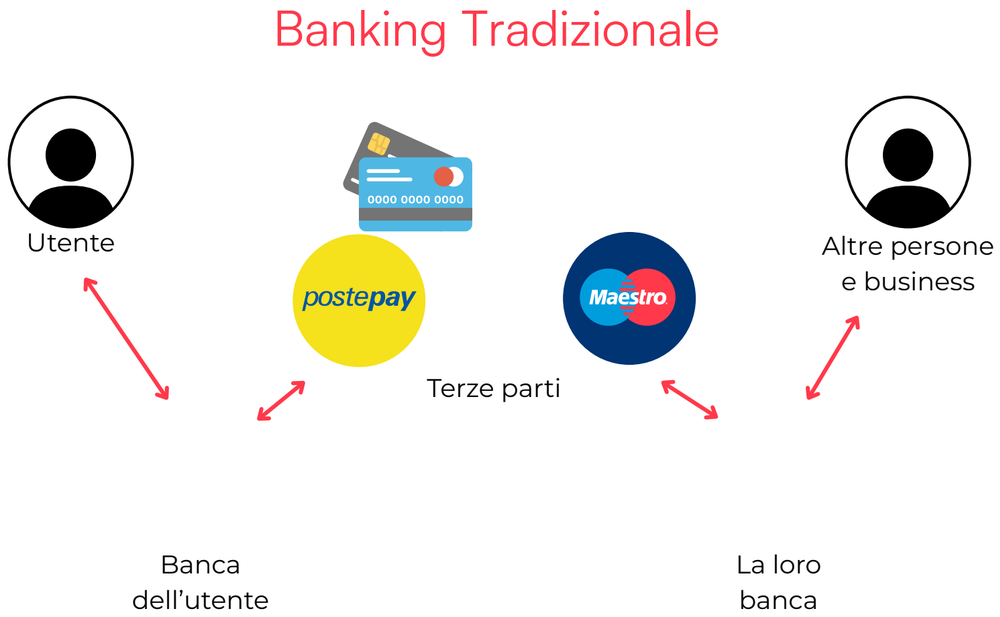
Perchè nasce la blockchain?
Problema di fiducia

Introduzione

Banking Tradizionale VS Blockchain

- Sistema Bancario → Centralizzato, basato sulla fiducia nelle istituzioni
- Blockchain → Decentralizzato, basato sulla verifica crittografica

La validazione non la fa una banca, la fanno migliaia di nodi che eseguono un protocollo crittografico condiviso

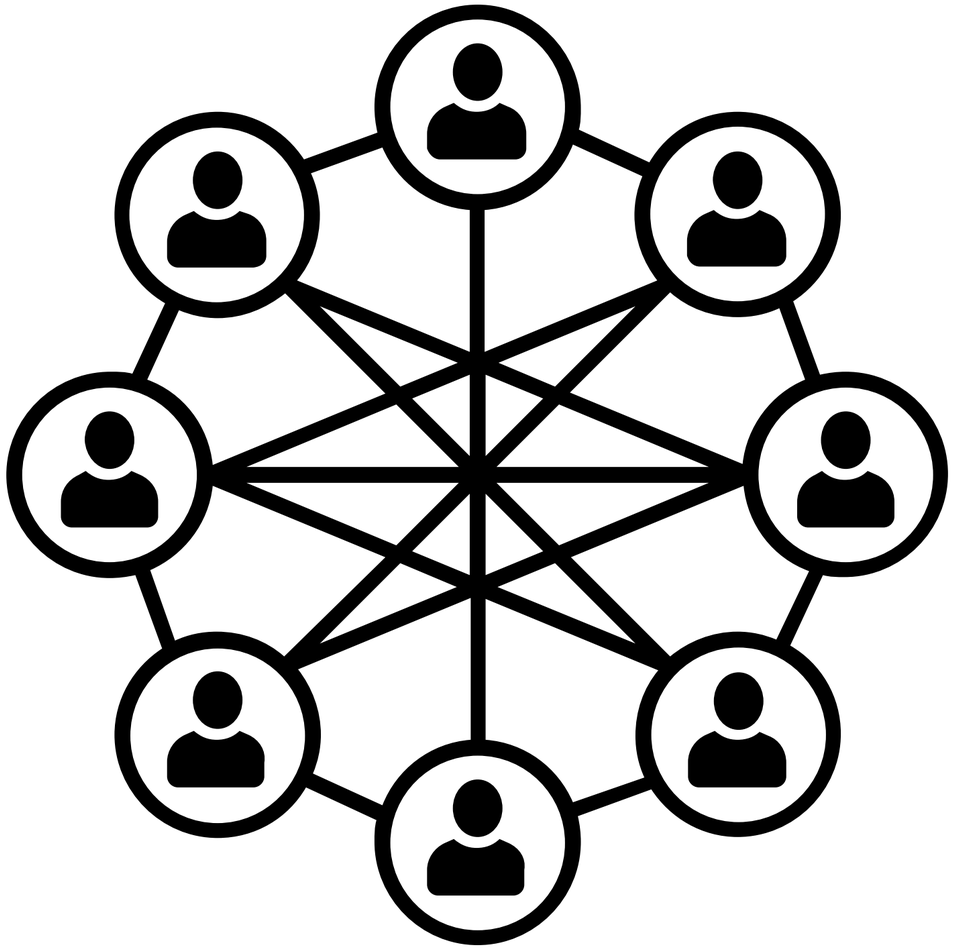


Introduzione

Perché sono nate le blockchain

Nella blockchain l’intermediario non serve più, perché la fiducia non è riposta in un ente, ma nella crittografia, nelle regole condivise del protocollo e nel consenso distribuito della rete

Non dobbiamo più fidarci di chi gestisce il sistema, dobbiamo fidarci di come il sistema è costruito



Ma come è possibile?
E cosa lo rende possibile?

Funzionamento della blockchain

Funzioni Hash

Blockchain → catena di blocchi collegati crittograficamente (?)

Funzione hash: prende in input qualcosa e produce una stringa di lunghezza fissa che sembra casuale

Esempio:

- Ciao → 8f434...9a3b
- ciao → hash completamente diverso

Effetto valanga → minuscolo cambio dell'input comporta un cambaimento totale dell'hash

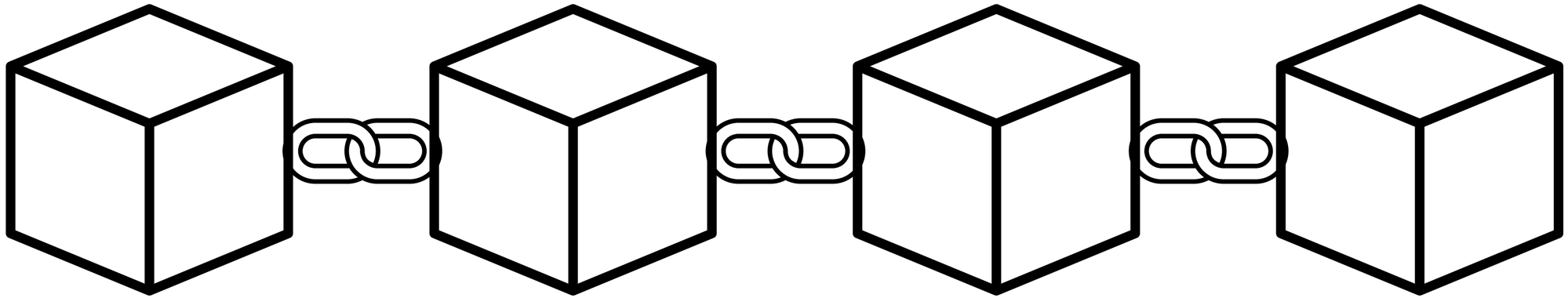
Introduzione

Di cosa è fatto il registro? Di blocchi

Blockchain → catena di blocchi

Un blocco è contenitore digitale di informazioni:

