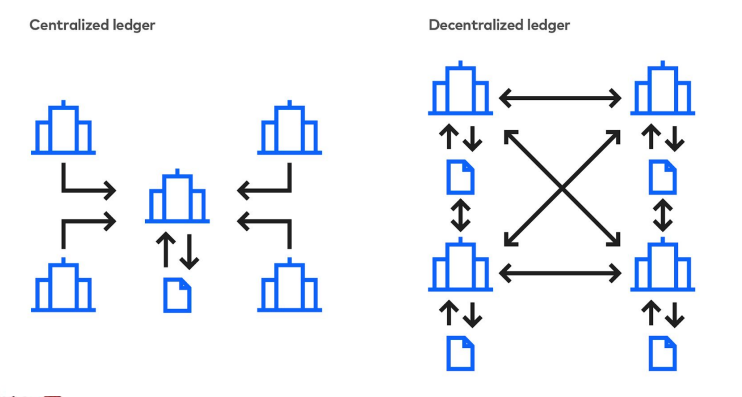
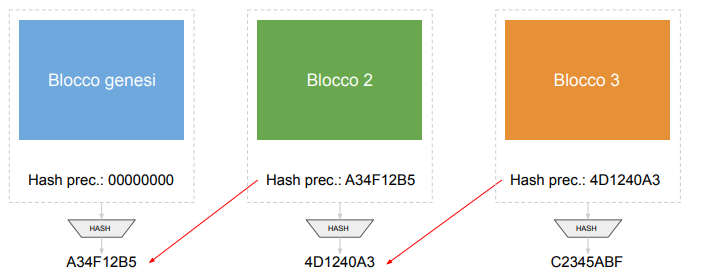
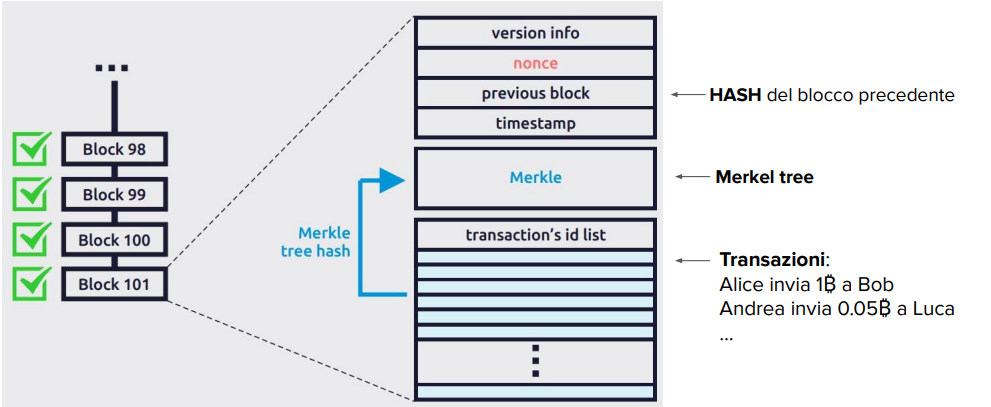
# [Tecnologie Blockchain](https://fad.its-ictpiemonte.it/course/view.php?id=944)

