Blockchain

Rocco Lo Russo
roc.lorusso@studenti.unina.it

Università di Napoli Federico II — 28/12/2025

Introduzione

In questo documento verrà approfondita la blockchain, seguendo il materiale didattico fornito dal professor Stefano Russo durante il corso di Sistemi distribuiti e IOT erogato nell'anno accademico 2025/26 presso l'università degli studi di Napoli Federico II.

1 Introduzione alla blockchain

Il problema fondamentale che si vuole risolvere consiste in due nodi di una rete che intendono scambiarsi un bene, in maniera affidabile e sicura. Il *modello dei guasti* prevede:

1. Fallimenti dei nodi, di tipo crash o hang;
2. Fallimenti dei canali, di tipo crash o intermittente;
3. Malfunzionamenti della rete: perdita, alterazione o ritardo anomalo di pacchetti, partizionamento della rete.

Info: Con fallimento di tipo **hang** si indica una situazione in cui il nodo, pur non essendosi arrestato (il processo è ancora attivo), si "blocca" o si congela. In questo stato, il nodo non riesce a progredire nell'esecuzione del protocollo o a rispondere in tempi utili, pur non essendo tecnicamente crashato.

Il *modello degli attacchi* invece prevede:

1. Replay attack: una transazione valida viene maliziosamente ripetuta o ritardata;
2. Men-in-the-middle attack: la comunicazione è osservata o alterata da una terza parte non autorizzata;
3. Masquerade Attack: un nodo malizioso utilizza un'identità forgiata appositamente per la comunicazione, compreso il caso in cui un nodo assume maliziosamente l'identità di un altro nodo della rete;
4. Comportamento bizantino: un nodo assume un comportamento non previsto dal protocollo.

1.1 Soluzione classica

La soluzione comunemente adottata prevede che la gestione delle transazioni venga realizzata impiegando un'entità di terze parti, come ad esempio una banca, che *convalida*, *supervisiona* e *preserva* le transazioni. Tale soluzione può essere implementata ad esempio mediante il 2PC, strumento attraverso il quale l'entità centrale garantisce la consistenza e la validità delle operazioni tra i nodi, fungendo da garante unico. La presenza di un mediatore permette di gestire attacchi di tipo *Masquerade* o *Bizantini* centralizzando la fiducia. Il sistema richiede esplicitamente che la terza parte sia fidata ed affidabile. Nei riguardi del masquerade attack, il mediatore agisce come autorità di certificazione. Le tecnologie basate su trusted entities utilizzano la firma digitale e le autorità di certificazione per garantire l'autenticazione dei nodi e l'integrità dei messaggi. In questo modello, è la trusted entity a creare e gestire le coppie di chiavi (pubbliche e private), impedendo a un nodo malizioso di falsificare la propria identità o quella di altri,

poiché l'identità è garantita dall'ente centrale. Nei riguardi dei comportamenti bizantini, la soluzione con mediatore risolve questo problema rimuovendo la necessità di consenso distribuito tra i nodi paritari; è il mediatore unico a decidere quale transazione è valida. Poiché il mediatore è assunto come "fidato", la rete non deve preoccuparsi del comportamento traditore dei singoli partecipanti, in quanto l'unica verità accettata è quella validata dal mediatore stesso. Pur risolvendo problemi legati alla sicurezza, questa architettura centralizzata presenta i classici problemi di bottleneck prestazionale e singolo punto di fallimento.

1.2 Soluzione alternativa

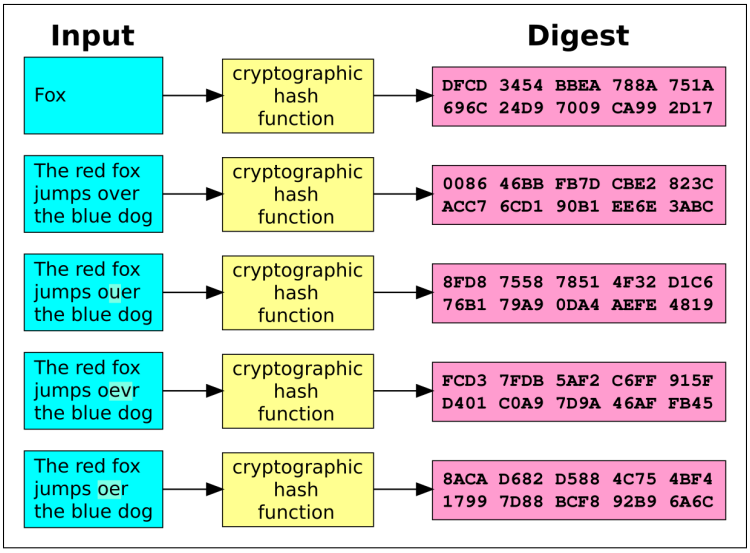
Per inviare un bene senza la collaborazione di un'entità terza, si può definire un protocollo per eseguire transazioni affidabili e sicure in un ambiente *inaffidabile* (si veda il modello dei guasti) e *insicuro* (si veda il modello degli attacchi). Tra le varie problematiche legate a questa soluzione vi sono:

- Verifica dell'integrità del messaggio ricevuto;
- Verifica dell'identità dei nodi partecipanti;
- Verifica della validità delle transazioni;
- ...

Problematiche queste che vengono agevolmente risolte con la soluzione classica. L'idea dl Bitcoin e della Blockchain, inizialmente (2008) applicata in ambito finanziario, era proprio quella di effettuare transazioni in un'infrastruttura non sicura, con il vantaggio di superare la necessità della collaborazione di *trusted entities*. Componente fondamentale della blockchain è il **ledger** (libro mastro), ovvero un registro in cui sono contenute tutte le transazioni eseguite dai partecipanti di un sistema. Si tratta di un *log* di transazioni, append-only, ordinato cronologicamente, in cui le scritture sono sistematiche, ovvero vengono registrate in base all'oggetto a cui si riferiscono. In un ledger distribuito, ciascun nodo della rete memorizza record in copia locale del libro in maniera sequenziale, sotto forma di *blocchi* ordinati temporalmente. Affinchè i *blocchi* vengano considerati validi, i partecipanti devono raggiungere un quorum di consenso. Il libro mastro è costruito come una catena di blocchi, ed ogni blocco contiene un'informazione riguardo il blocco precedente della catena.

Info: Genesis block: è l'unico blocco della catena che non contiene informazioni (hash) riguardo al blocco precedente, poiché non esiste nessun blocco prima di esso.

Un blocco altro non è che un insieme di transazioni.



Info: Mentre una normale funzione hash potrebbe tollerare collisioni (due input diversi che danno lo stesso output), le funzioni crittografiche cercano di rendere computazionalmente inammissibile trovare due sequenze diverse che producano la stessa impronta. Questo concetto è legato al "Paradosso del compleanno", dove la probabilità di collisione esiste statisticamente, ma deve essere resa impossibile da sfruttare nella pratica.

Ogni blocco, ad eccezione del genesis, riporta un set di transazioni e il valore hash del blocco precedente. La modifica di un blocco della catena da parte di un nodo, comporterebbe la generazione di un valore hash differente. In questo modo, è possibile che gli altri nodi verifichino l'integrità dei blocchi, ed è possibile che i nodi rifiutino modifiche a transazioni già validate. Il contenuto del *ledger* è reso immutabile.

Sebbene le tecnologie blockchain siano nate per fare a meno di trusted entities, è necessario comunque utilizzare firme digitali e trusted authorities di certificazione, per far fronte a problemi di sicurezza tipici dei sistemi distribuiti come:

- Autenticazione dei nodi nella rete;
- Integrità: Verifica di violazioni dell'integrità dei messaggi scambiati;
- Non ripudiabilità da parte dei nodi dei messaggi inviati sulla rete.

Info: Crittografia Simmetrica - è il metodo più antico e semplice ("a chiave segreta"). Si basa su un'unica chiave condivisa tra mittente e destinatario. La stessa identica chiave (password) viene usata sia per chiudere (cifrare) che per aprire (decifrare) il messaggio. **Crittografia Asimmetrica**: è la soluzione moderna al problema dello scambio delle chiavi ("a chiave pubblica"). Si usano due chiavi diverse ma matematicamente collegate. La chiave pubblica è nota a tutti. Serve per cifrare i messaggi destinati a te o verificare la tua firma; la chiave privata è segreta, serve per decifrare i messaggi ricevuti o per firmare. Ciò che viene cifrato con una chiave può essere decifrato solo dall'altra chiave. Il collegamento tra chiave pubblica e privata è legato alla teoria dei numeri, e viene fatto ampio utilizzo di algoritmi basati su artimetica modulare. Una trusted entity crea ad esempio la coppia di chiavi (es. HTTPS).

1.3 Problema del double spending

Consideriamo un nodo che vuole vendere un bene attraverso una transazione. Il nodo predispone autonomamente la transazione, che deve sottoporre agli altri nodi affinchè sia approvata da una maggioranza. È possibile che il nodo provi a spendere due volte lo stesso bene, sia in maniera maliziosa (comportamento bizantino) che in maniera inconsapevole (malfunzionamenti della rete, replay attack). Il primo passo per risolvere questo problema è determinare se un messaggio è *fresco* oppure è una copia ritrasmessa. È necessario poter identificare univocamente un messaggio contenente un blocco di transazioni eseguite. Si può utilizzare un **nonce**, definito come "un valore tempo variabile che ha al più una possibilità trascurabile di ripetersi", come ad esempio un valore random generato da capo per ogni utilizzo, come un timestamp, un numero seriale o una combinazione di questi. Ogni blocco può essere identificato da un nonce. Per garantire che ogni blocco sia caratterizzato da un valore differente, è necessario che tale valore sia generato da una fonte *trusted*.

È necessaria inoltre una certificazione del tempo, utile a verificare quando un documento è stato creato, oppure quando è stata apportata l'ultima modifica in maniera affidabile e non falsificabile. Per ottenere questa certificazione si può utilizzare un **Trusted Time Server** (TTS), con due possibili soluzioni:

1. L'utente invia al TTS una copia del documento; il TTS aggiunge data e ora, e conserva il documento.
2. L'utente invia al TTS l'hash del documento; il TTS aggiunge data, ora e firma, e invia il certificato all'utente.

La seconda soluzione è utile perchè protegge la privacy del documento, e riduce i costi di trasmissione e memorizzazione. Indipendentemente dalla correttezza dei tempi certificati, si vuole che due documenti sottoposti sequenzialmente al TTS abbiano un nonce che esprima questa sequenzialità. Ogni certificato rilasciato contiene (hash del documento, data, ora, firma TSS, hash certificato precedente). Complessivamente, i certificati formano una catena di blocchi. È inammissibile inserire successivamente un certificato all'interno della catena, in quanto il TTS dovrebbe rilevare collisioni per la funzione di hash. La certificazione nel tempo comporta il vantaggio che l'hash è più resistente alle collisioni, e risolve il double spending assegnando un ordine temporale alle transazioni, pubblicandole.

1.4 Transazioni con consenso distribuito

Tutti i nodi della rete sono a conoscenza di tutte le transazioni, e partecipano attivamente alla validazione dei blocchi attraverso il raggiungimento del consenso.

Warning: Il modello di sicurezza deve contemplare anche il cosiddetto **51% attack**, che consiste in un gruppo di nodi maliziosi ($\geq 50\%$), che potrebbero compromettere maliziosamente il consenso.

È fondamentale ricordare che a causa del teorema di Fisher per i sistemi asincroni, per raggiungere il consenso in presenza di fallimenti di tipo crash si possono rilassare i vincoli di consenso, e rendere meno asincrono il sistema, sfruttando periodi di sincronia. Il rilassamento dei vincoli consiste nel consentire che in alcuni casi non si raggiunga il consenso tra tutti i nodi della rete, ovvero contemplare il fatto che solo un sottoinsieme di nodi raggiungano il consenso, e si *tollera* il 51% attack. Esistono delle soluzioni diverse che consentono di irrobustire il consenso, per evitare questo tipo di attacchi. Ad esempio, chi valida un blocco deve spendere del tempo computazionale tale da scoraggiare l'attacco, con tanto di meccanismo di incentivazione (ricompensa per il lavoro svolto).

1.5 Funzionamento della blockchain

Una blockchain è un *ledger pubblico* distribuito di transazioni o eventi digitali eseguiti e condivisi tra i partecipanti. Ogni transazione riportata nella blockchain è validata mediante il raggiungimento del consenso tra i nodi del sistema. Una volta memorizzate, le informazioni non possono essere ne' cancellate, ne' modificate. Ogni blocco contiene una o più transazioni (record). La storia dei blocchi è immutabile e può essere verificata a partire dal primo blocco.

2 Classificazione delle blockchain

Le proprietà di una blockchain 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 partecpante del sistema;
- Assenza di una Trust Anchor.

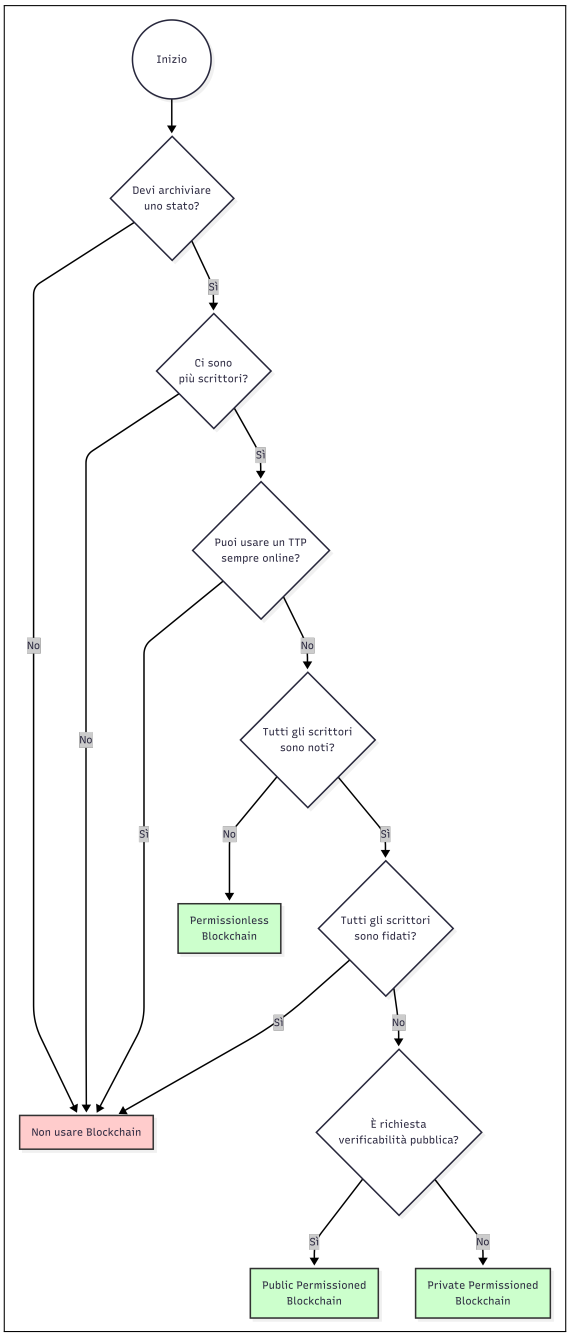
Info: Trust Anchor: rappresenta un'entità centrale o una "terza parte" (come una banca o un'autorità governativa) che detiene il controllo esclusivo sui dati e sulle transazioni, fungendo da garante unico della loro validità.

L'effettiva implementazione della tecnologia blockchain dipende fortemente dal tipo:

1. **Permissionless**: i partecipanti non devono preventivamente essere autorizzati per il ruolo che intendono svolgere;
2. **Permissioned**: le operazioni (tutte o solo alcune) devono essere svolte solo da nodi preventivamente autorizzati.

Un' ulteriore distinzione può essere effettuata per quanto riguarda l'ambito di utilizzo:

1. Public: tutti i blocchi sono visibili a tutti i nodi e ogni nodo può partecipare al consenso;
2. Private: una specifica **organizzazione** può decidere quali nodi possono leggere i blocchi e partecipare al consenso;
3. Consorzio: pochi nodi predeterminati possono leggre i blocchi e partecipare al consenso.



3 Permissionless Blockchain

3.1 Bitcoin

Bitcoin è una cryptocurrency, definita come *un conio digitale nel quale tecniche di crittografia sono usate per regolare la generazione di unità e verificare il trasferimento di fondi, operando indipendentemente da una banca centrale*.

Info: Una criptovaluta è una risorsa digitale o una rappresentazione virtuale di valore che non esiste in forma fisica e che utilizza tecniche avanzate di crittografia per rendere sicure le transazioni, controllarne la creazione e verificarne il trasferimento. Secondo le definizioni fornite da istituzioni come la Banca Centrale Europea e il Fondo Monetario Internazionale, essa si distingue radicalmente dalle valute tradizionali, definite fiat, perché non è emessa né garantita da una banca centrale o da un'autorità pubblica, ma si basa su protocolli informatici condivisi tra una moltitudine di partecipanti. In sintesi, la criptovaluta rappresenta il primo esperimento su scala globale di un sistema finanziario *disintermediato*, dove la sovranità monetaria è sostituita da un algoritmo e la trasparenza è garantita dalla condivisione capillare dei dati tra tutti i membri della rete.

Le transazioni di Bitcoin avvengono su reti P2P, in cui i peer monitorano e verificano ogni trasferimento delle monete, attraverso un'infrastruttura basata su blockchain. Il design di bitcoin è pubblico, ed è un progetto open: nessuno possiede o gestisce Bitcoin, e tutti possono farne parte. Per svincolarsi dall'utilizzo di un TSS (timestamp server), Bitcoin sfrutta un meccanismo differente per la generazione di **nonce**: il **Proof-of-Work** (PoW). PoW è il meccanismo di consenso originale utilizzato nella tecnologia blockchain, per convalidare le transazioni e garantire la sicurezza della rete senza necessità di autorità centrale. La PoW è la risoluzione di un enigma matematico che i computer della rete, denominati *miners*, devono risolvere per avere il diritto di aggiungere un nuovo blocco alla blockchain. Questo enigma è progettato per essere onereso dal punto di vista computazionale, ma estremamente facile da verificare per tutti gli altri partecipanti. Di fatto, nella tecnologia Bitcoin la PoW consiste nel calcolare un valore hash tale da avere un determinato numero di bit iniziali pari a 0. Ogni nodo, cerca di trovare un nonce da codificare insieme al blocco tale da soddisfare il vincolo sul numero di zero prodotti dalla funzione hash. Cambiando il Nonce anche di una sola cifra, l'hash dell'intero blocco cambia completamente. Il minatore prova miliardi di combinazioni di Nonce al secondo. Il lavoro medio richiesto è esponenziale rispetto al numero di bit zero richiesti e può essere verificato eseguendo un singolo hash. Poichè l'operazione è molto onerosa dal punto di vista computazionale, sono disincentivati gli attacchi 51%.

L'algoritmo di Bitcoin è, per grandi linee, il seguente:

1. Ogni nuova transazione viene inviata in broadcast a tutti i nodi. Praticamente una transazione consiste nell'invio di bitcoin. Questa transazione viene inoltrata a tutti i nodi della rete P2P;
2. Ogni nodo che svolge il ruolo *miner* colleziona le nuove transazioni in un blocco; ogni miner può scegliere transazioni diverse, in generale scelgono quelle che offrono commissioni più alte;
3. Ogni nodo *miner* lavora per trovare una Proof-of-Work per il suo blocco;
4. Non appena un miner fortunato o potente trova la soluzione, interrompe immediatamente il lavoro e trasmette il blocco completo di soluzione a tutta la rete. È il momento in cui il miner *rivendica* il diritto di aggiungere la pagina successiva al ledger e di ricevere la relativa ricompensa;
5. I nodi *miner* riceventi non si fidano ciecamente del vincitore. Eseguono istantaneamente dei controlli:
    - Verificano che la PoW sia corretta (operazione istantanea);
    - Controllano che ogni transazione nel blocco sia autentica (firme digitali corrette);
    - Si assicurano che non ci sia **double spending**, ovvero che per ogni transazione non siano stati spesi gli stessi beni già spesi in un blocco precedente.
6. L'accettazione non avviene tramite un voto esplicito, ma in modo implicito e operativo. I nodi scrivono il nuovo blocco nella loro copia locale del registro e iniziano subito a lavorare al blocco

successivo. Il legame tra i blocchi è garantito dal fatto che il nuovo blocco dovrà contenere l'impronta digitale (hash) del blocco appena accettato. Se un miner provasse a lavorare su una versione diversa della storia, la sua catena diventerebbe più corta e verrebbe infine scartata dalla rete,

Può accadere che due nodi diversi trovini contemporaneamente una soluzione per la PoW e trasmettano in broadcast due blocchi differenti nello stesso momento; A causa della latenza di rete, alcuni nodi riceveranno prima uno e poi l'altro blocco. Ogni nodo inizia a lavorare sul primo blocco che riceve, considerandolo valido per la propria catena locale, ma conserva comunque l'altro ramo in memoria nel caso quello diventi il ramo principale in futuro. L'incertezza su quale sia la catena *vera* si risolve nel momento in cui uno dei due rami diventa più lungo dell'altro. Questo accade quando un miner trova la sulezione per il blocco successivo basandosi su uno dei due rami contendenti. I nodi che stavano lavorando sul ramo più corto lo abbandonano, e si spostano immediatamente sulla catena più lunga per proseguire il lavoro. Quanto presentato è noto come **meccanismo di autocorrezione** della rete bitcoin.

È interessante una considerazione sulla scalabilità di bitcoin.Ad ogni passo la blockchain aggiunge un ulteriore blocco, ogni blocco aumenta con i dati rappresentativi dei precedenti. Man mano che un maggior numero di utenti si unisce alle rete e i blocchi aumentano, il sistema rischia di deformarsi. Inoltre, dato che ogni nodo conosce le transazioni eseguite dagli altri, in quanto deve verificarne la validità, vengono utilizzati degli pseudonimi per garantire la pricacy degli utenti.

4 Permissionless blockchain e smart contracts

Uno smart contract è un'applicazione informatica strettamente legata alla tecnologia Blockchain, progettata per facilitare, verificare o far rispettare la negoziazione e l'esecuzione di un accordo. Si presenta sotto forma di codice, che risiede e viene eseguito direttamente all'interno della blockchain.

4.1 Ethereum

Si tratta della più grande piattaforma software decentralizzata e open source che consente lo sviluppo di *smart contracts* e applicazioni decentralizzate senza entità terze. I contratti possono essere utilizzati in maniera sicura per eseguire un basto numero di operazioni. La valuta digitale gestita è l'Ether. Ogni operazione richiede una commissione (*gas*). Il costo del gas è espresso in *wei* ($\approx 10^{-18}$Ether). Una **Transazione** è un messaggio firmato che esegue una determinata operazione associata alla blockchain. Nel caso di criptovalute si tratta di inviare una certa quantità di valuta ad altri nodi della rete, ma altre transazioni possono essre ad esempio una registrazione dei nomi di dominio, la realizzazione e l'adempimento di offerte commerciali e la stipula di contratti. Lo **Stato** è un set di dati di una blockchain network deve assolutamente tenere traccia, e rappresenta i dati attualmente rilevanti per le applicazioni implementate sulla catena. Lo stato è costituito da un insieme di oggetti denominati *Accounts*. L'esecuzione di una transazione comporta un cambiamento di stato. Gli Accounts possono essere di due tipi:

- **Externally owned accounts** (EOA), in grado di inviare e ricevere Ether, inviare transazioni agli smart contract;
- **Contract accounts**, che oltre alle funzionalità degli EOA hanno del codice associato, le loro azioni vengono "triggerate" da EOA o altri Contract accounts, e l'esecuzione del codice modifica le informazioni contenute nel proprio spazio di archiviazione.

I contratti hanno generalmente quattro scopi:

1. Gestire un data store, che rappresenti qualcosa di utile per altri contratti o per il mondo esterno;
2. Comportarsi come un EOA con politiche di accesso più complesse;
3. Gestire un contratto o una relazione in corso tra più utenti;
4. Fornire funzionalità ad altri contratti.

Gli smart contracts sono solitamente scritti in linguaggi di programmazione di alto livello. Il codice viene compilato dalla Ethereum Virtual Machine e viene deployato nella blockchain sotto forma di EVM

bytecode. L'EVM consente a chiunque di eseguire l'EVM byte code. Come per Bitcoin, il consenso, almeno nella versione 1.0, è basato sulla PoW, con la differenza che l'algoritmo per il calcolo del nouce converge più velocemente, in quanto la complessità è spostata dalla computazione all'utilizzo di memoria.

5 NFT

Un **NFT**, acronimo di **Non-Fungible Token**, è un tipo di token crittografico utilizzato per identificare qualcosa o qualcuno in modo unico e non intercambiabile all'interno di una blockchain.

5.1 Concetto di Fungibilità

Per comprendere gli NFT, è fondamentale distinguere tra beni fungibili e non fungibili:

- **Beni Fungibili:** Sono mutuamente intercambiabili, come il denaro o il carburante. Se si presta una banconota da 10 euro, non è rilevante riceverne indietro una differente, purché il valore nominale sia lo stesso.
- **Beni Non Fungibili:** Sono unità uniche che non possono essere sostituite l'una con l'altra, come opere d'arte, case o auto usate. In questo caso, la restituzione deve riguardare esattamente l'oggetto originale.

5.2 Caratteristiche Tecniche e Standard

A differenza delle criptovalute comuni come Bitcoin o Ether (che sono divisibili), gli NFT sono generalmente **indivisibili** e devono essere trasferiti come un'unica entità. Sulla rete Ethereum, gli standard principali sono:

- **ERC-721:** Lo standard specifico per i token non fungibili, che definisce un'API per gestire la proprietà univoca di un asset all'interno di Smart Contract.
- **ERC-1155:** Uno standard multi-token più efficiente che permette di gestire combinazioni di token fungibili e non fungibili (o semi-fungibili) all'interno dello stesso contratto.

5.3 Storia ed Evoluzione

Sebbene il grande pubblico li abbia scoperti nel 2017 con la diffusione di collezioni sulla blockchain di Ethereum, la storia degli NFT è iniziata nel **2014** con "Quantum", creato da Kevin McCoy sulla blockchain Namecoin. Successivamente, Ethereum ha offerto una soluzione più flessibile per la creazione, programmazione e scambio di questi asset, abbassando le barriere all'entrata per sviluppatori e artisti.

5.4 Casi d'Uso Principali

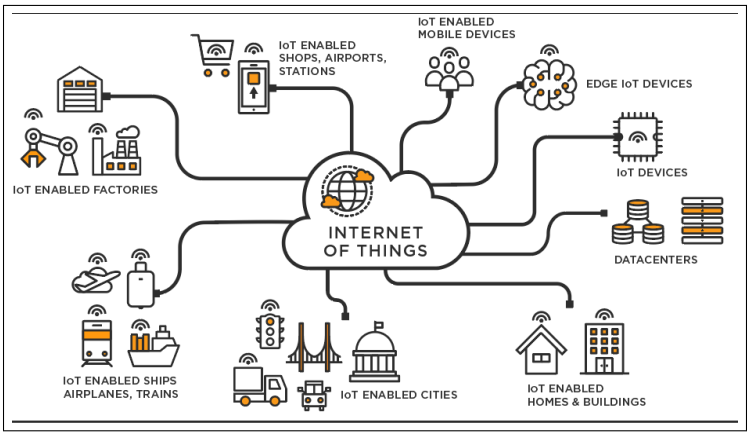
Gli NFT trovano applicazione in numerosi settori:

- **Arte Digitale:** Consente la vendita e l'asta di opere certificate digitalmente, garantendo la rarità del bene.
- **Gaming:** Utilizzati per fornire proprietà reale su oggetti di gioco e personaggi collezionabili univoci.
- **Real Estate:** Per la tokenizzazione di proprietà immobiliari, semplificando i passaggi di proprietà tramite smart contract.
- **Biglietteria:** Per prevenire il bagarinaggio e le frodi nella vendita di biglietti per eventi, assicurando l'autenticità del titolo di accesso.
- **Identità e Compliance:** Utilizzati per la conformità KYC (*Know Your Customer*), associando un token univoco all'identità verificata di un utente.

Nota conclusiva: L'NFT non è il file digitale stesso, ma un certificato di proprietà digitale registrato su blockchain che punta a tale risorsa, rendendo la sua provenienza e la sua storia pubblica e inalterabile.

6 Blockchain e IOT

L'Internet of Things (IoT) è descritto come una rete di portata mondiale composta da oggetti interconnessi che sono indirizzabili in modo univoco e basati su protocolli di comunicazione standard. Da una prospettiva generale, esso rappresenta un gruppo di infrastrutture che collegano questi oggetti e ne permettono la gestione, l'estrazione dei dati (data mining) e l'accesso alle informazioni che essi stessi generano. Fondamentalmente, un sistema IoT si avvale di sensori e attuatori in grado di svolgere funzioni specifiche e di comunicare con altre apparecchiature. Questi dispositivi fanno parte di un'architettura complessa che si occupa del trasporto, della conservazione, dell'elaborazione e della fruizione dei dati da parte degli utenti o di altri sistemi. In un contesto industriale, l'IoT viene visto come una collezione di elementi di calcolo ampiamente indipendenti che monitorano o controllano risorse fisiche, apparendo all'utente come un'unica operazione coerente finalizzata a una determinata logica di business.



La generica architettura di un sistema IoT risulta così composta:

`Ambiente locale`	Ambiente che contiene gli oggetti interconnessi e i *pickup points* locali.
`Trasporto`	Livello che permette agli oggetti e i pickup points di comunicare tra loro e con eventuali server.
`Storage`	Componente localizzato nel cloud, permette il processamento dei dati raccolti.
`API e GUI`	Componenti tramite i quali l'utente può accedere ai dati.

Due aspetti chiave di un sistema IoT è che i componenti devono essere interconnessi e autonomi. Molteplici componenti devono apparire all'utilizzatore come un sistema singolo e coerente, in accordo con la definiziona data da *Tanenbaum* di sistema distribuito.

Il collegamento tra l'Internet of Things (IoT) e la blockchain rappresenta una delle sinergie tecnologiche più promettenti dell'attuale era digitale, poiché permette di risolvere i limiti strutturali di sicurezza e centralizzazione insiti nei sistemi IoT tradizionali. Mentre l'IoT funge da interfaccia tra il mondo fisico e quello digitale attraverso sensori e attuatori che raccolgono dati in tempo reale, la blockchain fornisce il registro distribuito e immutabile necessario per memorizzare tali informazioni e gestirle in modo sicuro, trasparente e privo di autorità centrali. Questa integrazione trasforma l'IoT da una rete di dispositivi isolati o dipendenti da cloud proprietari in un ecosistema decentralizzato dove ogni oggetto può interagire, scambiare valore ed eseguire transazioni in modo autonomo e verificabile.

Uno dei pilastri fondamentali di questo legame è il superamento del modello centralizzato, che spesso costituisce un "single point of failure" vulnerabile ad attacchi informatici; al contrario, una struttura basata su blockchain distribuisce i dati e il potere di calcolo su molteplici nodi, garantendo che le informazioni provenienti dai sensori non possano essere manomesse o duplicate illegalmente. In questo contesto, gli smart contract (contratti intelligenti) fungono da motore operativo, permettendo ai dispositivi di coordinarsi tra loro sulla base di regole di ingaggio predefinite e basate sul consenso della rete. Ad esempio, un sensore IoT può attivare automaticamente un pagamento o una fornitura di servizi non appena vengono soddisfatte determinate condizioni fisiche registrate sulla catena di blocchi.

Un esempio concreto di questa integrazione è il progetto ADEPT (Autonomous Decentralized Peer-to-Peer Telemetry), un prototipo sviluppato da IBM e Samsung per implementare la blockchain nel dominio IoT. All'interno di ADEPT, sono stati realizzati casi studio rivoluzionari come quello di una lavatrice intelligente in grado di ordinare autonomamente il detersivo o i pezzi di ricambio quando necessario, o addirittura di negoziare direttamente il consumo di energia con altri elettrodomestici per ottimizzare i costi e il carico sulla rete elettrica. Questo sistema utilizza tecnologie come Telehash per il messaggistica peer-to-peer, BitTorrent per la condivisione di file e la blockchain di Ethereum per la coordinazione autonoma e l'autenticazione dei dispositivi.

L'implementazione tecnica di tali sistemi richiede però una gestione accurata delle risorse, poiché molti dispositivi IoT sono caratterizzati da bassa memoria e potenza di calcolo limitata. Per questo motivo, l'architettura si articola spesso in diverse tipologie di nodi: i Light Peers, come i semplici sensori, che mantengono solo le informazioni essenziali sul proprio stato; i Standard Peers, dotati di maggiore capacità per analizzare dati e mantenere parte della blockchain; e infine i Peer exchanges, che dispongono della potenza necessaria per gestire l'intera logica di business e l'esecuzione degli smart contract.

Oltre all'ambito domestico, il collegamento tra IoT e blockchain trova applicazioni vitali nella logistica e nella gestione delle catene di approvvigionamento, dove i sensori monitorano parametri critici come temperatura e posizione delle merci, registrandoli in modo indelebile sulla blockchain per garantire la tracciabilità e la certificazione dei prodotti. Anche nel settore industriale, questa combinazione abilita la manutenzione predittiva intelligente e la diagnostica avanzata, dove le macchine registrano il proprio stato operativo su un registro condiviso, permettendo interventi tempestivi e certificati senza la necessità di intermediari umani o server centrali.