- non è un concetto astratto contiene dati reali
- hash: identificatore crittografico
- hash del blocco precedente: crea la catena



Funzionamento della blockchain

Funzioni Hash: proprietà

Funzione $h : \{0,1\}^* \rightarrow \{0,1\}^n$

$\rightarrow$ stringa x di lunghezza $*$ associa un messaggio $h(x)$ di lunghezza finita n

Garantiscono le proprietà:

Premiarge Resistence: dato y non è possibile trovare $x : h(x) = y \rightarrow 2^{256}$ operazioni

Pre-immagine resistente → se io ti do un hash, NON puoi risalire all'input originale

Funzionamento della blockchain

Funzioni Hash: proprietà

Funzione $h : \{0,1\}^* \rightarrow \{0,1\}^n$

$\rightarrow$ stringa x di lunghezza $*$ associa un messaggio $h(x)$ di lunghezza finita n

Garantiscono le proprietà:

Second Premiarge Resistence: dati $x, y : h(x) = y$ non è possibile trovare $x_1 : x_1 \neq x \wedge h(x_1) = y$

Seconda pre-immagine resistente → se hai un messaggio e il suo hash, non puoi trovare un messaggio diverso con lo stesso hash

Funzionamento della blockchain

Funzioni Hash: proprietà

Funzione $h : \{0,1\}^* \rightarrow \{0,1\}^n$

$\rightarrow$ stringa x di lunghezza $*$ associa un messaggio $h(x)$ di lunghezza finita n

Garantiscono le proprietà:

Colluson Resistence: non è possibile trovare x e $z : x \neq z \wedge h(x) = h(z)$

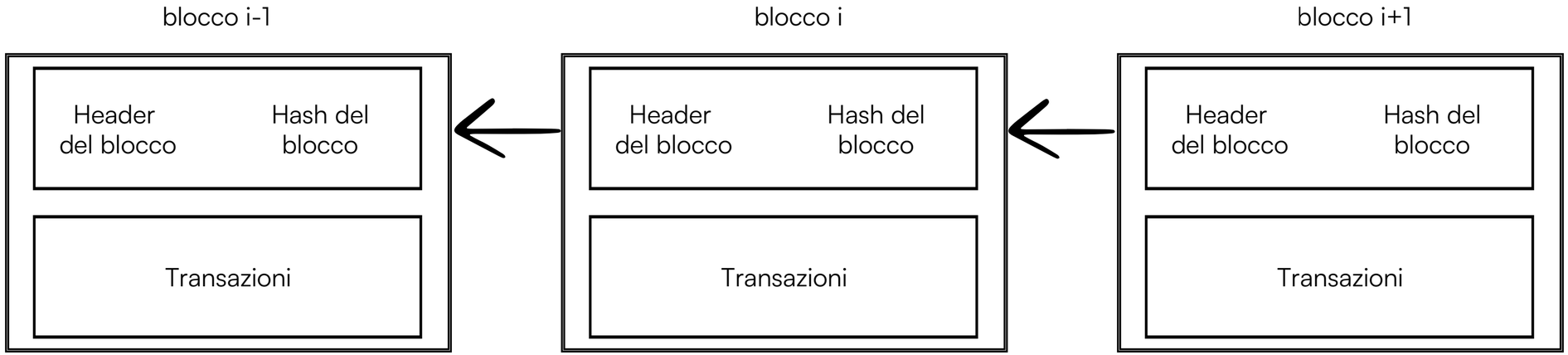
Collision resistant → non si possono trovare due input diversi che generano lo stesso hash

Punto cruciale:se due messaggi potessero avere lo stesso hash... potrei falsificare un documento mantenendo l'hash "uguale"

Introduzione

Come funziona (idea base)

Modifica di un informazione dentro un blocco vecchio → il suo hash cambia → non corrisponde più a quello registrato nel blocco successivo → la catena si spezza e tutta la rete se ne accorge
Implica che blockchain è immutabile: non si può cambiare senza invalidare tutto ciò che viene dopo



Funzionamento di Bitcoin

Bitcoin utilizza SHA-256

- Bitcoin → SHA-256
- Famiglia di funzioni crittografiche sviluppate dalla National Security Agency
- standard ancora oggi considerato sicuro → non è stato ancora violato

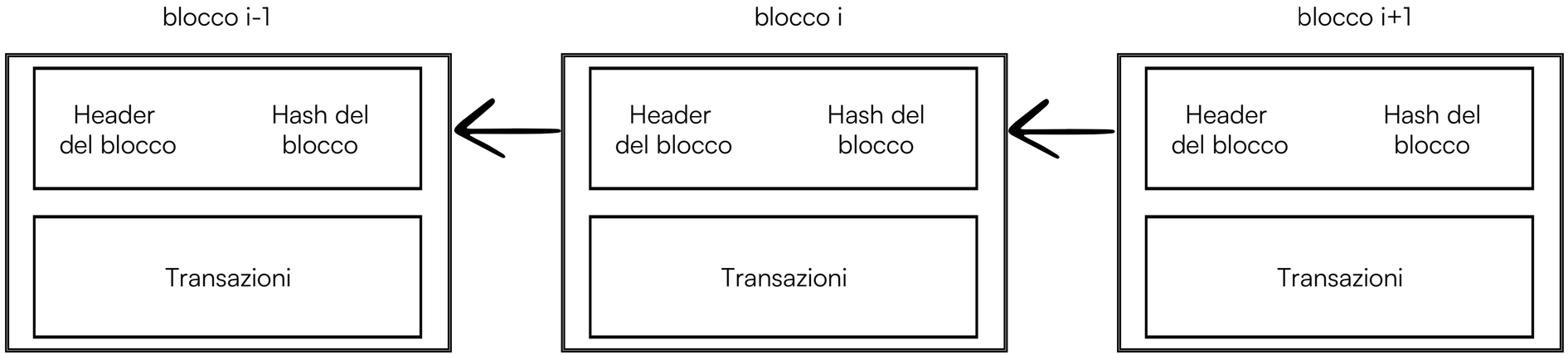
Senza entrare troppo nei dettagli matematici, quello che ci interessa è:

- prende un messaggio di qualsiasi lunghezza
- lo spezza in blocchi
- fa 64 operazioni matematiche per blocco
- produce un hash di 256 bit (64 caratteri esadecimali)

Funzionamento di Bitcoin

Perchè l'hash è così importante?

- Ogni blocco è identificato dal suo hash
- Se cambio una transazione nel blocco, l'hash cambia
- Se l'hash cambia, non corrisponde più a quello registrato nel blocco successivo
- Se non corrisponde, l'intera catena si rompe
- Quindi la blockchain è immutabile → è matematicamente incatenata tramite hash



Funzionamento di Bitcoin

L’hash non basta

Hash garantisce che i dati non siano stati modificati… ma come faccio a sapere CHI li ha generati?