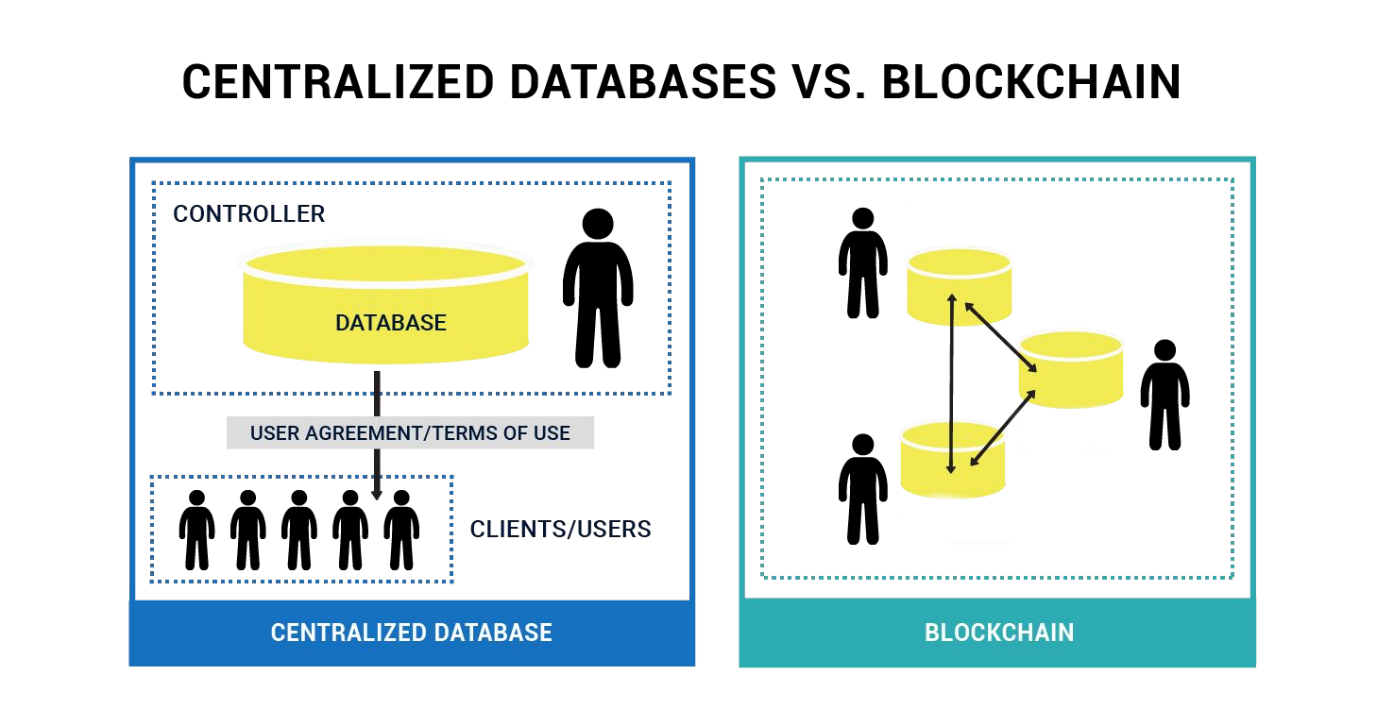
* Catena di blocchi-è un registro condiviso e immutabile dove ogni riga è un blocco di massimo 1MBImmagine che contiene testo, lavagnabianca

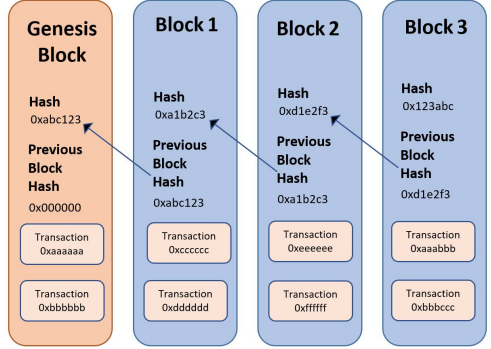
  Descrizione generata automaticamente
* Le informazioni sono pubblici come a chi ha mandato denaro o ricevuto
* “Time stamp” che è avvenuta in quel instante
* La transazione viene scritto nella blockchain ogni 10 minuti
* Ogni nodo ha la stessa copia del registro ed è condiviso tra tutti i nodi
* La rete di nodi è peer-to-peer(fiducia nel protocollo
* Immutabilità(una volta scritto non verrà mai cancellato)
* I blocchi sono collegati attraverso funzioni hash(catena leggere informazioni del blocco precedente
* Riassunto del documento (crittografia)
* Il primo blocco viene chiamato genesi(hash precedente:00000000
* Il hash serve a legare un blocco a quello dopo 
* Cè un articolo dove ha spiegato come funziona
* CYPHERPUNK 1992 movimento anni 80 promuove uso crittografia protezione
* Market cap del bitcoin (₿) è € 378\* miliardi
* BITCOIN=> Bitcoin è la tecnologia bitcoin è la moneta, numero massimo di bitcoin in circolazione 21 Milioni
* 
* MERKLE TREE HASH E UNA SPECIE DI SOMMA DI TUTTE LE HASH TRANSAZIONI
* HASH e una funzione non invertibile(poco probabile)
* Esempio SHA256 più utilizzato
* 