La firma digitale:

- Autenticità → sei davvero tu ad aver firmato
- Integrità → il messaggio non è stato alterato
- Non ripudio → non puoi dire ‘non l’ho firmato io’ dopo averlo firmato

Firma Digitale: Come funziona

Firma digitale:

- È un meccanismo per dimostrare l’autenticità e l’integrità di un messaggio
- Usa una chiave privata per firmare e una chiave pubblica per verificare
- Solo chi possiede la chiave privata può creare una firma valida

Firma Digitale: Algoritmo

Un digital signature scheme è una tripla di algoritmi probabilistici polinomiali (G,S,V) tali che:

- G (Generatore di chiavi): crea la coppia (pk, sk)
- S (Firma): usa sk per firmare un messaggio x → restituisce t
- V (Verifica): controlla con pk se t è una firma valida di x

Firma Digitale: Sicurezza

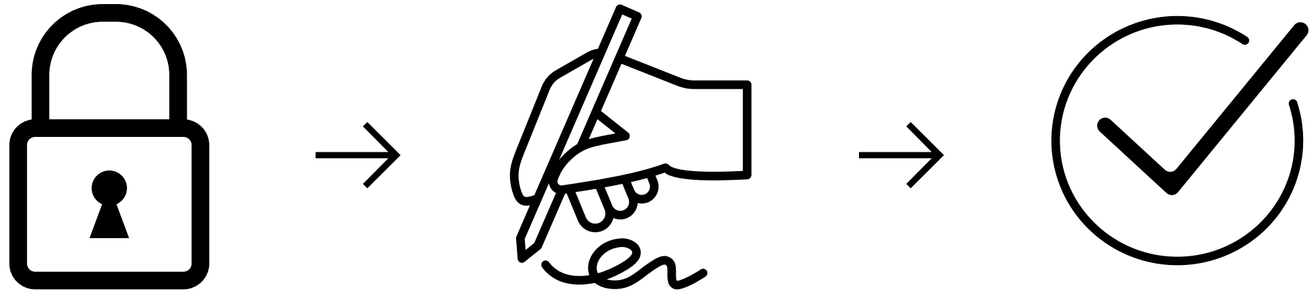
Anche conoscendo la chiave pubblica e potendo chiedere firme su messaggi a scelta, un attaccante non può creare una firma valida su un nuovo messaggio

Nessun algoritmo A efficiente può “indovinare” una nuova firma valida

Firma Digitale

La firma digitale usa crittografia asimmetrica

- Chiave privata → firma
- Chiave pubblica → verifica
- Garantisce autenticità, integrità e non ripudio

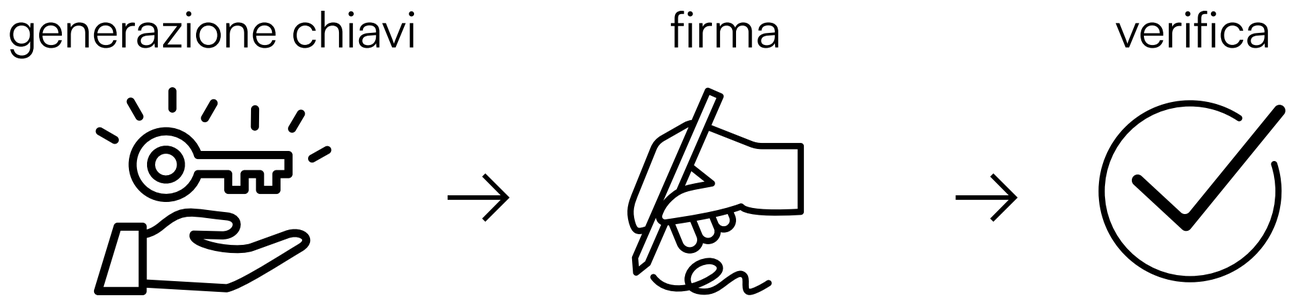


ECDSA

L’algoritmo di firma digitale utilizzato da Bitcoin è ECDSA

Tre fasi principali:

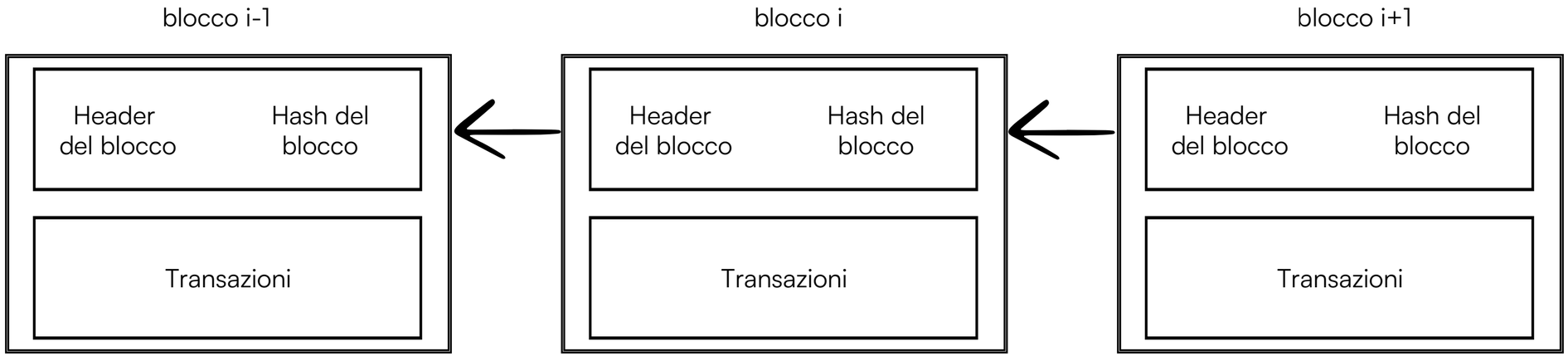
- Generazione delle chiavi
- Firma del messaggio
- Verifica della firma



Collegamento alla Blockchain

Quadro completo:

- Gli hash garantiscono integrità e incatenano i blocchi
- Le firme digitali garantiscono autenticità e non ripudio delle transazioni
- ECDSA è l'algoritmo che Bitcoin usa per farlo in modo sicuro ed efficiente



Bitcoin: moneta digitale decentralizzata

Bitcoin: A Peer-to-Peer Electronic Cash System (2008)

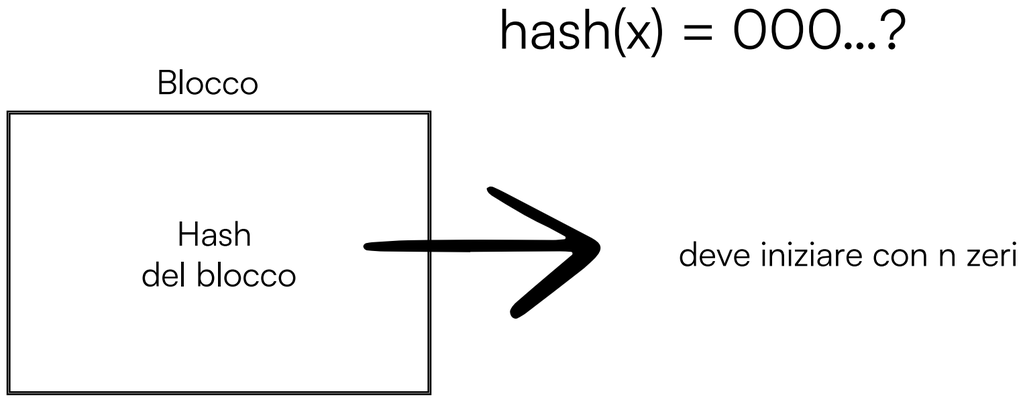
- Satoshi Nakamoto propone una moneta elettronica P2P
- Nessuna banca, nessuna autorità centrale
- Basata su firme digitali e un registro pubblico condiviso

- Ogni passaggio rappresenta una “catena” di firme digitali
- Ogni transazione firma l’hash della precedente e la chiave pubblica del nuovo proprietario
  → creando una catena verificabile da chiunque.

Bitcoin: moneta digitale decentralizzata

Proof-of-Work

- Ogni nodo cerca un nonce che renda l'hash del blocco con n zeri iniziali
- Richiede calcolo intenso → costo computazionale
- Cambiare un blocco → serve rifare tutto il lavoro successivo



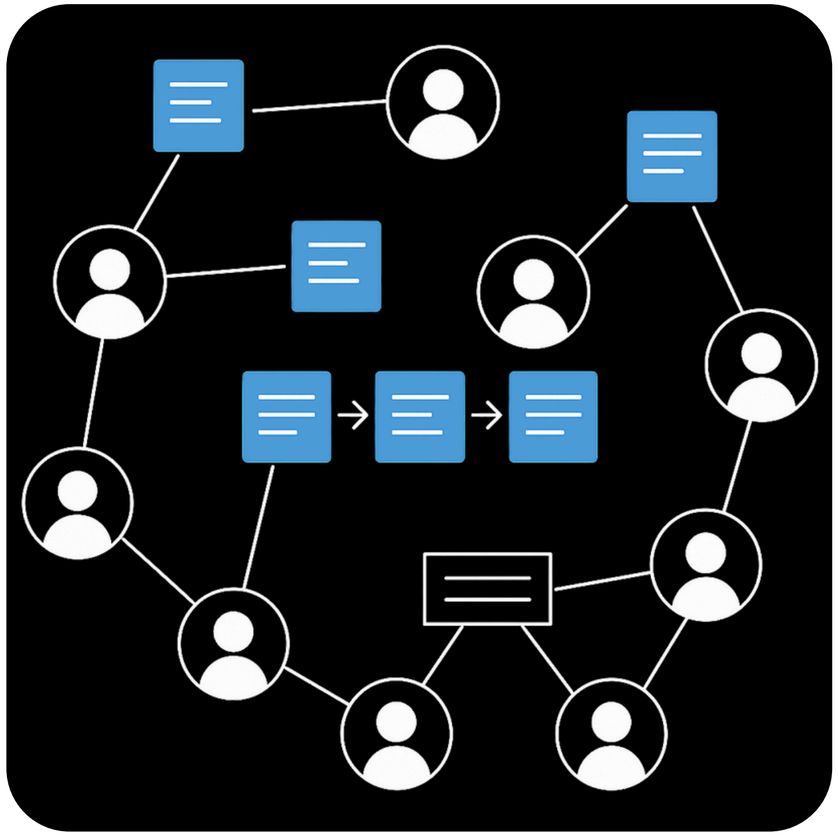
Solo la catena più lunga (più lavoro computazionale) è considerata valida

Bitcoin: moneta digitale decentralizzata

Il consenso

- Le transazioni vengono trasmesse ai nodi
- Ogni nodo crea un blocco con le nuove transazioni
- I nodi competono per trovare la proof-of-work
- Il primo a trovarla diffonde il blocco
- Gli altri lo verificano e lavorano sul prossimo blocco
- Tutti seguono la catena più lunga

Il consenso emerge naturalmente: la maggioranza dei nodi onesti determina la storia ufficiale delle transazioni

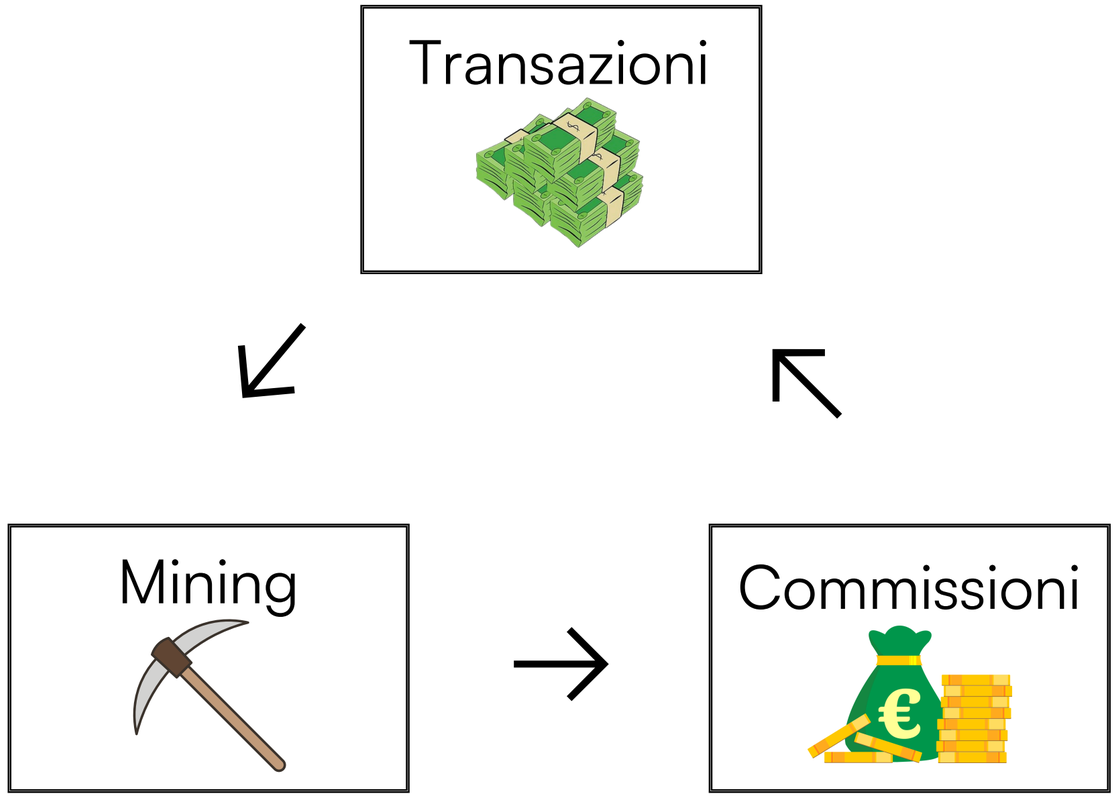


Bitcoin: moneta digitale decentralizzata

Ecosistema Bitcoin

- Transazioni → spostano valore
- Mining → valida e registra le transazioni
- Commissioni → incentivano i miner

È un sistema autosufficiente: gli utenti inviano transazioni, i miner le validano, e vengono ricompensati con nuovi bitcoin e fee



Bitcoin: moneta digitale decentralizzata

Conferma di una transazione

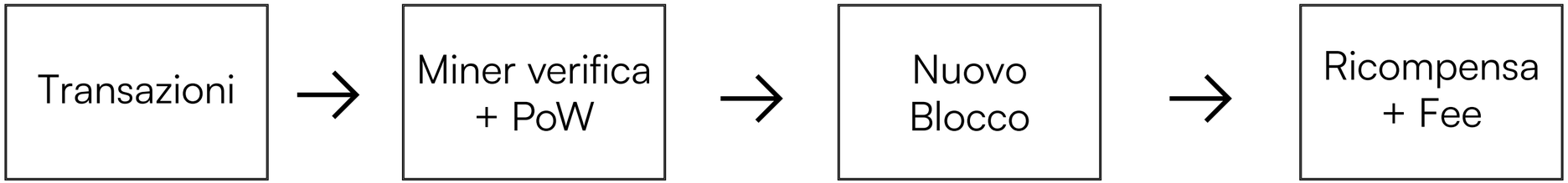
- Una transazione è sicura dopo 6 blocchi (~1 ora)

Ogni blocco successivo rende più difficile modificare le transazioni precedenti → è una sicurezza probabilistica

Bitcoin: moneta digitale decentralizzata

Mining

- I miner raccolgono e verificano le transazioni
- Aggiungono l'hash del blocco precedente
- Cercano un nonce che soddisfi la proof-of-work
- Il vincitore: ottiene la ricompensa in bitcoin (Coinbase) e le commissioni delle transazioni incluse



Il mining è ciò che mantiene il sistema onesto: costa energia modificare la storia

Da Bitcoin ad Avalanche

Cosa sono gli Smart Contract

- Programmi compilati che “hostati” sulla blockchain
- Trasparenti, immutabili, e pubblici
- Gestiscono logica senza intermediari (es. prestiti, scambi, votazioni...)