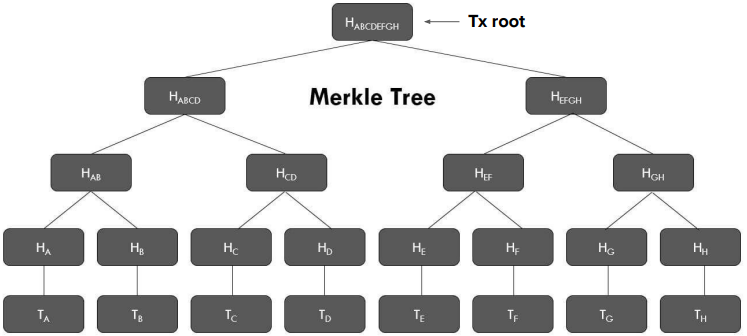
PROPRIETÀ DI HASH

* Facile da calcolare
* Impossibilità di calcolare il messaggio che ha generato lo stesso hash(non invertibilità)
* Modificando il messaggio non deve avere lo stesso hash
* Resistenza ad avere lo stesso hash di un altro

SHA256(è una funzione di calcolo con diverse procedure) da una stringa di 32 caratteri

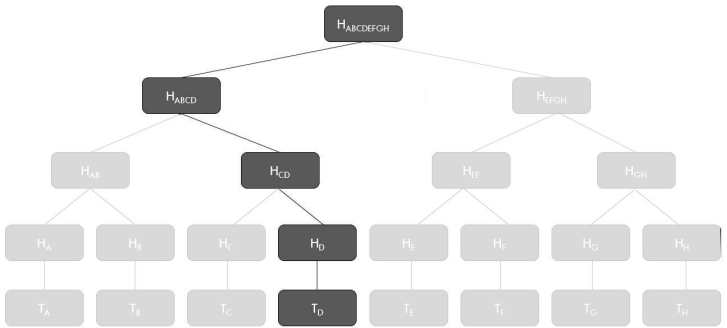


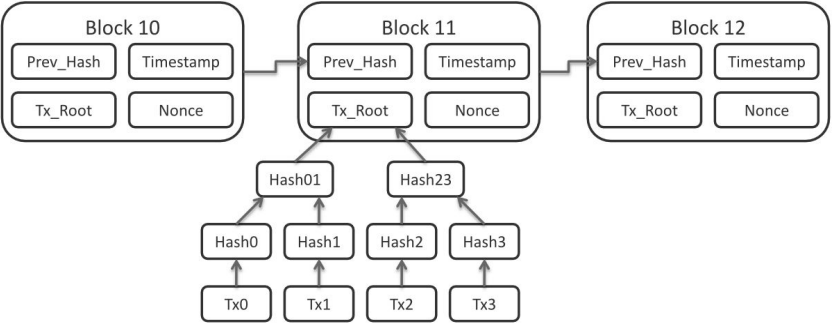


merkle tree(in questo caso è una struttura albero binario)

struttura dati ad albero: perché ogni blocco ha un massimo di 2 figli numero di transizioni deve essere pari

hash,hash,hash,hash e transazione

quando verifica se la transazione e quella giusta fa una ricerca controllatsa dove solo cerca gli hash che coinvolge la transazione



target è un numero prefissato dal protocollo

# BITCOIN

* ERC-721 è uno standar Ethereum per realizzare nft
* Lo standard contiene un’interfaccia da implementare (in Solidity)

Immagine che contiene testo

Descrizione generata automaticamente

Utilizzato per mercato dell’arte collezionismo, aste, verifica di autenticità di un prodotto, mercato immobiliare, brevetti e proprietà intellettuale, votazioni, videogiochi