Esempio

IF [condizione vera]
→ esegui [azione]

```
if (saldo >= prezzo) {
    trasferisci(token);
}
```

Gli smart contract sono la base di tutto il mondo DeFi e molte blockchain moderne

Da Bitcoin ad Avalanche

Dal Proof of Work al Proof of Stake

Proof of Work (Bitcoin):

- i nodi competono risolvendo problemi crittografici → alta sicurezza, ma alto consumo energetico

Proof of Stake:

- i nodi vengono scelti in base a quanti token “bloccano” come garanzia→ più efficiente e sostenibile

	Proof of Work	Proof of Stake
Consenso	Mining (hash)	Staking (validazione)
Energia	Alta	Bassa
Ricompensa	Blocchi minati	Ricompense da validazione
Sicurezza	Potenza computazionale	Valore in stake

Avalanche

Staking mettere in gioco valore per proteggere la rete

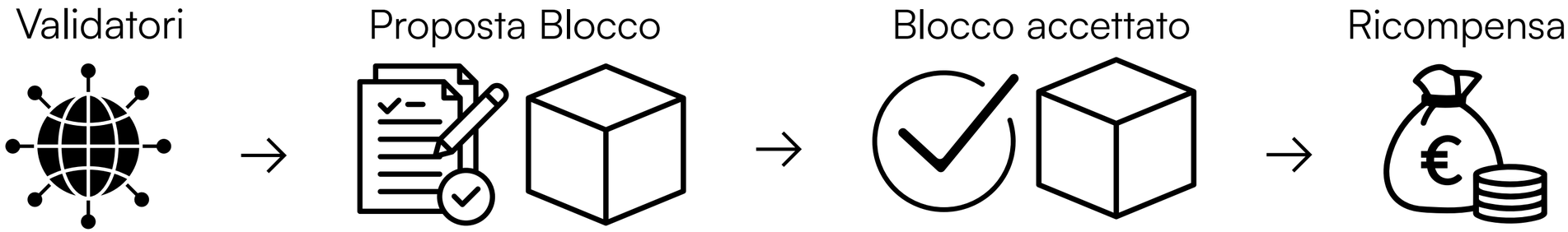
- I nodi bloccano una quantità di token (stake)
- Più stake = maggiore probabilità di essere scelti per validare
- Se si comportano male → perdono i rewards
- In cambio ricevono ricompense (nuovi token)

È come un sistema di “cauzione digitale”: chi partecipa al consenso deve dimostrare di avere qualcosa da perdere

Avalanche

Il consenso PoS in breve

- I validatori mettono in stake i propri token
- Il protocollo seleziona casualmente chi propone il prossimo blocco
- Altri validatori confermano la validità del blocco
- Se tutto è corretto → blocco aggiunto + ricompense distribuite
- Chi tenta di barare → perde parte del rewards



La sicurezza deriva dal valore economico messo in gioco, non dalla potenza di calcolo

Avalanche

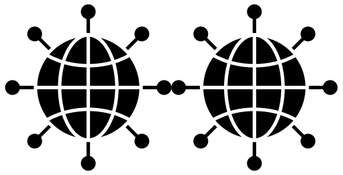
Cos’è Avalanche

- Blockchain di terza generazione
- Basata su Proof-of-Stake e un nuovo meccanismo di consenso “gossip-based”
- Architettura multi-chain: X-Chain, P-Chain, C-Chain

X-Chain

Scambio asset

P-Chain



Gestione variatori
e subnet

C-Chain

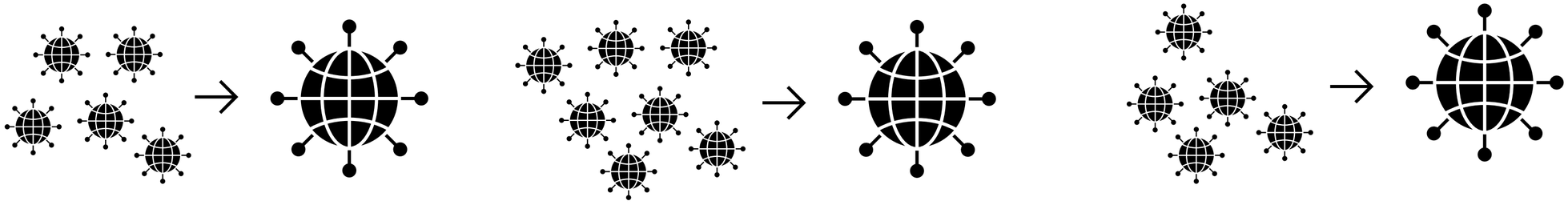
Samrt contract

Avalanche non è una singola blockchain, ma un ecosistema di reti interoperabili

Avalanche

Come funziona il consenso Avalanche

- I nodi non votano tutti: si interrogano casualmente in piccoli gruppi (gossip)
- Se una transazione riceve abbastanza consenso “statistico”, viene accettata
- Velocissimo → finalità in meno di 2 secondi



Invece di fare mining, Avalanche costruisce consenso statisticamente, tramite campionamento casuale tra nodi

Avalanche

Gossip e campionamento casuale in Avalanche

- Il protocollo Avalanche si basa su algoritmi che utilizzano il gossiping (propagazione epidemica) e il campionamento casuale dei nodi
- Ogni nodo, quando riceve una nuovo blocco, non chiede a tutti i nodi della rete, ma solo ad un piccolo sotto-insieme (“sample”) di nodi casuali
- L’uso del gossip rende la propagazione veloce, e il campionamento mantiene bassa la comunicazione per nodo, favorendo scalabilità

Avalanche

Passi del campionamento e decisione

- Un nodo propone o riceve un blocco
- Il nodo sceglie un campione casuale di k nodi e chiede la loro preferenza
- Se almeno α dei k rispondono con la stessa preferenza, il nodo adotta quella preferenza e incrementa un contatore di “successi consecutivi”
- Se lo stesso risultato (quorum ≥ α) si ripete per β volte consecutive, il nodo decide (accetta o rifiuta) la proposta
- Nel frattempo, la proposta viene propagata via gossip a nuovi nodi, aumentando la visibilità e velocizzando la convergenza

Avalanche

Architettura Multi-Chain: Le tre catene di Avalanche

Catena	Nome	Funzione principale
X-Chain	Exchange Chain	Gestisce la creazione e lo scambio di asset digitali
C-Chain	Contract Chain	Esegue smart contract
P-Chain	Platform Chain	Coordina validatori, subnet e staking

Avalanche

Perché questo meccanismo?

- Scalabilità: Ogni nodo invia/riceve un numero costante di messaggi (≈ k), indipendentemente dal numero totale dei nodi
- Velocità: La propagazione via gossip + campionamento rapido consente bassa latenza nella decisione
- Sicurezza probabilistica: Anche se il meccanismo è probabilistico, le scelte dei parametri (α, β) permettono di rendere la probabilità di divergenza arbitrariamente bassa
- Robustezza e decentralizzazione: Non c'è leader; ogni nodo può partecipare. Il gossip assicura che l'informazione raggiunga ampiamente la rete

Avalanche

Proof-of-Stake in Avalanche: Staking e Validatori

- I validatori mettono in stake AVAX per partecipare al consenso
- Maggiore stake → maggiore influenza nel processo di validazione
- Niente mining → consumo energetico minimo

Avalanche mantiene sicurezza e decentralizzazione, ma con consumi energetici molto più bassi del Proof-of-Work

Avalanche

Creare la tua blockchain personalizzata su Avalanche

- È possibile lanciare una blockchain dedicata come parte di una subnet

- Subnet: insieme di validatori che raggiungono consenso su una o più blockchain specifiche

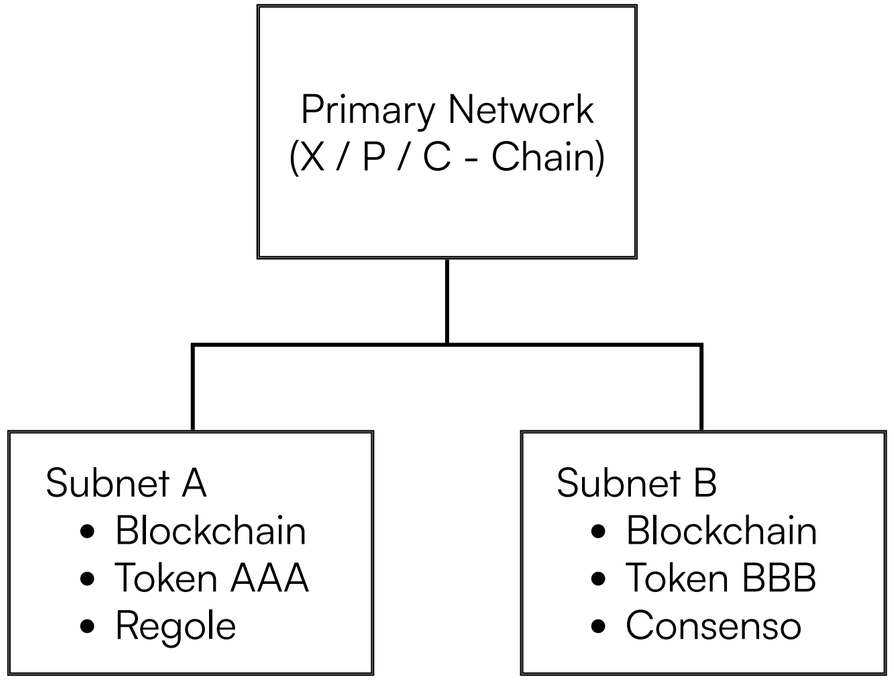
Le blockchain create possono avere:

- Proprie regole di consenso (Avalanche, Snowman, o personalizzate)
- Proprio token nativo (per gas o governance)
- Parametri di rete dedicati (fee, limiti, privacy, compliance)
- Le subnet sono indipendenti, ma possono interagire con la Primary Network per interoperabilità e sicurezza

Avalanche

Creare la tua blockchain personalizzata su Avalanche

- Scalabilità orizzontale: più subnet = più capacità di rete
- Personalizzazione totale: consenso, fee, governance
- Interoperabilità: connessione con la rete principale
- Isolamento: eventuali errori o carichi non impattano altre chain



Avalanche

Perché Avalanche

1. Smart contract sulla C-Chain (EVM compatibile):

   Facilità di sviluppo

   - Possiamo riutilizzare librerie, strumenti e linguaggi (Solidity, Remix, Hardhat)

   - Distribuzione rapida di smart contract su una rete veloce e a basso costo

   - Ecosistema DeFi e NFT già consolidato sulla C-Chain

Da Bitcoin ad Avalanche

Perché Avalanche

2. Backend che interagisce con X-Chain e P-Chain:

Integrazione diretta con le componenti base

- X-Chain: gestione e scambio di asset digitali (transazioni, token)
- P-Chain: gestione di validatori e creazione di subnet
- Permette di costruire un’infrastruttura applicativa completa, dal layer transazionale alla governance

Da Bitcoin ad Avalanche

Perché Avalanche

3. Creare la nostra blockchain personalizzata:

Scalabilità e controllo totale

- Possiamo creare una blockchain dedicata con regole, token e consenso personalizzati
- Ogni subnet ha validatori propri, indipendenti dalla rete principale
- Offre prestazioni dedicate e isolamento da congestione o gas fees esterne

Linguaggio Solidity

Cos’è Solidity

- Linguaggio orientato agli oggetti per smart contract
- Sintassi simile a JavaScript/C++
- Compilato in EVM
- Tipizzato e case-sensitive: ogni variabile deve avere un tipo specifico

Linguaggio Solidity

Dove viene eseguito

- Sulla blockchain → codice immutabile dopo il deploy
- Ogni esecuzione ha un costo in gas
- Dati memorizzati in modo trasparente e persistente

Linguaggio Solidity

Strumenti utilizzati

Framework → Foundry include strumenti integrati:

- Forge → compila, testa e fa deploy degli smart contract
- Cast → interagisce con la blockchain da terminale (leggere/scrivere dati, chiamare funzioni, ecc.)
- Anvil → crea una blockchain locale per test

Testnet → Fuji è la rete di test di Avalanche:

- Simula la rete reale senza usare soldi veri → le transazioni usano AVAX finti
- I contratti deployati qui non hanno valore economico

Wallet: Core Wallet

Linguaggio Solidity

Tipi di dato

Tipo	Esempio
uint / int	uint256 x = 10;
bool	bool attivo = true;
string nome = "Alice";	string nome = "Alice";
address	address wallet;
array	uint[] numeri;
mapping	mapping(address => uint)

Linguaggio Solidity

Modificatori di visibilità

Modificatore	Accesso
public	leggibile da tutti
private	solo nel contratto
internal	contratto + derivati
external	solo chiamate esterne

Linguaggio Solidity

Modificatori di funzione

Modificatore	Accesso
pure	funzioni di utility senza stato
view	non modifica ma legge lo stato
payable	permette di ricevere AVAX

Linguaggio Solidity

Variabili speciali

Variabile	Uso
msg.sender	indirizzo che chiama
msg.value	ETH inviati
block.timestamp	tempo del blocco
address(this)	indirizzo del contratto

Core Wallet

Creare un core wallet

Scaricare l’estensione web di Core Wallet

- https://chromewebstore.google.com/detail/core-wallet-crypto-made-e/agoakfejjabomempkjlepdflaleeobhb