# PROTOCOLLO BITCOIN

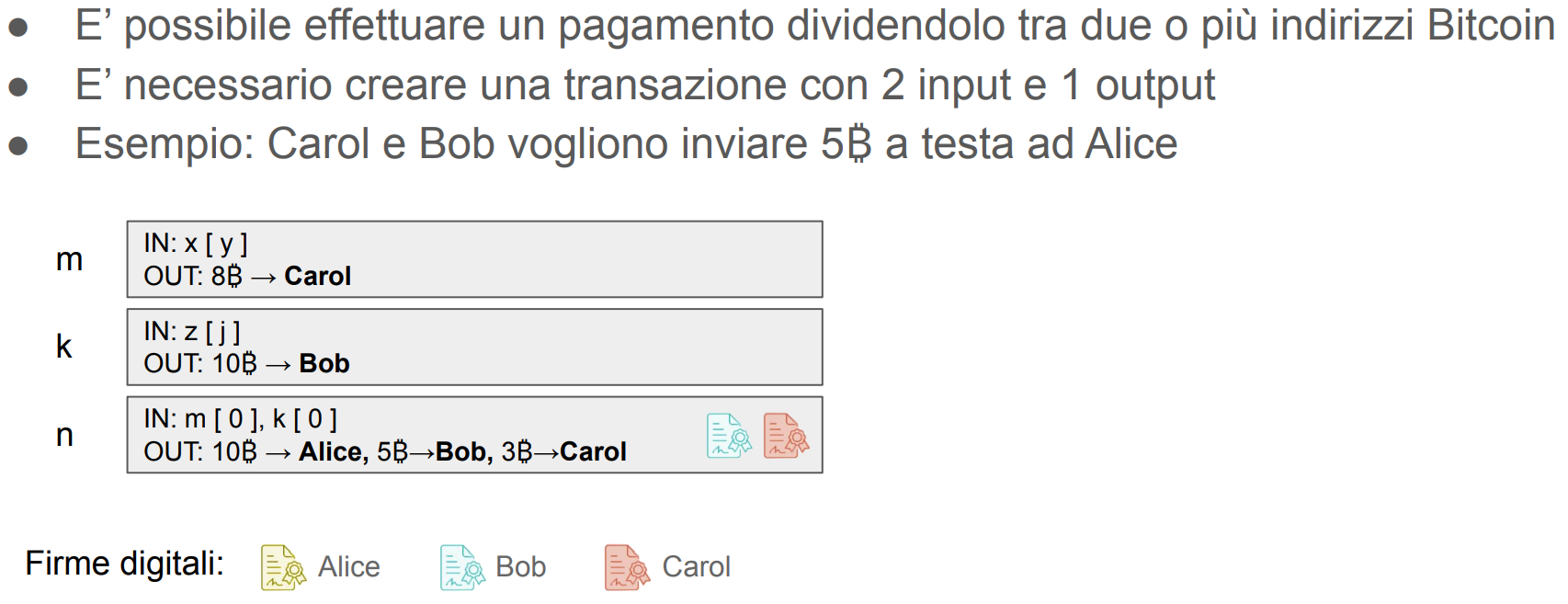
* Insieme di regole che governano la rete bitcoin
* Basato su TCP per comunicare tra nodi
* Algoritmo di mining tramite [Proof of Work](https://it.wikipedia.org/wiki/Proof-of-work)
* Wallet e strumenti crittografici come SHA256, RIPEMD160, ECDSA, etc.
* Software open source disponibile all’indirizzo <https://github.com/bitcoin/bitcoin>
* Risorsa per sviluppatori Bitcoin: <https://developer.bitcoin.org/index.html>

# TRANSAZIONI BASATE SU CONTO(ACCOUNT BASED)

È lingreso creato per quello non cè niente in IN

Il zero e posizione array

# CAMBIO INDIRIZZO

* è possibile creare piu indirizzi Bitcoin(inviare tipo residuo ad un altro indirizzo) **change address**
* verifica efficiente conrtrolla solo le transazioni del blocco non su tutta la cronologia
* posibilita di avere piu input e output(gestire i movimenti a piacimento) 

# DETTAGLIO DI UNA TRANSAZIONE

# Immagine che contiene testo Descrizione generata automaticamenteImmagine che contiene testo Descrizione generata automaticamenteImmagine che contiene testo Descrizione generata automaticamente

* Un blocco ha una dimensione di 1MB(≃2000 transazioni per blocco)

# VALIDAZIONE DI UNA TRANSIZIONE

* La verifica avviene tramite uno script Bitcoinscript **if return true is ok/if return false error**
* **Double-spending è quando si paga due volte lo stesso pagamento**