Creare un account con l’opzione *“Manually Create a Wallet”*

Copiare la seed phrase e salvarla in un posto sicuro

Core Wallet

Come richiedere AVAX di test

Creare un account sul builder hub:

https://build.avax.network/console/primary-network/faucet

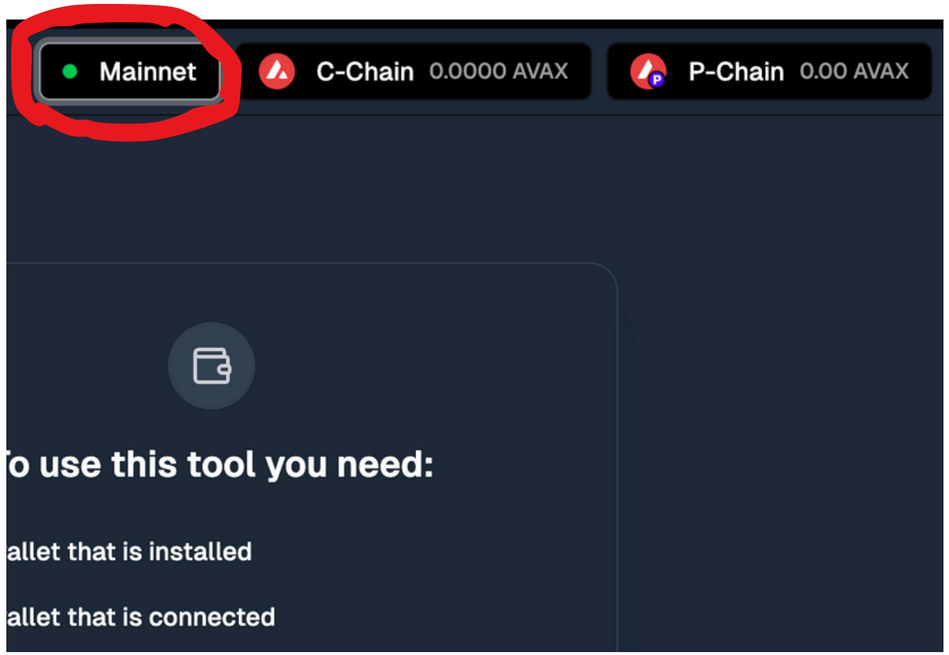
Richiedere fondi per Avalanche C-Chain

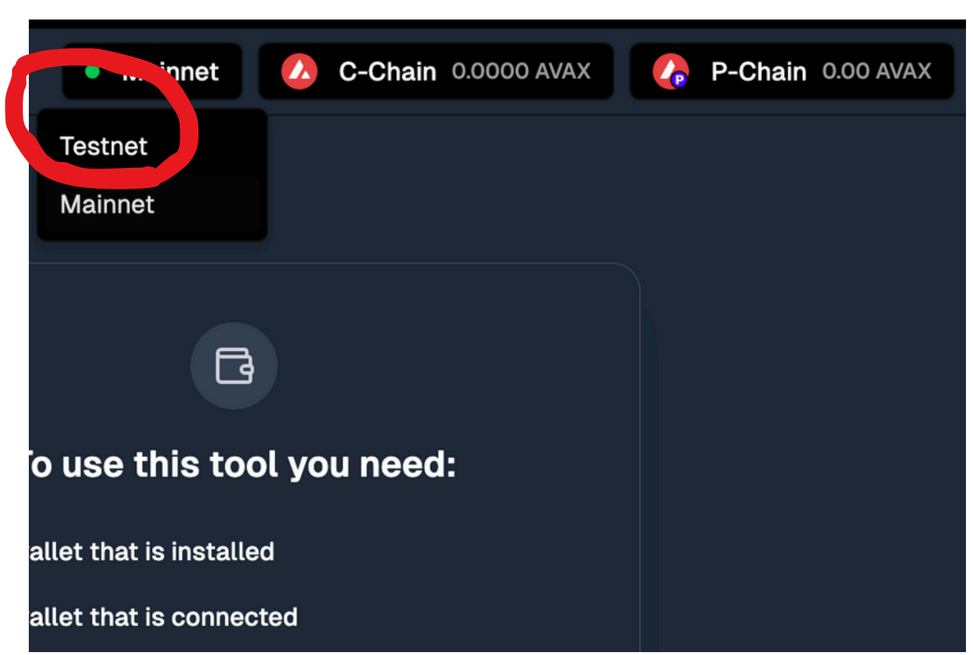
Connettere Core Wallet

Core Wallet

Come richiedere AVAX di test

Creare un account sul builder hub:
https://build.avax.network/console/primary-network/faucet

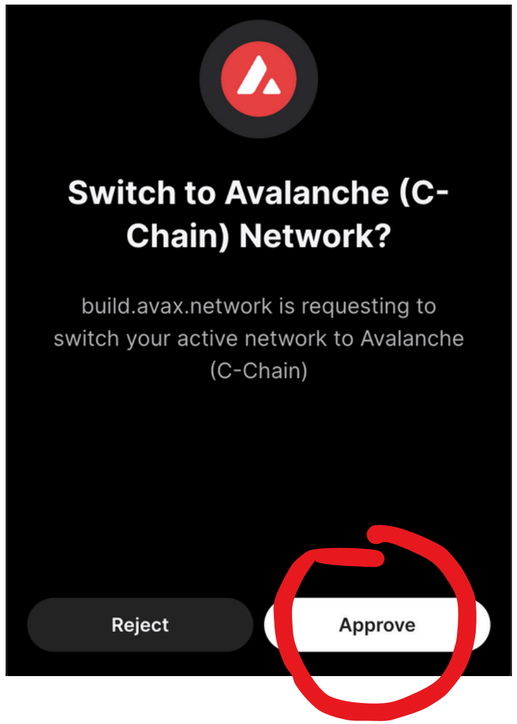


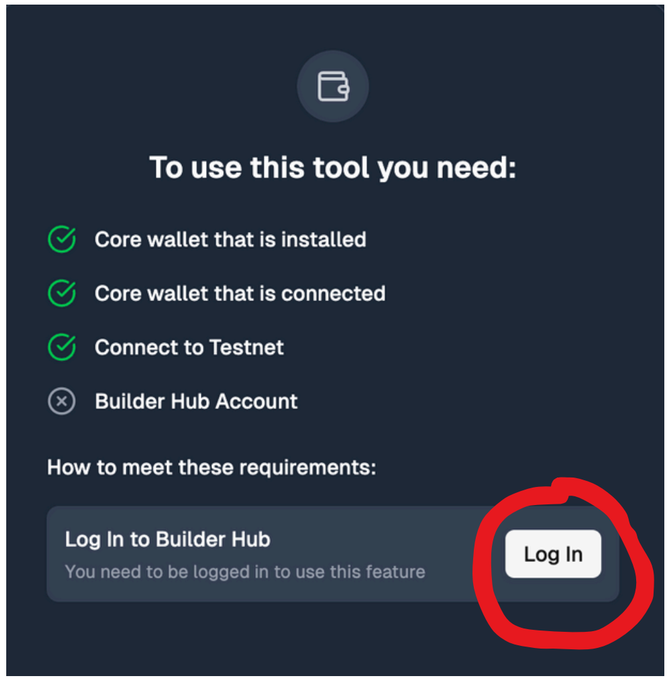


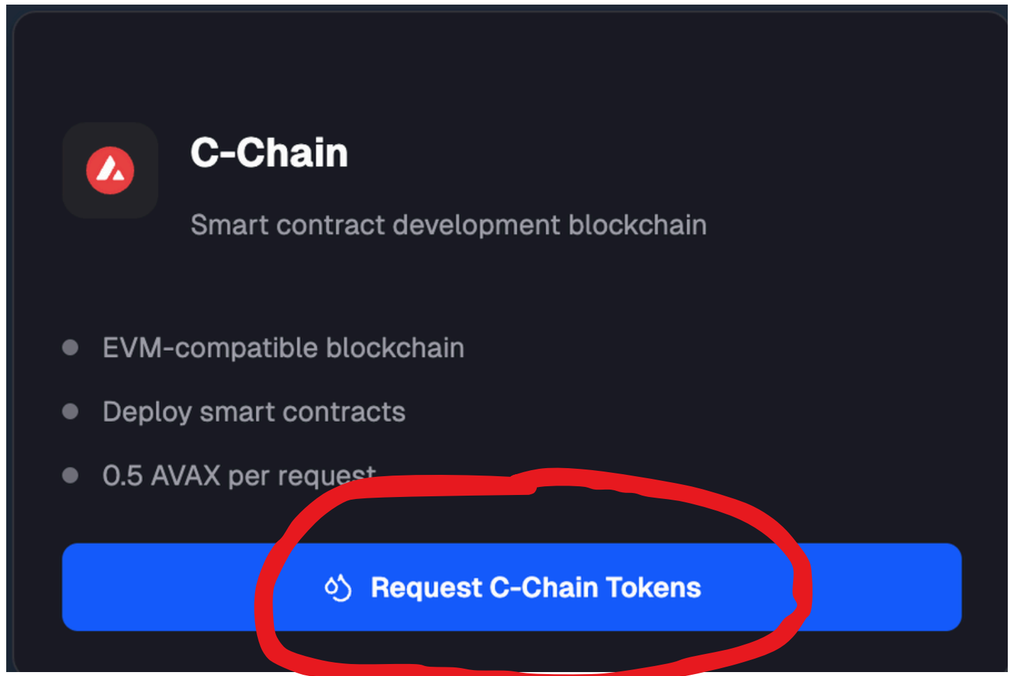
Core Wallet

Come richiedere AVAX di test

Creare un account sul builder hub:
https://build.avax.network/console/primary-network/faucet

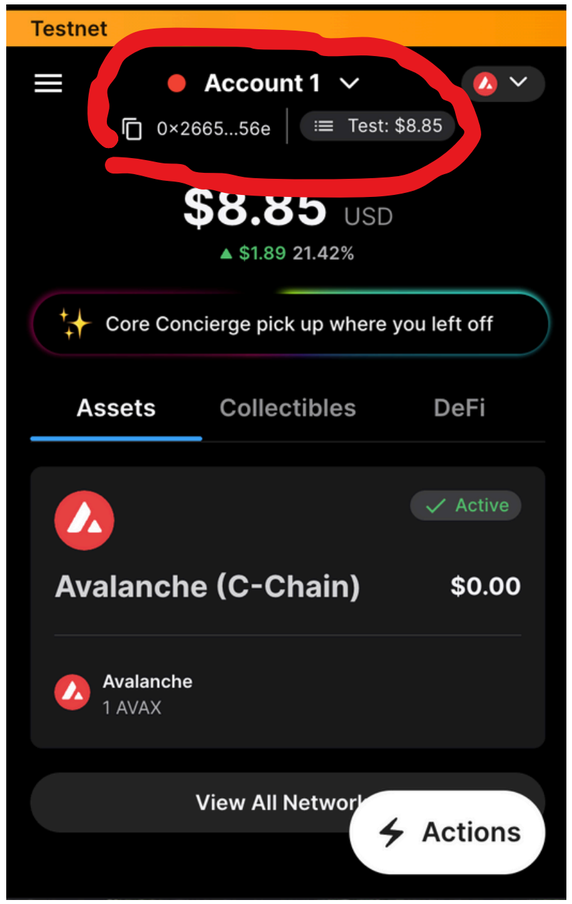


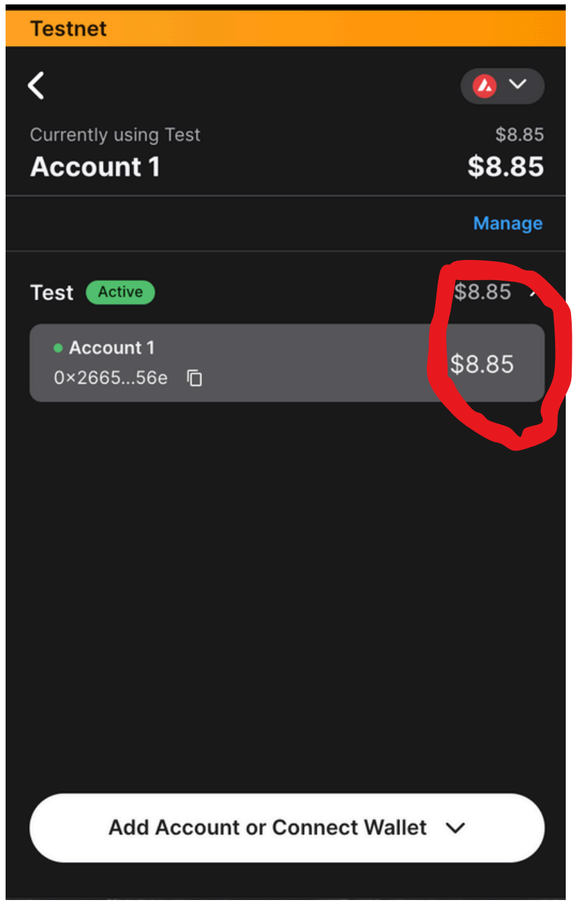


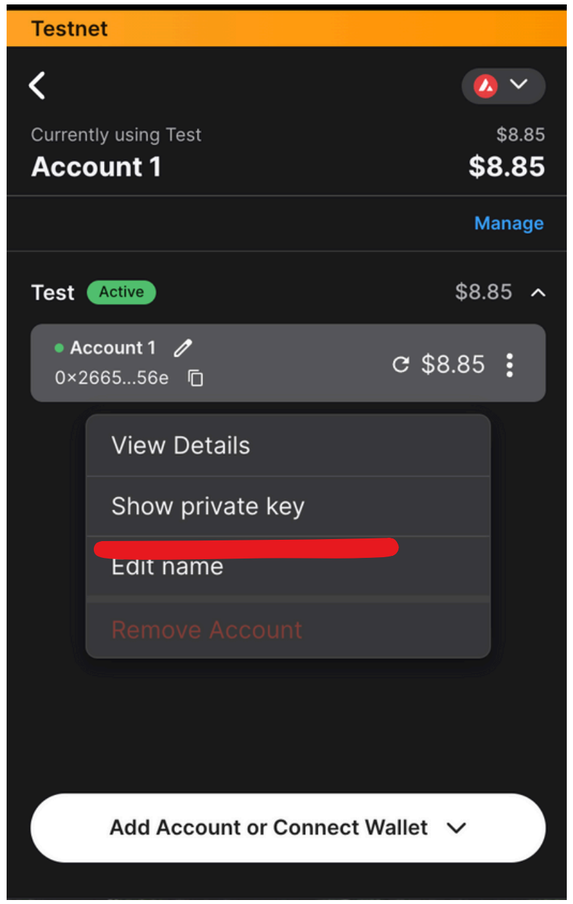


Core Wallet

Creare un core wallet







Iniziamo a lavorare

Istallazione Foundry

Aprite Visual Studio Code: curl -L https://foundry.paradigm.xyz | bash

Riavviate il terminale: source ~/.zshenv

Istallazione di foundry: foundryup

Verifica istallazione:

- forge --version
- anvil --version
- cast --version

Iniziamo a lavorare

Creazione progetto

- Create una cartella vuota: mkdir test_avalanche
- Entrate nella cartella: cd test_avalanche
- Create un nuovo progetto: forge init

Come testare gli script:

forge script script/NOME.s.sol --rpc-url https://api.avax-test.network/ext/bc/C/rpc --broadcast

Domande?

Grazie per la Vostra attenzione