# BITCOIN SCRIPT

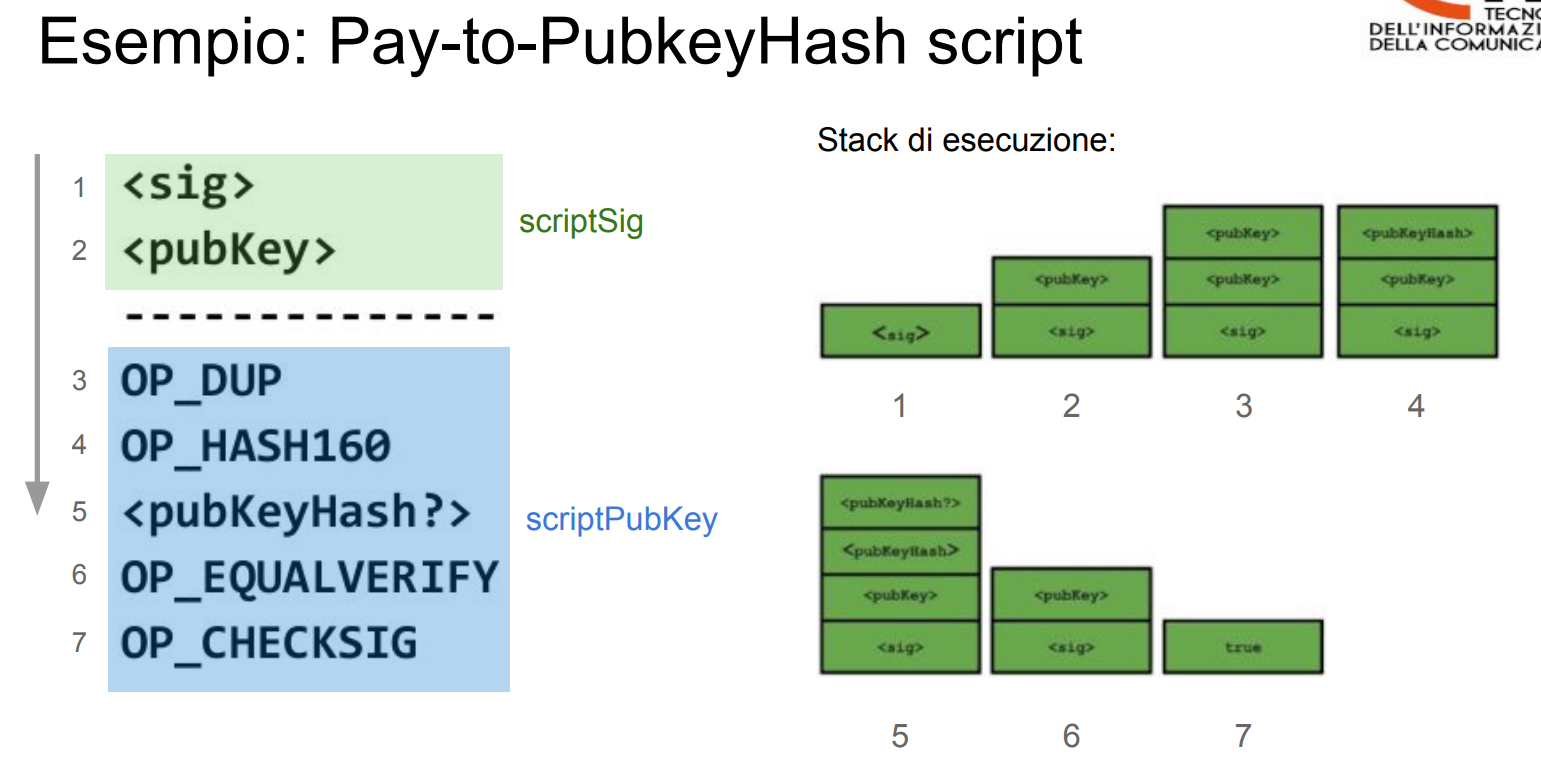
* È un linguaggio nel protocollo Bitcoin per il processo di validazione di un blocco
* Ideato basandosi sul linguaggio [Forth](https://en.wikipedia.org/wiki/Forth_(programming_language)) linguaggio basato su stack(stack-based)pile push(inserir)/pop
* Ogni istruzione viene eseguita in modo lineare cioè senza loop(per evitare loop infiniti)
* Non è turing completo?
* Ha solo 256 istruzioni esprese tramite 1 byte di cui 15 sono disabilitate e 75 riservate per usi futuri cioè sono in totale 166 istruzioni
* Istruzione **CHECKMULTISIG** che consente di verificare firme multiple

Immagine che contiene tavolo

Descrizione generata automaticamente

# ESECUZIONE

* Per eseguire uno script si utilizza uno stack vengono inseriri con push o pop istruzioni o dati
* Linguaggio molto semplice a livello computazionale perche non viene salvato al di fuori dallo stack



<https://siminchen.github.io/bitcoinIDE/build/editor.html>

# TIPOLOGIE DI SCRIPT

script piu utilizzate:

1. **Pay-to-PubkeyHash:** Questo script utilizza una chiave pubblica e richiede la firma della stessa per effettuare una transazione
2. **Proof of burn:** uno script che “distrugge” una transazione. L’implementazione prevede l’utilizzo di un OP\_RETURN. Tutte le istruzioni dopo quella di ritorno non vengono eseguite e quindi possibile scrivere qualsiasi cosa nel ledger
3. **Pay-to-script-hash:** transazione basata sull’hash di uno script al posto di un indirizzo Bitcoin

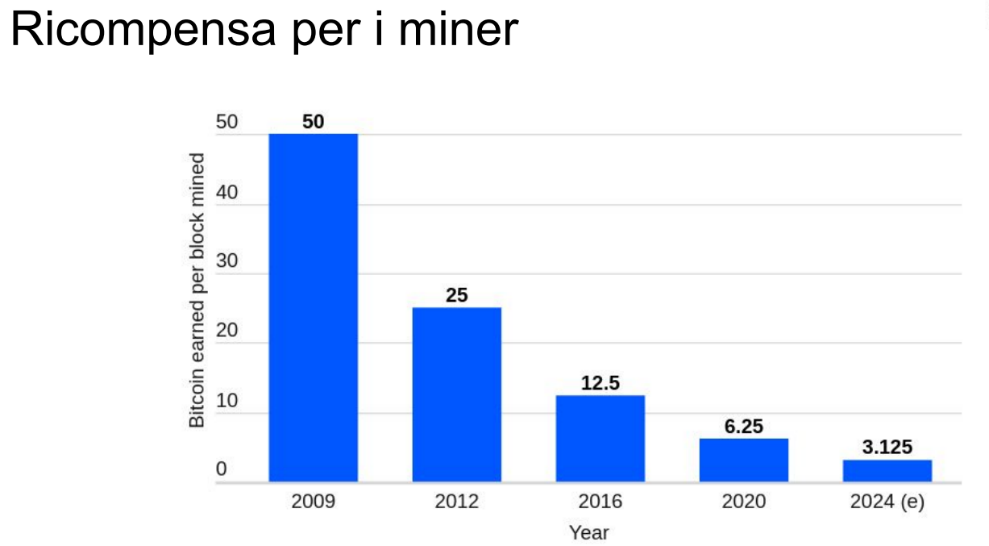
# DEPOSITO A GARANZIAImmagine che contiene testo Descrizione generata automaticamente

# Lock time

# BLOCCHI IN BITCOIN

# COINBASEImmagine che contiene testo Descrizione generata automaticamente

# Blocco genesi

* Il campo coinbase del blocco genesi(primo blocco)3 gennaio 2009 “The Times 03/Jan/2009 Chancellor on brink of second bailout for banks.”

# La rete Bitcoin

# È una rete decentralizzata tipo peer-to-peer

# Tutti i nodi sono equivalenti non ce gerarchia

# Un nodo is collega agli altri in modo casuale

# Quando un nodo si collega alla rete invia un msg ad un seed nod che conosce gia questo risponde con un elenco di nodi a cui è conesso

# Quando c’è un nodo ha un nuovo messaggio lo propaga agli altri nodi di sua conoscenza (algoritmo di flooding o ossip)

# Esempio di transazione

# Alice vuole inviare x ₿ a Bob

# Il client di Alice invia la transazione verso un nodo della rete

# Questo client memorizza la transazione in un memoria temporanea (mempool) e propaga la transazione verso gli altri nodi; se un nodo ha già la transazione nel suo pool non viene più propagata verso altri

# Il mempool contiene dunque le transazioni in attesa di risoluzione del PoW da parte di un miner

# Nodi leggeri(lightweight node)

* Una leightweight node è un nodo che non memorizza lintera blockchain ma soltanto una porzione utile ad eseguire le operazioni correnti (chiamati thin client o Simple Payment Verification (SPV))
* Di solito la dimensione del registro memorizzata in un thin client è di circa 1/1000 del totale della blockchain

# Limiti della rete Bitcoin

# Limiti di transazzioni

# Ogni blocco è di 1MB/ogni transazzione è almeno Un grande limite della rete Bitcoin è legato al numero di transazioni al secondo

# Considerando che un blocco è di 1 MB e ogni transazione è almeno di 250 byte, in un blocco ci possono essere al massimo 4000 transazioni

# Un blocco è inserito nel registro distribuito ogni 10 minuti, quindi circa 7 transazioni al secondo (tps)

# Il circuito delle carte di credito Visa gestisce in media 2000 tps con picchi di 24’000 tps. Paypal gestisce in media 100 tps

# Come evolvere la rete Bitcoin

# Hard fork: ossia delle modifiche sostanziali del codice che fanno deviare la blockchain verso una nuova versione del protocollo non compatibile con la precedente

# Soft fork: introdurre delle modifiche sulla validazione dei blocchi che siano più restrittive. In questo modo i vecchi nodi continuano ad accettare tutti i blocchi mentre quelli nuovi accetteranno solo i nuovi blocchi. Propagando il nuovo software nella rete si può ottenere un effetto di migrazione verso il nuovo

# Ad esempio, l’introduzione del Pay-to-script-hash è stata eseguita grazie ad un soft fork

# Storia dei for di Bitcoin

# 2014: Bitcoin XT, blocchi di 8 MB con 24 tps. Progetto di Mike Hearn, dopo un inizio promettente non è mai decollato ed è stato abbandonato

# 2015: Pieter Wuille, sviluppatore del core di Bitcoin, ha presentato l’idea del Segregated Witness (SegWit) come soft fork. L’idea è quella di ridurre la dimensione di una transazione, spostando la firma della transazione nell’header.

# 2016: Bitcoin Classic, alcuni programmatori entusiasti delle modifiche introdotte con Bitcoin XT hanno ridotto il blocco a 2 MB

# 2017: Bitcoin Cash, hard fork di Bitcoin con blocchi di 8 MB

# Ethereum

# Blockchain pubblica

# Ethereum virtual machine computer che esegue queste applicazioni

# Ogni transazione modifica lo stato della EVM

# Creata da ….

# Proof of Stake

# Nuovo algoritmo di consenso a partire dal 2022.

# Ah sostituito il proof of work con un soft fork denominato The Merge.

# Il proof of stake ha ridotto il consumo energetico necessario per far girare la rete del 99,95%.

# L’idea del proof of stake è riservare temporaneamente un deposito di almeno 32ETH e essere inseriti in coda.

# Quando un validatore è attivato in questa coda esegue la validazione del blocco e invia un voto(Attestato) a favore del blocco.

# Slot ed epoche

# Pow ogni 10 minuto

# Pos tempo fisso e diviso in slot ogni 12 secindi ed epoche 32 slot

# Un nodo validatore viene scelto a caso tra quelli in coda validatori con maggiore riserva avranno più possibilità di essere eletti

# In ogni slot è presente una commissione di validatori preposta

# Ricompensa validatore

# Validatore viene ricompensato apparte mancie

# Se il validatore manomette sulle transazione viene penalizzato, prelevando una percentuale di ETH dal suo deposito inoltre sara escluso per tempo

# Ogni validatore è cosi incentivato a comportarsi correttamente

# Ether

# È lunica forma di pagare le tasse(fee) delle transazioni

# È la moneta piu utilizzata nel mercato DeFi(decentralized finance)

# È lo strumento utilizzato per gas fee(tassa che tutti devono pagare) per lesecuzione di transazioni

# Gli ETH vengono coniati per ogni blocco proposto e per ogni verificata dal comitato di validatori

# Circa 1/8 degli ETH coniati va al validatore scelto a caso e gli altri sono equamente distribuiti tra tutti i participanti al processo di verifica

# Burning ETH

# Burning ether equivale a rimuoverlo permanentemete

# Quando un utente paga il gas fee, questo viene bruciato per far in modo che non venga riutilizzato dai validatori e inoltre aiuta all’economia della moneta

# Dapp

# Applicazione decentralizata(dapp) è una aplicazione creata su una ete distribuita che utilizza uno o piu smart contract e un front end con interfaccia utente

# Carateristiche:

# Decentralizata vengono eseguite nella rete etereum

# Deterministica il risultato dellesecuzione di una dapp deve essere lo stesso a prescindere saòòambiente nel quale viene eseguito

# Turning compoleta,possono fare qualsiasi azione con le risorse adeguate

# Isolata lesecuzione di unapp avviene in un ambiente virtuale denomiato Ethereum Virtual Machine

# Smart contract

# Rapresenta la parte backend di unapp

# È un programa che viene eseguito dalla blockchain ethereum

# Una volta inserito non puo essere modificato

# Pro:

# zero downtime

# privacy

# resistenza alla censura…………………….

# Data integryty

# Trusless computation

# Contro:

# manutenzione

# performance

# congestione della rete…………………………………………………

# esperienza utente

# centralizzazione

# Account

# Externally-owned account:gestiti da qualcuno atrverso chiave private costo zer puo costruito tramite una coppia di chiavi privata/pubblica

# Contract account:smart contract publicato nella rete Ethereum costa perche si usa le risorse della rete per memorizzare

# Fomato degli account

# Patricia trie o radix trie o radix tree

# Strutura di dati che consente di memorizzare delle stringhe(trie=>Retrieval)

# Immagine che contiene testo, dispositivo Descrizione generata automaticamente

# Account ethereum

# 

# Trasazioni

# Un cambio di stato deve essere propagato in tutta la rete

# Tutte le transazioni richiedono una tassa (fee) per essere eseguite

# Le transazioni contengono le seguenti informazioni:

# recipient, indirizzo del destinatario

# signature, firma della transazione da parte del mittente (con la sua chiave privata)

# nonce, un numero incrementale che identifica la transazione per l’account(doppie transazioni)

# value, l’ammontare di ETH oggetto della transazione (espresso in [WEI](https://www.investopedia.com/terms/w/wei.asp))

# data, qualsiasi dato si voglia specificare (opzionale)

# gasLimit, il massimo numero di gas unit che la transazione può consumare

# maxPriorityFeePerGas, il massimo numero di gas come mancia da dare al validatore

# maxFeePerGas, la quantità massimo di gas disponibile per pagare la transazione (inclusa la maxPriorityFeePerGas)

# [Gas](https://etherscan.io/gastracker) Immagine che contiene testo Descrizione generata automaticamente

# Consumo del gas

# 

# Gas refund

# 

# Blocchi

# A blocchi (decine, centinaia)cosi tutti i nodi si sincronizzano(12 secondi)

# Dove sono memorizzati i dati?

# 

# Etherscan.io

# Gli smart contract vengono scritti in un linguaggio ad alto livello, come Solidity e compilati in opcode

# Solidity Smart-contract

# Linguaggio a oggetti con tipizzazione statica {} ideato per etherum nato il 2014 Gavin Wood

# Quando si sviluppa in solidity si deve per forza sempre usare l’ultima versione porche ha le [security fixes](https://github.com/ethereum/solidity/security/policy#supported-versions) (tranne casi speciali) attualmente versione 0.8.17

# È una collezione di funzioni e dati che risiedono in uno specifico indirizzo della blockchain ehtereu

# Immagine che contiene testo Descrizione generata automaticamenteImmagine che contiene testo Descrizione generata automaticamente

# Costruttti di base del linguaggio

# Immagine che contiene testo Descrizione generata automaticamente

# Funzioni modifier

# Sono funzioni speciali che cocndizionano lessecuzionen del codice

# Dsds

# Hyperleadger

# Progetto opensource della [linux Fondation](https://www.linuxfoundation.org/)(2015) propone soluzioni in campo aziendali(piu utilizzata)(membri:Accenture, IBM, JP Morgan, Hitachi, Fujitsu,etc…)sono principalmente permissioned

# 

# Tipi:

# Permissionless:senza permessi publicche etherum bitcoin

# Permessioned:necessitano permessi private lan azienddali

# Caratteristiche:

# I partecipanti devono essere identificati/identificabili

# La rete deve essere di tipo permissioned

# Supportare un elevato numero di transazioni

# Bassa latenza nella conferma di una transazione

# Sicurezza e privacy delle transazioni

# Permissioned:

# prevede lutilizzo di autenticazione,meno rischio di avere malintenzionati, proprieta identiche a permissionless: registro temporale,inmutabilita,condivisione

# Caratteristiche:

# Hyperleadger Fabric:

# framework blockchain standar per aziende scritto in GO svilupato principalmente per aziende

# Pluggable consensus protocol

# Problema dei generali bizantini in questo caso è meno sicura in quanto se il numero di processi non corretti è maggiore o uguale a un terzo, ma ha anche la asicurazine della autenticazione

# Smart Contract(chaincode):

# utilizza il principio execute-order-validate al posto di order-execute :non avviene la verificazione ad ogni execute in quanto si fiida dellautenticazione e non ce il gas(tasse)

# Assets:gli oggetti ceh vogliono gestire nella blockchain

# Chaincode:

# Ledger:

# Privacy:

# Secureity & Membership:

# Consenus: