Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Informatica

Sviluppo e Implementazione di un Marketplace per la Gestione di Asset Basato su Token ERC6956

Relatore:
Chiar.mo Prof.
Stefano Ferretti

Presentata da:
Matteo Boccali

Sessione II
Anno Accademico 2023-2024

"If things are not failing, you are not innovating enough."

Elon Musk

Introduzione

Nel 2008, anno in cui è avvenuta la pubblicazione del whitepaper di Bitcoin[1], il mondo ha visto affermarsi una nuova tecnologia: la blockchain.

La blockchain è un registro digitale distribuito reso sicuro tramite l'utilizzo della crittografia; gli utenti di tale sistema possono inviare transazioni, destinate ad altri utenti o, in alcuni casi, a programmi, denominati smart contract, firmando la transazione attraverso la loro chiave privata. Le transazioni inviate dagli utenti vengono ricevute da dei nodi, che hanno il compito di conservare l'intero storico delle transazioni registrate sulla blockchain.

Negli anni l'interesse per le blockchain ha subito un notevole incremento, come è possibile vedere dalla capitalizzazione del mercato delle criptovalute, come visibile nel grafico in figura 1.3.

Ma è solo nel 2014, anno in cui Kevin McCoy ha creato un'immagine digitale di un ottagono pulsante che cambia colore, intitolata Quantum, e l'ha caricata sulla blockchain di Namecoin, che iniziò a circolare il concetto di NFT, o Non-Fungible Token. Ovvero, un token in grado sia di memorizzare informazioni sia di definire la proprietá, o ownership, del token di riferimento.

Le sue potenzialità non furono chiare da subito, infatti all'epoca non sembrava un fatto così importante, ma questa immagine ha portato a ulteriori esperimenti sulla blockchain.

È però con l'avvento un nuovo tipo rivoluzionario di blockchain, Ethereum, che le limitazioni intrinseche della blockchain Bitcoin vengono meno, facilitando lo sviluppo, la programmazione, la custodia e il trading dei to-

[1]Vedi S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.

ken, la blockchain di Ethereum offre una soluzione affidabile ai problemi degli NFT.

Successivamente, nel 2017 il mondo è venuto a conoscenza degli NFT ma è nel 2021 che effettivamente questi hanno raggiunto la loro massima notorietà.

Con l'aumento del numero di progetti basati su NFT, iniziarono però a comparire anche i primi progetti scam, che mirano solamente a fregare gli utenti, recuperando chiavi private e svuotando i wallet. Ed è anche per questo motivo che l'adozione degli NFT sta procedendo a rilento da allora. Gli utenti hanno ancora dubbi relativi ad alcuni fattori: volatilità del mercato delle criptovalute ancora molto elevata e possibilità di frodi e truffe sempre dietro l'angolo.

È proprio quest'ultimo punto che il presente lavoro vuole andare ad approfondire.

Partendo con un riepilogo dei principali eventi della storia degli NFT, e le principali limitazioni che ne hanno rallentato l'adozione, passando per la revisione dello stato dell'arte, come ad esempio il protocollo ERC-721, si vuole poi andare a introdurre e spiegare il possibile utilizzo di un nuovo standard, ERC-6956, per rappresentare asset, fisici o digitali, nella blockchain Ethereum. Questo nuovo standard è basato su un nuovo meccanismo di verificare della proprietà: un oracolo crea una sorta di attestazione, la quale deve necessariamente essere usata per la verifica dei trasferimenti all'interno della blockchain.

Il contributo di questo progetto è quello di verificare la possibilità di utilizzare lo standard ERC-6956 per la gestione di NFT collegati ad asset reali tramite la creazione di una applicazione decentralizzata, un marketplace, che semplifica l'interazione con la blockchain per gli utenti. Cercando così di rendere più trasparente la gestione di NFT e dando la possibilità agli utenti di verificare l'autenticità degli NFT.

Quindi, per quanto l'obiettivo principale è la verifica di una possibile implementazione reale, è presente anche un secondo obiettivo, ovvero il cercare

di costruire una applicazione decentralizzata che gli utenti possano usare con facilità, cercando di aumentare la fiducia degli utenti rispetto agli NFT per far sì che sempre più utenti possano entrare a far parte di questo mondo.

Per finire, trattandosi di un progetto sperimentale verranno svolti dei test per quanto riguarda la parte relativa ai contratti, e quindi tutte le possibili operazioni concesse dai contratti creati, mentre per validare il tutto sono state svolte delle prove sperimentali non solo in una blockchain di test locale, bensì anche nella testnet Sepolia. Relativamente alle operazioni svolte su Sepolia testnet, sono state svolte un minimo di 10 prove per ogni operazione, scandite su date differenti per far sì di avere delle metriche il più generali possibili su cui valutare l'implentazione svolta.

Vedremo infine l'implementazione di tale progetto, e parleremo dei requisiti funzionali e non funzionali che sono stati raggiunti, utilizzando le prove sperimentali e i risultati numerici per validare le conclusioni.

L'organizzazione dell'elaborato è descritta di seguito:

- il Capitolo 1 introduce lo stato dell'arte degli NFT, fornendo una prima panoramica sulla storia delle blockchain, degli smart contract e degli NFT. Verranno quindi discussi alcuni standard e alcuni progetti già presenti nel mondo reale, per poi passare all'introduzione ad un nuovo standard, ERC-6956, che è quello su cui si basa il presente lavoro.

- il Capitolo 2 discute della fase di progettazione del sistema nella sua interezza. Partiremo definendo gli scenari e i casi d'uso, con cui poi capiremo come strutturare il sistema, quali sono i componenti che ne fanno parte e come questi interagiscano tra di loro per far funzionare il sistema.

- il Capitolo 3 concerne l'implementazione del sistema che è stata eseguita sulla base di quanto detto nel capitolo di progettazione. Verranno discusse le implementazioni componente per componente e per finire verrà descritta la fasi di validazione e test.

- il Capitolo 4 vuole mostrare i risultati raggiunti dall'implementazione del sistema. Verranno mostrati degli screenshot per dare un'idea del risultato della parte frontend dell'applicazione, e successivamente verranno mostrati i risultati numerici delle prove sperimentali che sono state eseguite.

- nelle conclusioni verranno discussi i risultati mostrati nel capitolo precedente, e verranno definiti alcuni possibili sviluppi futuri per proseguire con l'avanzamento dell'utilizzo di questo nuovo standard.

Indice

Introduzione **i**

1 Stato dell'arte **1**

1.1 Blockchain 1

1.2 Gli smart contract e il problema del Gas 3

1.3 NFT 5

1.3.1 Storia degli NFT 6

1.4 Andamento degli NFT 10

1.5 Physical NFT 12

1.6 Standerd ERC 13

1.6.1 ERC-20 13

1.6.2 ERC-721 14

1.6.3 ERC-6956 15

2 Progettazione **19**

2.1 Motivazione e casi d'uso 20

2.2 Personas 24

2.2.1 Marco Rossi 24

2.2.2 Alessio Bianchi 27

2.2.3 Martina Rossi 30

2.2.4 Elena Boschi 32

2.3 Requisiti 34

2.4 Architettura del sistema 35

2.4.1 Descrizione del sistema 35

2.4.2 Componenti del sistema 36
2.4.3 Interazioni del sistema 46

3 Implementazione **49**
3.1 Processo di sviluppo 49
3.1.1 Infrastruttura blockchain 50
3.1.2 Oracolo 56
3.1.3 IPFS 56
3.1.4 Subgraph 57
3.1.5 Backend 58
3.1.6 Frontend 59
3.1.7 Tecnologia dell'ancora - Arduino 61
3.2 Validazione e Test 63
3.2.1 Test degli smart contract 64
3.2.2 Test dei subgraph 66

4 Risultati **71**
4.1 Applicazione finale 72
4.1.1 Create NFT 73
4.1.2 My NFT 74
4.1.3 Marketplace 76
4.1.4 NFT Info 77
4.1.5 Redeem 78
4.2 Risultati quantitativi 79

Conclusioni **83**

Bibliografia **87**

Elenco delle figure

1.1 Grafico sugli NFT traders univoci attivi mensilmente. 2
1.2 Grafico sul volume di scambio degli NFT. 3
1.3 Capitalizzazione del mercato delle criptovalute. 10
1.4 Schema di funzionamento di ERC-6956. 15

2.1 Diagramma dei componenti del sistema 37
2.2 Legenda per le blueprint. 43
2.3 Blueprint dell'applicazione. 43
2.4 Blueprint dedicata per la pagina NFT Info. 44
2.5 Diagramma di sequenza del caso d'uso di acquisto. 47
2.6 Diagramma di sequenza del caso d'uso di creazione. 48

3.1 Diagramma di Arduino con display. 62
3.2 Risultato dei test della sicurezza su ERC6956. 65
3.3 Risultato dei test della sicurezza su ERC6956Full. 65
3.4 Risultato dei test della sicurezza su NFTMarketplace. 66
3.5 Risultato dei test relativi al contratto ERC6956. 66
3.6 Risultato dei test relativi al contratto ERC6956Full. 67
3.7 Risultato dei test relativi al contratto NFTMarketplace. 67
3.8 Risultato del controllo di solidity-coverage. 68
3.9 Risultato dei test relativi ai subgraph. 69
3.10 Risultato relativo alla copertura dei test sui subgraph. 70

4.1 Wrb3Modal Connect Wallet. 72

4.2 Pagina relativa alla creazione di un NFT. 73
4.3 Pagina relativa alla creazione di un NFT. 74
4.4 Pagina degli NFT di un utente. 75
4.6 Dialog per il listaggio di un NFT. 75
4.5 Scheda riassuntiva di un NFT. 76
4.7 Scheda riassuntiva di un NFT, con bottone Buy. 77
4.8 Pagina delle informazioni di un NFT. 78
4.9 Pagina delle informazioni di un NFT. 78
4.10 Dialog con lettore di codice QR. 79

Elenco delle tabelle

2.1 Scheda di Marco Rossi. 27
2.2 Scheda di Alessio Bianchi. 29
2.3 Scheda di Martina Rossi. 31
2.4 Scheda di Elena Boschi. 34
2.5 Tabella dei requisiti. 35

3.1 Gas-report relativo ai test in locale. 68

4.1 Report delle Transaction Fee per ogni operazione. 82
4.2 Report del prezzo del Gas per ogni operazione. 82

Capitolo 1

Stato dell'arte

Il presente capitolo ha l'obiettivo di fornire una panoramica delle ricerche e delle tecnologie attualmente disponibili nell'ambito delle blockchain e degli NFT. L'analisi dello stato dell'arte permette di identificare i principali progressi, le sfide ancora aperte e le opportunità per ulteriori sviluppi.

Nelle sezioni che seguono, partiremo con un breve riepilogo sulla storia delle blockchain, partendo da Bitcoin e arrivando a Ethereum, successivamente passeremo a fare un riepilogo sulla storia degli NFT, sul come sono nati e spiegando il perché hanno ricevuto sempre più attenzione. Infine arriveremo a parlare degli standard più comuni relativamente alla creazione di token e NFT sulla rete Ethereum, ed infine arriveremo a dettagliare il nuovo standard ERC-6956, che sta alla base del presente lavoro.

1.1 Blockchain

Come già anticipato nell'introduzione, le blockchain rientrano all'interno del quadro più ampio delle Distributed Ledger Technologies (DLT), ossia tecnologie in cui una rete di nodi mantiene un registro distribuito contenente un insieme di record. Nel caso delle blockchain il contenuto del registro è rappresentato da un insieme ordinato di transazioni. Una volta che una transazione viene ricevuta da un nodo, questa viene aggregata con altre transazioni all'in-

terno di un blocco; ogni blocco, oltre alle transazioni in esso incluse, contiene ulteriori dati, tra cui un riferimento al blocco precedente. Cambiare la storia delle transazioni passate richiederebbe quindi di riscrivere anche tutti i blocchi successivi a quello modificato e, grazie a questo meccanismo, viene garantita l'immutabilità della storia delle transazioni passate. A partire dal 2008, sempre più persone si sono cimentate nello sviluppare nuovi progetti basati sulle blockchain, e sopratutto dopo la nascita della blockchain Ethereum, sempre più progetti sono nati nel mondo degli NFT. Conseguenza di questo, è stata la bolla degli NFT avvenuta nel 2021, dove sia il numero delle transazioni sia i prezzi sono letteralmente schizzati verso l'alto. A conferma di questo troviamo il grafico in figura 1.1, che indica il numero di "NFT traders" univoci attivi mensilmente, e il grafico in figura 1.2, che mostra il volume delle transazioni eseguite per ogni chain.

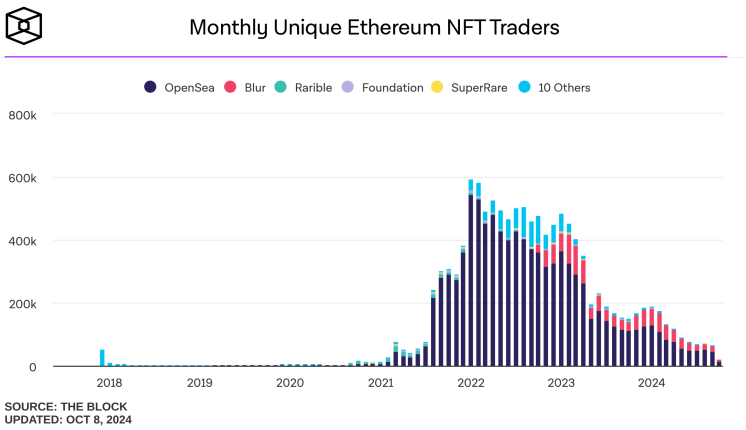

Figura 1.1: Grafico sugli NFT traders univoci attivi mensilmente.

Possiamo notare che per quanto riguarda la tecnologia degli NFT, una volta conclusa la bolla del 2021, siamo entrati in un trend ribassista.

Inizialmente le ragioni potevano essere collegate all'andamento del mercato delle criptovalute, ma queste dopo una correzione importante, si sono lentamente riprese e stanno tenendo comunque un trend positivo.

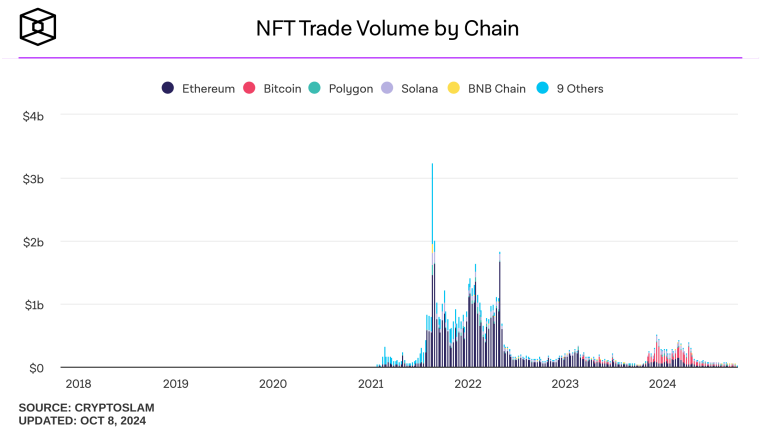

Figura 1.2: Grafico sul volume di scambio degli NFT.

Quindi, il motivo sembra dover riguardare la fiducia che sempre meno persone hanno nei confronti degli NFT. Molte persone infatti sono rimaste vittime di progetti NFT che si sono rivelati degli scam, perdendo grande parte dei soldi investiti. In più la sempre più stretta morsa dei regolatori riguardante il mondo delle criptovalute e degli NFT, ha fatto ricredere tutte quelle persone che pensavano potessero fare "soldi facili" con gli NFT.

Le ragioni alla base dell'incessante crescita di utilizzo degli NFT fino al 2021 infatti è stata principalmente dovuta dal fatto che gli utenti pensavano di poter lucrare con gli NFT, e non erano realmente interessati agli NFT per cosa sono davvero.

Ad oggi quindi, solo gli utenti che realmente credono nel potenziale degli NFT sono ancora interessati ad investire sui progetti basati su NFT.

Ma, nonostante il forte scetticismo che aleggia intorno agli NFT, questi continuano ad evolversi.

1.2 Gli smart contract e il problema del Gas

Fatto un riepilogo sul punto in cui ci troviamo andiamo a parlare ora di cosa sono gli smart contract.

Gli smart contract, sono dei programmi che vengono rilasciati ed eseguiti sulla blockchain, nelle quali l'esecuzione del codice avviene in modo distribuito da parte dei minatori che operano sulla blockchain. L'esecuzione delle funzionalità offerte da uno smart contract viene richiesta dagli utenti tramite una transazione, nel cui campo data vengono inseriti i dati da utilizzare durante l'esecuzione del contratto; qualora le condizioni richieste per l'esecuzione della funzione venissero soddisfatte, la transazione dell'utente produrrebbe una modifica dello stato della blockchain. La crescita del tasso di utilizzo delle blockchain, avvenuta negli ultimi anni, comporta un importante aspetto da tenere in considerazione, ossia l'aumento dei costi di transazione che gli utenti si trovano a dover sopportare. Nelle blockchain la sicurezza del sistema viene mantenuta mediante l'utilizzo della crittografia e un meccanismo di incentivi e disincentivi economici: vale a dire le Fee per gli utilizzatori del network e le ricompense per i miner. In Ethereum le Fee vengono quantificate mediante il Gas, per cui il costo di una singola transazione si basa su due aspetti: le unità di Gas da utilizzare e il prezzo che si è disposti a pagare per tale quantitativo. La formula per il calcolo delle Fee è quindi:

$$\text{TotalFee} = \text{GasUnits} \times \text{GasPrice} \tag{1.1}$$

Il quantitativo di Gas utilizzato dipende dalla lunghezza e complessità dell'operazione che il soggetto inviante la transazione vuole eseguire. Il Gas può quindi essere visto come "potenza computazionale" della EVM che gli utenti acquistano per eseguire le loro operazioni. Il prezzo del Gas viene espresso in GWei e dipende dalla congestione del network: in momenti di alto utilizzo della blockchain il costo di una transazione aumenta, in quanto tutti i partecipanti desiderano che le loro transazioni vengano eseguite, ma il Gas utilizzabile per ogni blocco è limitato; al contrario, in momenti in cui il network non viene utilizzato al massimo delle sue capacità, il costo del Gas sarà inferiore poiché la richiesta di utilizzo è bassa.

A prima vista potrebbe sembrare che il Gas comporti solo svantaggi per

gli utenti di Ethereum; tuttavia, come accennato in precedenza, la funzione delle Fee è quella di disincentivare comportamenti malevoli sulle blockchain. Infatti, un soggetto che volesse bloccare il network inviando continuamente transazioni per impedire che altri utenti possano a loro volta effettuare transazioni si troverebbe a dover spendere un enorme somma di denaro in Gas e dunque l'attacco, che costituisce in tutto e per tutto una forma di Denial of Service (DoS), risulterebbe economicamente non conveniente. La seconda funzione del Gas è quella di incentivare i miner a continuare ad aggiungere i blocchi contenenti le transazioni; infatti, oltre ai nuovi Ether che vengono creati per ogni blocco minato, i miner possono tenere parte delle Fee come ulteriore ricompensa. Per quanto le Fee consentano di impedire attacchi alle blockchain, risulta evidente come il loro pagamento possa scoraggiare lo sviluppo e il rilascio di smart contract sulle blockchain a causa dei costi per interagirvi; per questo motivo, negli ultimi anni, vi è stata una notevole proliferazione di chain che propongono differenti soluzioni a livello di protocollo ed architettura al fine di consentire costi di transazione più bassi rispetto a piattaforme come Ethereum e Bitcoin.

Questo, mostra come anche il concetto del Gas può essere tra i vari motivi per cui gli utenti sono sempre meno invogliati all'interazione con il mondo degli NFT, ma sebbene possano sembrare un costo eccessivo da pagare, permettono il corretto funzionamento di tutta l'infrastruttura.

1.3 NFT

Abbiamo parlato di Bitcoin, di Ethereum e degli smart contract, tutti concetti alla base degli NFT, ma ancora non abbiamo detto nulla riguardo a cosa è davvero un NFT.

Un NFT, acronimo di Non-Fungible Token, è un bene digitale unico. A differenza di asset fungibili come il petrolio, l'oro, le banconote o le criptovalute, che sono identici tra loro e completamente intercambiabili, gli NFT rappresentano proprietà di oggetti unici, sia nel mondo digitale che in quello

reale, grazie alla tecnologia blockchain. Un esempio di NFT potrebbe essere un'opera d'arte digitale, dove anche due copie identiche restano uniche perché ciascuna ha un'identità distinta e non può essere sostituita.

Gli NFT, inizialmente definiti come "monetized grapics", hanno una vasta gamma di usi. Possono rappresentare la proprietà di qualsiasi tipo di media, permettendo di condividere e possedere contenuti digitali in modo sicuro e verificabile tramite la blockchain. Con il tempo, la loro definizione è diventata più flessibile, poiché stanno evolvendo continuamente. Oltre al loro utilizzo attuale, come ad esempio nella raccolta fondi, in futuro gli NFT potrebbero essere usati per conservare documenti personali importanti, come certificati di nascita o cartelle cliniche, ampliando ulteriormente il loro potenziale.

In sintesi, gli NFT offrono una nuova forma di proprietà digitale, trasformando il modo in cui concepiamo e gestiamo beni sia fisici che virtuali, con prospettive interessanti per il futuro.

Introdotto il concetto di NFT, passiamo ora a vedere la storia di questi, ovvero come sono nati e il motivo per cui sono nati.

1.3.1 Storia degli NFT

Passiamo ora a discutere della storia degli NFT, facendo un breve riepilogo sui principali eventi.

2012: Il precursore degli NFT, Colored Coins

Sebbene il termine NFT non fosse stato ancora coniato, nel 2012 si assisteva a qualcosa di molto simile. Meni Rosenfield, l'inventore della prima borsa Bitcoin di Israele, pubblicò un documento innovativo sul concetto di Colored Coins, "Overview of Colored Coins" [1].

Queste monete, costituite da piccole denominazioni di un bitcoin e talvolta grandi quanto un singolo satoshi, potevano raccogliere informazioni, confermare la proprietà e rappresentare una vasta gamma di asset, tra cui proprietà, buoni, azioni di aziende, abbonamenti, token di accesso e collezionabili digitali.

Tuttavia, i Colored Coins presentavano due problemi principali.

Innanzitutto, il valore di questi asset era determinato dal consenso tra i partecipanti, il che significava che il loro valore poteva fluttuare notevolmente, rendendoli vulnerabili a fallimenti. In secondo luogo, la blockchain di Bitcoin aveva limitazioni significative; contrassegnare la proprietà e gestire applicazioni correlate richiedeva troppe transazioni, rendendo i Colored Coins meno praticabili.

Nonostante questi difetti, i Colored Coins rappresentavano un enorme passo avanti nelle capacità di Bitcoin. Funzionavano meglio in ambienti autorizzati, il che in alcune circostanze rendeva più sensato utilizzare un database tradizionale.

Sebbene i Colored Coins non abbiano risolto tutti i problemi legati alla rappresentazione degli asset sulla blockchain, portarono molte persone a rendersi conto dell'enorme potenziale di emissione di asset, tra cui beni del mondo reale, sulle blockchain. Questo ha segnato l'inizio di un'era di innovazione che ha portato alla creazione degli NFT che conosciamo e amiamo oggi.

Tuttavia, le persone compresero anche che Bitcoin stesso, nella sua attuale iterazione, non era progettato per abilitare queste funzionalità aggiuntive.

2014: Il "primo" NFT, Quantum

La questione su chi abbia creato il primo NFT ha suscitato molte controversie nel corso degli anni, ma secondo svariate ricerche, si tratta di Kevin McCoy, un artista digitale. Il 3 maggio 2014, McCoy creò un'immagine digitale di un ottagono pulsante e cangiante, intitolata Quantum [1], e la caricò sulla blockchain di Namecoin. All'epoca, non era un grande affare, ma questa immagine portò a ulteriori sperimentazioni con la blockchain.

[1] https://www.sothebys.com/en/buy/auction/2021/natively-digital-a-curated-nft-sale-2/quantum

Nel 2014, Robert Dermody, Adam Krellenstein ed Evan Wagner fondarono Counterparty [2], una piattaforma finanziaria peer-to-peer e un protocollo Internet distribuito e open-source costruito sopra la blockchain di Bitcoin. Counterparty consentiva la creazione di asset, disponeva di uno scambio decentralizzato e persino di un token crittografico con il ticker XCP. Aveva numerosi progetti e asset, inclusi un gioco di carte collezionabili e il trading di meme.

Si determinò che Bitcoin non era ideale per supportare gli NFT e così nacque un nuovo tipo rivoluzionario di blockchain: Ethereum.

2015: Spells of Genesis su Counterparty

Nel aprile 2015, Counterparty ha stretto una collaborazione con lo studio di sviluppo di giochi EverdreamSoft per creare Spells of Genesis (SoG), il primo gioco mobile basato su blockchain. I giocatori avevano la possibilità di "blockchainizzare" una carta convertendola in un NFT altamente ricercato e commerciabile.

Solo tre mesi dopo il lancio della blockchain di Ethereum, Etheria, il primo progetto NFT, è stato messo sul mercato, sebbene la reazione iniziale fosse piuttosto deludente. Quasi nessuno dei 457 esagoni digitali di Etheria è stato venduto per i successivi cinque anni.

Inutile dire che ci è voluto del tempo prima che il mondo dell'arte riconoscesse il valore degli NFT.

2016-2017: Cryptopunks, Cryptokitties e Rare Pepes

A differenza del 2012, ora abbiamo una manciata di blockchain oltre a Bitcoin che supportano l'uso degli NFT, inclusa Ethereum, che è diventata il fulcro di tutti gli asset NFT (inclusi i celebri Rare Pepes) grazie al loro supporto per la creazione e l'implementazione degli NFT.

E così, la gente ha iniziato a creare più NFT.

[2] https://www.counterparty.io/

Nel ottobre 2016, uno dei meme più amati di Internet, Pepe the Frog, è approdato su Counterparty, poichè un certo numero di meme "Rare Pepe" sono stati rilasciati come asset sulla piattaforma. E, se c'è una cosa che unisce le persone, sono i meme. Si è formata una grande fanbase e i Rare Pepes sono diventati il primo progetto NFT virale. Nei due anni successivi, artisti di tutto il mondo hanno partecipato al progetto. Hanno rilasciato 1.774 carte.

Nel giugno 2017, una piccola agenzia di software canadese, Larva Labs, ha lanciato "Cryptopunks", un progetto composto da 10.000 personaggi pixelati unici sulla blockchain di Ethereum, e li ha distribuiti gratuitamente. Sono stati rapidamente afferrati e rivenduti a prezzi elevati.

Nel settembre 2017, Dete Shirley, CEO di Dapper Labs, ha coniato il termine "token non fungibile", affermando di non essere completamente soddisfatto del termine, ma che "token" sembrava una scelta migliore rispetto all'utilizzo di "oggetto" o "cosa".

Infine, nel dicembre 2017, la società madre di Dapper Labs, Axiom Zen (un'altra agenzia di software canadese), ha lanciato "Cryptokitties", un gioco NFT in cui i giocatori potevano adottare, allevare e scambiare adorabili gatti virtuali. E se c'è una cosa che Internet ama più dei meme, sono i gatti! Cryptokitties ha fatto notizia in tutto il mondo e ha persino bloccato l'intera Ethereum a causa dell'enorme valore delle transazioni.

2018–2020: L'ascesa dei marketplaces di NFT

Gli NFT sono in crescita poiché sempre più piattaforme offrono servizi legati alla blockchain e sempre più blockchain cercano di creare le proprie versioni degli NFT. Grandi marketplace come Rarible e NiftyGateway sono emersi, dove le persone possono coniare, acquistare, vendere e scambiare i propri beni digitali. Il più noto di questi marketplace è diventato OpenSea. Ha visto gli utenti affluire sul sito web nella speranza di scoprire il prossimo progetto NFT virale. Nei successivi anni, OpenSea sarebbe diventato la casa di oltre 48 milioni di NFT e ha movimentato oltre 4 miliardi di dollari

in valuta digitale. Attualmente, la società ha una valutazione di oltre 1,5 miliardi di dollari.

Artisti di grande fama come Tyler Hobbs, Dmitri Cherniak e Monica Rizolli hanno iniziato a capitalizzare il craze degli NFT e a vendere arte virtuale. Nel giugno 2020, i creatori di Cryptokitties hanno lanciato NBA Top Shot. È essenzialmente una collezione digitale dei migliori momenti delle partite di NBA.

Inoltre, con le persone a casa durante la pandemia, i giochi basati su NFT hanno sperimentato un aumento di nuovi utenti. Uno dei giochi più popolari è Alien Worlds, lanciato nel dicembre 2020. Il gioco ha i giocatori che viaggiano tra mondi per estrarre Trillium (il token di gioco) e combattere tra loro mentre guadagnano NFT e criptovalute. Con oltre cinque milioni di utenti, è attualmente il gioco blockchain di maggior successo fino ad oggi.

Figura 1.3: Capitalizzazione del mercato delle criptovalute al 2024. Immagine tratta da CoinMarketCap.

1.4 Andamento degli NFT

Dopo aver fatto una revisione della storia delle blockchain e degli NFT, con tutte le motivazioni che hanno portato all'avanzamento in questi campi,

ora vorremmo capire cosa ci aspetta nel futuro, cioè quali passi in avanti verranno fatti ed in quale direzione.

Come è possibile vedere in figura 1.2, il mercato degli NFT è molto volatile e per di più sembrerebbe anche esserci una certa correlazione con l'andamento del mercato delle criptovalute. Questo deriva dal fatto che in un mercato con un andamento positivo sempre più persone tendono ad essere interessate ad entrare, e viceversa, quando il mercato segue un trend negativo le persone tendono a lasciare perdere.

Ma è solo questo il motivo per cui il numero degli utenti ha un trend negativo dal 2021 ad oggi?

La risposta è da ricercare anche sotto l'aspetto della fiducia delle persone rispetto alle blockchain e agli NFT. Infatti, nonostante queste nuove tecnologie stiano prendendo sempre più piede, ancora ci sono alcuni aspetti per cui le persone tendono a non fidarsi e quindi ad evitare l'utilizzo di queste tecnologie.

Tra le principali motivazioni troviamo soprattutto problemi di usabilità ed accessibilità. Infatti, nonostante sempre più progetti stanno nascendo nell'ambito degli NFT, crescono di pari passo anche i progetti scam, ovvero quei progetti che tentano di applicare delle frodi nei confronti di utenti benevoli.

Per di più, stanno nascendo nell'ultimo periodo progetti basati su NFT che mirano a collegare in qualche modo il mondo digitale degli NFT e quello dei beni reali, che siano essi asset fisici o digitali a loro volta.

Nonostante l'obiettivo sia molto ambizioso, anch'esso va inconto ad alcuni problemi per quanto riguarda possibili utenti malevoli, interessati semplicemente a portare avanti delle truffe solo per un resoconto economico. Ad esempio, un utente malevolo potrebbe creare un NFT digitale che corrisponde ad un bene fisico, quando in realtà il bene fisico non è sicuro che esista, o se esiste potrebbe essere in qualche modo contraffatto.

Proprio per questo motivo, sono nati alcuni progetti che mirano a poter verificare in qualche modo che l'asset reale esiste davvero e che il rispettivo NFT sia effettivamente il suo "gemello digitale".

1.5 Physical NFT

Avendo quindi parlato dei vari problemi che accomunano i vari progetti che stanno nascendo nell'ambito del mondo NFT che mira a collegarsi al mondo reale, andiamo ora a parlare dei cosidetti Physical NFT.

Per prima cosa, i Physical NFT sono token digitali legati agli asset del mondo reale. Definiti anche NFT phygital, questi asset combinano il digitale e il fisico e possono essere utilizzati per dimostrare la proprietà degli asset del mondo reale, come opere d'arte, articoli di moda, beni di proprietà, biglietti e altro ancora.

I Physical NFT sono composti da due parti. Una parte si riferisce all'asset digitale rilasciato su una blockchain attraverso l'uso di smart contract. L'altra parte è l'asset fisico, che spesso è collegato a un corrispondente identificatore unico, come un codice QR o un tag NFC (Near Field Communication). Ad esempio, gli acquirenti delle scarpe da ginnastica Nike appena uscite, le CryptoKicks, riceveranno anche un NFT collegato alla scarpa.

Anche se l'autenticazione è una delle più grandi applicazioni degli NFT fisici, può essere utile anche all'interno della supply chain, dove la tecnologia blockchain può aiutare a fornire tracciabilità e garanzie di certificazione.

Ora però ci chiediamo quali tipi di problemi questo nuovo tipo di NFT può andare a risolvere?

Contraffazione dei prodotti Un grande vantaggio degli NFT fisici è che possono dimostrare l'autenticità e la provenienza. In un mondo in cui molti beni fisici possono essere facilmente contraffatti, gli NFT fisici possono essere uno strumento prezioso sia per i consumatori che per i produttori.

Le aziende possono collegare i numeri di serie dei loro prodotti agli NFT o collegare l'articolo fisico a un NFT utilizzando la tecnologia NFC o un codice QR, per verificare che il prodotto fisico sia autentico e monitorare il suo storico. Dato che il prodotto è registrato digitalmente sulla blockchain, la manomissione o la falsificazione diventano quasi impossibili.

Trasparenza A volte l'esperienza di acquisto di oggetti da collezione di seconda mano può essere impegnativa per gli acquirenti. Questo è ancora più vero quando non hanno una chiara comprensione della storia dei prezzi e del valore di mercato di un articolo di seconda mano. Un NFT collegato a questi articoli può fornire agli acquirenti una panoramica completa della storia delle transazioni dell'articolo.

1.6 Standerd ERC

In una delle sezioni precedenti, abbiamo parlato dei primi NFT ed abbiamo vagamente fatto riferimeno agli standard ERC-20 e ERC-721. Ora con la presente sezione andremo a far luce sulle funzionalità di questi standard e sul perché si è passati da uno all'altro per quanto riguarda la creazione degli NFT. Per finire andremo ad introdurre un nuovo standard, ERC-6956, che crediamo poter essere un notevole passo in avanti per il futuro dei Physical NFT e della loro adozione nel mondo reale.

1.6.1 ERC-20

ERC-20 [2] è uno standard per i token fungibili sulla blockchain di Ethereum. I token fungibili sono identici tra loro e intercambiabili, il che significa che ogni unità ha lo stesso valore e può essere scambiata con un'altra senza differenza. Questo li rende ideali per rappresentare valute digitali, utility token o qualsiasi asset che può essere suddiviso e trattato in modo uniforme.

Le sue caratteristiche principali sono:

- Fungibilità: Ogni token ERC-20 è identico agli altri (come avviene per criptovalute tipo ETH o BTC). Un token ERC-20 vale esattamente come qualsiasi altro dello stesso tipo.
- Divisibilità: I token ERC-20 possono essere suddivisi in unità più piccole. Il numero di decimali è specificato dallo sviluppatore del contratto.

- Interoperabilità: Gli smart contract ERC-20 seguono un'implementazione standard, il che significa che tutti i token creati con questo standard sono compatibili con qualsiasi portafoglio, exchange o dapp che supporta ERC-20.

1.6.2 ERC-721

ERC-721 [3] è uno standard per i token non fungibili (NFT, Non-Fungible Tokens) sulla blockchain di Ethereum. Definisce un insieme di regole che gli smart contract devono seguire per creare token unici e indivisibili. A differenza dei token basati su ERC-20 (che sono fungibili e intercambiabili come le criptovalute), ogni token ERC-721 ha un'identità distinta, il che significa che due token ERC-721 non sono intercambiabili tra loro, anche se appartengono allo stesso contratto.

Le sue caratteristiche principali sono:

- Unicità: Ogni token ERC-721 è unico e può rappresentare qualsiasi cosa digitale o fisica, come arte digitale, collezionabili, proprietà immobiliari o diritti di proprietà intellettuale.
- Indivisibilità: Non può essere diviso in parti più piccole (es. non si può possedere "metà" di un NFT ERC-721).
- Ownership: La proprietà di ogni token può essere tracciata e trasferita tramite smart contract, consentendo vendite, aste o altre forme di scambio.

Questo nuovo standard rispetto ad ERC-20 introduce una serie di altri metodi che permettono di comportarsi come abbiamo scritto sopra, tra cui:

- `transferFrom(address from, address to, uint256 tokenId)`: metodo usato per trasferire la proprietà di un token.
- `ownerOf(uint256 tokenId)`: metodo usato per capire chi sia il vero proprietario del token.

1.6.3 ERC-6956

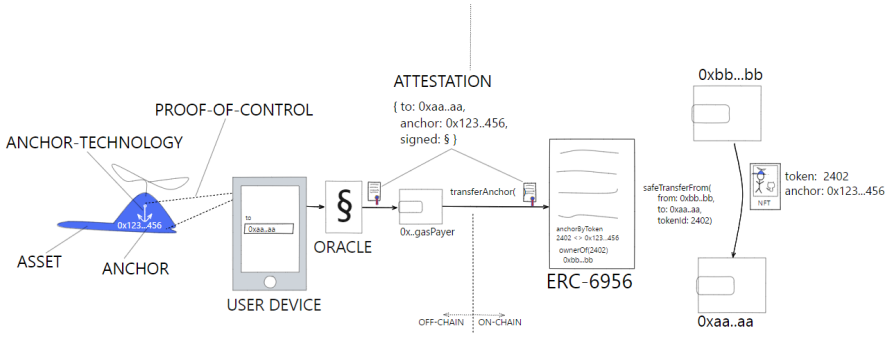

Figura 1.4: Schema di funzionamento di ERC-6956.

ERC-6956 [4], noto anche come Asset-Bound NFTs, è uno standard proposto per Ethereum che mira a collegare un token blockchain 1:1 con un bene fisico o digitale. Questo standard consente di tokenizzare vari oggetti, come collezionabili fisici (ad esempio opere d'arte o monete), parti di macchine, beni digitali (come oggetti di gioco) o asset astratti come abbonamenti o licenze.

Una caratteristica chiave dell'ERC-6956 è l'uso di un'autorizzazione basata su oracoli, che verifica il controllo del bene sottostante per garantire che i trasferimenti di token siano autentici. Questo rende lo standard adatto a casi d'uso in cui è necessario gestire in modo sicuro sia la proprietà digitale che quella fisica su blockchain. Un esempio del funzionamento è visibile in figura 1.4.

Prima di proseguire con l'introduzione dello standard, è bene specificare cosa intendiamo con alcuni termini:

- Ancora: identifica in modo univoco l'ASSET off-chain, sia che esso sia fisico o digitale.
- Tecnologia dell'ancora: deve garantire che un oracolo possa stabilire una Proof-of-Control sull'asset al di là di ogni ragionevole dubbio.

- Asset: si riferisce alla "cosa", sia essa fisica o digitale, che è rappresentata attraverso NFT secondo lo standard proposto.
- Attestazione: è la conferma che la Proof-of-Control è stata stabilita al momento della specifica dell'indirizzo to (destinatario).
- Proof-of-Control: significa possedere o comunque controllare un asset. Come viene stabilita la Proof-of-Control dipende dall'asset e può essere implementata utilizzando mezzi tecnici, legali o di altro tipo. Per gli asset fisici, il controllo è tipicamente verificato dimostrando la prossimità fisica tra un asset fisico e un dispositivo di input (ad esempio uno smartphone) usato per specificare l'indirizzo to.
- Oracolo: ha capacità di firma. Deve essere in grado di firmare attestazioni off-chain in modo tale che le firme possano essere verificate on-chain.

Un asset, ad esempio un oggetto fisico, è contrassegnato con un'ancora identificabile in modo univoco. L'ancora è legata in modo sicuro e inseparabile 1:1 a un NFT on-chain, durante l'intero ciclo di vita dell'asset.

Attraverso un'attestazione, un oracolo certifica che un determinato asset associato a un'ancora è sotto il controllo di una dato indirizzo, specificato come to (destinatario). L'oracolo firma l'attestazione off-chain. Le operazioni sono autorizzate verificando on-chain che l'attestazione sia stata firmata da un oracolo affidabile. È importante notare che l'autorizzazione viene fornita esclusivamente attraverso l'attestazione, o in altre parole, attraverso la Proof-of-Control sull'asset. Il controllore dell'asset è garantito come il controllore dell'NFT legato all'asset.

Le operazioni autorizzate tramite attestazione, come `transferAnchor(attestation)`, sono permissionless, cioè né l'attuale proprietario (indirizzo from) né il destinatario (indirizzo to) necessitano di utilizzare un wallet per interagire con il contratto.

Ma quale problema va effetivamente a risolvere questo standard? L'idea è quella di creare un collegamento tra un asset e un NFT. Negli anni sono

stati tentati approcci in cui ad esempio veniva salvato il collegamento tra asset e NFT tramite un identificatore presente come campo aggiuntivo nei metadati. Ovviamente questa soluzione non è sicura ed utilizzabile.

Ci sono anche svariati lavori in cui se sei il proprietario dell'NFT allora sei anche il proprietario dell'asset. Questa soluzione sembra poco logica e per di più potrebbe incappare in problemi qual'ora l'utente possessore dell'asset dovesse perdere il possesso sull'NFT (idealmente non sarebbe più il possessore dell'asset reale!).

Per questo motivo lo standard ERC-6956 propone:

- Di ancorare in modo inseparabile l'asset ad un NFT.
- Essere in controllo dell'asset reale dovrà corrispondere all'essere in controllo anche dell'NFT ancorato all'asset.

Ovviamente questo porta con sè il fatto che cambiando il possessore dell'asset dovrà cambiare necessariamente anche il possessore dell'NFT, e non importa se il precedente possessore dell'asset sia d'accordo o meno.

La proposta degli autori dello standard è quindi quella di complementare il già presente meccanismo di trasferimento di ERC-721, con un altro meccanismo, ovvero la prova tramite attestazione.

Tutto lo standard si basa sul fatto che l'oracolo debba essere fidato e non corretto, altrimenti il funzionamento diviene dell'intero sistema compromesso.

Tra i benefici secondari di ERC-6956, troviamo anche la possibilità di far pagare le Fee ad un terzo wallet e non all'utente che inizia la transazione di trasferimento.

Per riassumere quindi, la reale differenza con ERC-721 si trova nel metodo `transferAnchor` basato su attestazione. Ma ulteriori interfacce si possono estendere per prevedere altre funzionalità. Tra quelle citate dagli autori ci sono `IERC6956AttestationLimited` , `IERC6956ValidAnchors` e `IERC6956Floatable`.

Mentre `IERC6956Floatable` e `IERC6956AttestationLimited` , sono poco interessanti ai fini del presente lavoro, `IERC6956ValidAnchors` permette al mantainer del contratto (solitamente il deployer) di indicare una lista di ancore considerate valide, ovvero permette di limitare e tenere traccia delle ancore su cui il sistema può lavorare.

Ora, mostrato il funzionamento di questo standard, ai fini del presente lavoro, si vuole andare a proporre qualche modifica, che non intaccheranno le funzionalità mostrate fino ad ora, ma che metteranno qualche paletto.

Per prima cosa, il meccanismo di `transferAnchor` basato su attestazione sembra molto valido, ma il fatto che chiunque possa eseguire la transazione senza accordo del precedente proprietario sembra stonare con la direzione che gli NFT stanno prendendo.

Per questo motivo l'idea è quella di aggiungere una ulteriore funzione, che permetterà di trasferire l'asset solo previa acquisto dell'NFT, in poche parole, si vuole andare a forzare il cambio di proprietario solo quando effettivamente l'NFT viene dato via dal precedente proprietario.

Mentre il metodo `transferAnchor` potrebbe sempre essere utilizzabile, ma a questo punto mostrerebbe solo che l'NFT è stato trasferito, cambiando quindi il possessore dell'asset, ma non il suo proprietario. Questo cambierà solo chimando un ulteriore funzione che da qui in avanti chiameremo `redeemItem`.

Questo cambiamento permette quindi questa doppia funzionalità, cambio del possessore e cambio del proprietario con metodi separati.

Un ulteriore paletto che è stato pensato di mettere, è che l'utente che vuole iniziare la transazione, deve essere collegato con il suo wallet e dovrà pagare la Fee. Questo per via della natura del presente lavoro, dovendo interagire con una applicazione decentralizzata, è sembrato più logico che l'utente che inizia una trnsazione di trasferimento sia lui a pagare la Fee per il trasferimento.

Capitolo 2

Progettazione

In questo capitolo viene illustrata la progettazione del sistema che si vuole andare a sviluppare. La fase di progettazione è stata fondamentale per garantire che la soluzione proposta, fosse in grado di soddisfare le esigenze funzionali e non funzionali identificate durante l'analisi.

La progettazione del sistema è stata orientata da una serie di obiettivi fondamentali, i quali sono stati individuati mediante l'uso di personas, e da alcuni obiettivi secondari, tra cui la scalabilità, la facilità e l'esperienza utente. Tali obiettivi hanno guidato le scelte architetturali e tecnologiche, come descritto nelle sezioni successive.

Il processo di progettazione è stato suddiviso in più fasi, a partire dalle motivazioni e dai possibili casi d'uso completando con i requisiti identificati con le varie personas definite, è stato possibile definire l'architettura complessiva del sistema, successivamente si è passati alla progettazione dettagliata dei singoli componenti, finendo con il dettagliare le interazioni tra le varie parti del sistema, e quindi anche come i dati passano da un componente ad un altro.

Il capitolo è quindi organizzato in diverse sezioni: nella prima parte sarà condotta una analisi sulle motivazioni e sui possibili casi d'uso e verranno definite delle personas per definire i requisiti del sistema, nella seconda parte verrà descritta l'architettura generale del sistema, dettagliando per ogni

componente, il comportamento che vogliamo ottenere per arrivare al nostro obiettivo. Per finire saranno mostrate le interazioni che ci aspettiamo di avere fra i vari componenti.

2.1 Motivazione e casi d'uso

Nel capitolo precedente, nello specifico nella sezione 1.6.3, è stato introdotto il nuovo standard ERC-6956, ne sono state discussi gli obiettivi ed il perché è stato creato, ora però vogliamo andare a capire se questo effettivamente potrebbe essere di una qualche utilità pratica per risolvere un qualche problema nel mondo degli NFT, visto che ERC-6956 si pone come una sorta di passo in avanti rispetto ad ERC-721.

Ritornando a quanto visto nella sezione 1.5 riguardante i Physical NFT, abbiamo visto come questi si pongano come obiettivo quello di riuscire a collegare degli asset reali con il loro corrispettivo digitale, questo per poter avere benefici nel mondo digitale, quali ad esempio la prova di essere veramente in possesso dell'asset reale.

Ed è proprio qui che il presente lavoro vuole porre l'attenzione.

Quasi tutti i progetti che mirano al collegare asset reali con il corrispettivo NFT digitale, si basano sull'assunzione che chi è in possesso dell'NFT allora sia anche il possessore dell'asset reale, ma questa assunzione potrebbe essere pericolosa. Per fare un esempio pratico, potrebbe succedere che il proprio wallet Metamask venga hackerato (ad esempio qualcuno scopra la seed phrase) e che gli NFT presenti vengano trasferiti risultando alla perdita degli NFT. Questo caso è molto pericoloso, perché a questo punto il ladro diventerebbe il reale possessore dell'asset fisico, risultando il possessore dell'NFT. Ovviamente, per casi come questo, possono essere definiti dei limiti legali che possano appunto limitare comportamenti di questo tipo, per esempio dimostrando di essere i veri possessori dell'asset reale e che l'NFT sia stato davvero rubato, allora l'utente a cui è stato rubato l'NFT potrebbe tornare in possesso dell'NFT.

Viene da se, che tutta questa trafila risulti piuttosto scomoda, per via delle possibili complicazioni legali e burocratiche e per le possibili lunghe tempistiche. Inoltre sicuramente azioni di questo tipo fanno perdere la fiducia degli utenti rispetto all'utilizzo di una qualche piattaforma basata su NFT.

L'esempio citato sopra, è solo una delle motivazioni per cui i Physical NFT non hanno ancora ricevuto molta notorietà. Il problema che sta alla base deriva dal fatto che lo standard ERC-721 non è stato ideato e progettato inizialmente per collegare il mondo reale con quello fisico tramite l'utilizzo di una qualche forma di verifica del possesso dell'asset reale.

Alcuni progetti hanno provato a sopperire ai limiti di ERC-721, forzando la prova del possesso dell'NFT prima di poter vendere il proprio asset reale, questa prova del possesso consiste in qualche forma di prova per prossimità, che può essere verificata ad esempio con codici QR oppure NFC [1].

Ed è proprio qui che entra in gioco lo standard ERC-6956. L'obiettivo è quella di dare una strattura di base, tramite funzioni ad hoc, per far sì che gli utenti possano gestire gli NFT solo se sono in possesso di una attestazione che verifica il possesso dell'asset reale.

Ora che è stato definito il contesto in cui ERC-6956 vuole andare ad operare, cerchiamo di capire quali possano essere dei casi d'uso.

Essendo che gli utenti vogliono poter gestire il proprio NFT nel Web3.0, quindi potendo trasferire un NFT (sia esso stato comprato o venduto), servirà un sistema in grado di eseguire queste operazioni. Ma non solo, l'obiettivo del presente lavoro è anche quello di riuscire a risolvere un problema reale tramite l'implementazione di ERC-6956.

Quindi come scegliere, quale applicazione tentare di progettare? La risposta a tale domanda è da ricercare nel mercato degli NFT e sul perché questo stia vivendo un trend negativo a partire dalla bolla del 2021, come possiamo vedere dall'immagine in figura 1.2.

Il forte trend positivo è stato sicuramente favorito dal COVID-19 e da Beeple, ma evidentemente il mondo non era pronto per l'adozione di mas-

[1] https://rtfkt.com/faq/rtfkt-wm-chip

sa degli NFT. Infatti il forte trend positivo ha visto nel frattempo anche una grossa crescita di progetti scam, che hanno fatto vedere agli utenti anche l'altro lato della medaglia. Tra questi progetti, possiamo citare i primi marketplace NFT come OpenSea, Rarible, e così via.

Ora, bensì questi marketplace siano tutt'oggi molto utilizzati, hanno ricevuto una correzione nei volumi di scambio e nei prezzi che è ovviamente proporzionale all'andamento degli NFT nel mondo.

Per di più, nonostante abbiano continuato il proprio sviluppo, nessuno di questi si è mai posto l'obiettivo di abbracciare i Physical NFT, o comunque un concetto simile.

L'idea è quindi quella di creare un marketplace NFT basato su ERC-6956 per portare gli NFT al prossimo livello, quello in cui possano essere scambiati come controparte del loro corrispettivo asset reale (ovviamente prevedendo anche il successivo trasferimento dell'asset reale).

Un concetto simile è stato sviluppato da Boson Protocol [12]. Boson Protocol è un'infrastruttura decentralizzata progettata per collegare il mondo fisico con il mondo digitale attraverso la tecnologia blockchain. Il suo obiettivo principale è quello di consentire transazioni commerciali senza intermediari, permettendo a chiunque di scambiare beni fisici o servizi nel metaverso e in altri ambienti digitali utilizzando NFT. Tutto questo però, basandosi su ERC-721. Il presente lavoro invece, vorrebbe basarsi su ERC-6956.

Quindi, non solo quest'idea potrebbe aiutare gli utenti a fidarsi del mondo degli NFT, potrebbe anche risolvere uno dei problemi più comuni riguardanti i marketplace NFT e più in generale gli e-commerce. Il problema dei prodotti contraffatti.

La contraffazione ogni anno porta via molto denaro dal settore, inoltre per gli utenti che incappano in prodotti contraffatti porta anche ad un sentimento di incapacità e rabbia, che alla lunga possono portare ad abbandonare la piattaforma.

A conferma di quanto si è scritto, troviamo alcuni articoli tra cui "Blockchain Technology for Secure Supply Chain Management: A Comprehensive

Review" [5], dove viene mostrato come in effetti si prevede un grande sviluppo della tecnologia blockchain nella gestione della supply chain. Questo per via della capacità della blockchain di offrire trasparenza e tracciabilità, oltre che il passaggio da un sistema centralizzato ad uno decentralizzato.

Un altro articolo che dimostra come il problema della contraffazione sia reale, è "Block-Supply Chain: A New Anti-Counterfeiting Supply Chain Using NFC and Blockchain" [6]. In questo, viene mostrato come il passaggio da un sistema centralizzato ad uno decentralizzato porta con sè notevoli benefici in termini di performance e sicurezza.

Ulteriori conferme, le possiamo trovare nei lavori [7] e [8], che mirano ad utilizzare sistemi basati su blockchain per risolvere il problema della contraffazione nel campo della medicina.

Possiamo notare però, come questi lavori siano tutti stati sviluppati con tecnologie blockchain, non con NFT. Il presente lavoro vuole appunto dare una visione alternativa per risolvere un problema comune, come la contraffazione dei prodotti.

Per riassumere, l'idea del presente lavoro sarà quella di progettare un sistema, un marktplace, basato sullo standard ERC-6956 che possa funzionare come un normale marketplace o e-commerce, ma che con sè porti tutti i vantaggi che derivano dall'utilizzo dell'innovativo standard.

Possiamo vedere nell'articolo [9] come i marketplace possano completarsi perfettamente con lo standard ERC-6956, questo perchè la possibilità di avere una Proof-of-Control sugli asset nel marketplace, permetterebbe di risolvere i problemi citati precedentemente.

Il più comune caso d'uso che ne deriva è quindi l'acquisto da parte di un utente di un NFT, che in realtà corrisponde ad un asset reale, a seguito dell'acquisto avverrà la consegna dell'asset fisico, ed è solo durante la consegna che l'utente proverà, tramite una verifica per prossimità, di essere davvero in possesso dell'asset reale acquistato (una sorte di redeem). Alla fine di questa procedura, l'utente avrà finalmente il nuovo asset reale e si troverà nel wallet Web3.0 usato anche l'NFT corrispondente.

Infine, pensando al futuro e a come si possano sviluppare applicazioni come i marketplace e gli e-commerce, vediamo che di recente aziende molto grandi, come ad esempio Amazon [2], hanno iniziato a testare l'utilizzo di droni per fare consegne rapide. Il progetto, molto ambizioso, ha incontrato notevoli ostacoli burocratici ma continua a proseguire con ulteriori test.

Con la grande crescita che le stime attendono dal settore di consegne, come visibile anche dalla ricerca condotta negli articoli [14] e [15], attraverso l'utilizzo di droni, le applicazioni di questo meccanismo di consegna aumenteranno a vista d'occhio.

Si potrebbe vedere questa tipologia di consegna come un caso d'uso (futuro) per il sistema che si vuole andare a presentare, ponendo l'infrastruttura di base alle nuove tipologie di consegne del futuro.

2.2 Personas

In questa sezione andremo a definire in modo più specifico quali siano le funzionalità che il sistema dovrà possedere, basandosi su quanto detto e concluso dalla sezione precedente.

Verranno ora descritte alcune personas[3] per dettagliare più precisamente le necessità e le aspettative che i potenziali utenti avrebbero nei confronti del sistema nel mondo reale.

2.2.1 Marco Rossi

Attore	Marco Rossi, 45 anni, impiegato amministrativo

[2]Nello specifico, si rimanda al link su Amazon Prime Air: https://www.aboutamazon.it/notizie/innovazioni/prime-air

[3]Per ulteriori informazioni sulle personas e sui loro utilizzi si rimanda a: https://digital.gov/2023/05/19/personas-learn-how-to-discover-your-audience-understand-them-and-pivot-to-address-their-needs/

Scenario	Marco è un impiegato amministrativo da ormai 10 anni. È una persona piuttosto schematica ed abitudinaria, come dimostra il fatto che si porta sempre le stesse cose da mangiare in pausa pranzo. Sin da piccolo è sempre stato uno studioso di materie umanistiche e tende a fidarsi di qualcosa solo se ne comprende appieno il funzionamento. Marco ha sentito parlare di blockchain e NFT, ma non ha mai approfondito perché percepisce questi temi come complessi e, soprattutto, poco affidabili. È abituato a fare acquisti online per beni di valore ridotto e si affida principalmente a e-commerce tradizionali di cui si fida. Tuttavia, un amico gli ha mostrato un marketplace NFT che vende oggetti collezionabili con una controparte fisica (ad esempio un bene di lusso, un orologio di marca), e questo ha suscitato la sua curiosità. Vorrebbe acquistare un orologio di valore perché è da quando era piccolo che desidera portarne uno, ma allo stesso tempo ha paura di incappare in una truffa. Nel sito mostratogli dal suo amico ha trovato un orologio che gli piacerebbe comprare ed ha visionato il certificato di questo sotto forma di NFT, ma non riesce a capire il perché si dovrebbe, pensando che anche questo potrebbe essere falsificato. Prima di considerare l'acquisto, quindi, vuole verificare attentamente che gli NFT siano effettivamente legati a beni reali e che il marketplace offra informazioni affidabili sulle vendite e le proprietà degli asset fisici.

Analisi	Marco è una persona pragmatica e razionale, abituata a prendere decisioni basate su dati concreti e verificabili. Le sue esperienze passate lo hanno portato a essere diffidente verso le truffe online, e ha una visione critica su nuovi fenomeni come le criptovalute e gli NFT, spesso associati a speculazioni e frodi. Sebbene sia aperto a esplorare nuovi mercati, necessita di una solida base di fiducia e trasparenza prima di investire denaro. Motivazione: Verificare la solidità del marketplace NFT, avere conferma che gli oggetti acquistati siano collegati a beni reali e poter tracciare le vendite e la storia dei beni in modo trasparente.
Problemi	Mancanza di informazioni verificabili: Uno dei principali ostacoli è l'assenza di prove tangibili che un NFT sia legato a un bene fisico reale. Vuole poter tracciare l'intera storia dell'asset, dal momento della sua creazione fino alla sua attuale proprietà.

Obiettivi	Accedere a dati di vendita e proprietà recenti: Prima di acquistare, Marco vuole consultare lo storico delle vendite, i precedenti proprietari, e qualsiasi informazione rilevante legata all'asset fisico collegato. Questo gli darebbe una maggiore sicurezza sull'acquisto. Comprendere facilmente le transazioni: Ha bisogno che l'interfaccia del marketplace gli mostri i dettagli delle transazioni in modo chiaro e semplice, senza utilizzare gergo tecnico. Vuole poter tracciare l'intero ciclo di vita dell'NFT in modo intuitivo. Ridurre il rischio di frode: Marco cerca garanzie o sistemi di protezione che lo mettano al riparo da eventuali frodi. Vorrebbe che il marketplace integrasse metodi di sicurezza, come la verifica automatica dell'autenticità dell'NFT e del bene fisico associato.

Tabella 2.1: Scheda di Marco Rossi

2.2.2 Alessio Bianchi

Attore	Alessio Bianchi, 32 anni, Insegnante alle scuole superiori

Scenario	Alessio è una persona che lavora nella sua azienda di consulenza in ambito digitale. Si è laureato in matematica, ma ha sempre cercato di apprendere come funzionasse l'informatica e la programmazione. Prima del periodo Covid non stava molto tempo al computer, ma successivamente si è abituato alle nuove pratiche ed ha approfondito sempre più. Alessio ha scoperto gli NFT durante il picco del 2021 e da allora è diventato un sostenitore attivo della tecnologia. È sempre alla ricerca di nuovi progetti innovativi e opportunità nel mondo delle criptovalute. Utilizza marketplace NFT per acquistare e vendere opere digitali e collezionabili, e ha iniziato a collezionare NFT che rappresentano beni fisici, come opere d'arte e cimeli vari. Siccome è arrivato a possedere una collezione di grande valore, vorrebbe iniziare a vedere se esistono potenziali acquirenti, ma non ha idea di dove iniziare e quindi ha pensato che piazzare annunci online fosse la soluzione migliore. Desidera quindi trovare un modo per dimostrare la sua reale proprietà su questi asset fisici, e desidera anche poter dimostrare l'autenticità degli asset, visto che online potrebbero credere che i suoi prodotti siano contraffatti. Inoltre, vorrebbe poter prestare qualche pezzo della sua collezione a qualche suo amico temporaneamente, non perdendone la proprietà ma solo il possesso.

Analisi	Alessio è aperto ai cambiamenti e considera la tecnologia blockchain come un'opportunità per innovare e creare valore. È curioso e proattivo, sempre alla ricerca di modi per sfruttare al meglio i suoi investimenti in NFT. Vuole che in qualche modo la sua esperienza sia anche redditizia. È consapevole delle potenzialità degli NFT e desidera che il suo possesso venga riconosciuto in modo ufficiale, così da sentirsi sicuro delle sue transazioni e delle sue proprietà. Motivazione: Cerca soluzioni che gli permettano di dimostrare facilmente la sua proprietà reale di un asset fisico associato a un NFT. Vuole esplorare e utilizzare i suoi NFT in modi innovativi, tra cui la possibilità di prestare i suoi asset senza perdere la proprietà.
Problemi	Prestito degli NFT: È frustrato dalla mancanza di funzionalità nei marketplace che consentano di prestare NFT ai propri amici senza cedere la proprietà. Vorrebbe avere la possibilità di "prestare" l'accesso ai suoi asset, mantenendo però il diritto di proprietà.
Obiettivi	Dimostrare la proprietà degli asset: Alessio cerca una soluzione che gli permetta di provare in modo semplice e diretto la sua proprietà su beni fisici legati agli NFT, con un meccanismo diverso dai soliti certificati di autenticità, ad esempio con un chip che possa inviare segnali e provare l'autenticità "su richiesta". Piattaforma sicura e affidabile: Alessio cerca una piattaforma su cui mettere in vendita i suoi prodotti, in modo da poter fare compravendita.

Tabella 2.2: Scheda di Alessio Bianchi.

2.2.3 Martina Rossi

Attore	Martina Rossi, 24 anni, Studentessa di Marketing e Comunicazione
Scenario	Martina è un'appassionata di tecnologia e un'utente esperta di e-commerce. Ha scoperto il mondo degli NFT di recente e ne è rimasta colpita per le loro potenzialità, in particolare per la trasparenza e l'affidabilità che possono offrire rispetto ai tradizionali acquisti online. È abituata a utilizzare servizi di consegna innovativi, come quelli di Amazon, in cui i prodotti vengono consegnati in punti di ritiro sicuri. Di recente, è incappata su un paio di truffe su dei prodotti che ha comprato e da quel momento non si sente molto sicura di continuare ad acquistare online. Vorrebbe poter verificare in modo affidabile e trasparente l'autenticità dei prodotti che vuole acquistare. Inoltre spesso eccede nello spendere soldi online, perché non sa stimare nel modo appropriato il prezzo di un prodotto (si fa spesso guidare dalle emozioni). Crede che gli NFT potrebbero essere utili in questo, ma online non ha ancora trovato un'applicazione che faccia al caso suo.

Analisi	Martina è molto aperta a nuove esperienze d'acquisto e apprezza la comodità di poter ritirare i suoi ordini in luoghi sicuri e accessibili. È attratta dall'idea di poter acquistare NFT in modo semplice e diretto, con la certezza che ciò che compra sia autentico e di qualità. La sua mentalità tecnologica la spinge a esplorare soluzioni innovative per migliorare l'esperienza di acquisto. Motivazione: Vuole un'esperienza d'acquisto che combini la comodità e la sicurezza dell'e-commerce tradizionale con i vantaggi della tecnologia blockchain, come la trasparenza e l'autenticità.
Problemi	Autenticità dei prodotti: Ha avuto esperienze negative con acquisti online in cui ha ricevuto prodotti falsificati. Vuole una soluzione che le garantisca autenticità e trasparenza nei suoi acquisti. Integrazione con l'e-commerce tradizionale: Vorrebbe un'esperienza di acquisto che combini elementi dell'e-commerce tradizionale con l'innovazione degli NFT, ma attualmente non trova piattaforme che offrano questa integrazione. Storico: Vorrebbe poter verificare in modo facile e veloce lo storico di un prodotto, per capire se il prezzo di acquisto sia corretto o meno.
Obiettivi	Acquisti sicuri e trasparenti: Vuole una piattaforma che garantisca l'autenticità degli NFT e dei beni fisici associati, così da sentirsi sicura nei suoi acquisti. Esperienza d'acquisto familiare: Cerca una piattaforma che possa offrire un'esperienza simile a quella di Amazon.

Tabella 2.3: Scheda di Martina Rossi.

2.2.4 Elena Boschi

Attore	Elena Boschi, 61 anni, Pensionata
Scenario	Elena è una persona che ha avuto la fortuna di andare in pensione in anticipo per via di un grande patrimonio che le è stato passato dalla sua benestante famiglia. Si è da poco trasferita in montagna perché stanca del caos della vita urbana, e sta cercando di ambientarsi. Nonostante lei sia una pensionata, trova sempre qualcosa da fare e, abitando in montagna, spesso deve utilizzare degli asset (ad esempio e-bike), di un servizio poco flessibile, che deve prenotare per dei periodi fissati di tempo, ovviamente pagando per tutto il periodo e non in base a quanto lei abbia utilizzato l'attrezzatura. Trova questo servizio molto scomodo, ed essendo centralizzato non c'è possibilità di negoziazione. Vorrebbe avere la possibilità di utilizzare temporaneamente un asset solo quando ne ha davvero necessità, avendo così maggiore flessibilità, di fatto diventando il possessore in quel lasso di tempo.

Analisi	Elena è una pensionata con disponibilità economiche, che ha scelto di vivere in montagna per sfuggire al caos urbano e trovare uno stile di vita più tranquillo e naturale. Nonostante sia in pensione, è una persona attiva che cerca continuamente nuovi modi per impegnarsi e mantenere uno stile di vita dinamico. Uno degli aspetti della vita in montagna è la necessità di utilizzare vari strumenti e attrezzature per spostarsi e svolgere attività outdoor, come le e-bike. Tuttavia, il servizio di noleggio disponibile per questi asset non si adatta alle sue esigenze di flessibilità. Elena si trova obbligata a prenotare attrezzature per periodi di tempo prestabiliti, indipendentemente dal reale utilizzo che ne fa. Questo comporta costi superflui e uno spreco di risorse, oltre a un senso di rigidità che non si allinea con il suo desiderio di libertà e adattabilità. Essendo abituata a soluzioni più comode e personalizzate, derivanti dal suo benessere economico e dal contesto urbano, Elena vede con frustrazione la mancanza di negoziazione e flessibilità nel sistema attuale, che è centralizzato e non permette una gestione su misura delle risorse.
Problemi	Rigidità del servizio di noleggio: Il sistema di prenotazione è impostato su periodi di tempo fissi, senza possibilità di pagare solo per il tempo effettivo di utilizzo. Centralizzazione del servizio: Il sistema non offre alcuna libertà di interazione diretta tra utenti o una gestione decentralizzata delle risorse.

Obiettivi	Decentralizzazione del servizio: Implementare un sistema più aperto che permetta maggiore autonomia e negoziazione tra gli utenti, riducendo la dipendenza da un'entità centrale. Servizio personalizzato: Offrire un'esperienza più user-friendly, con possibilità di noleggiare un asset tramite un qualche dispositivo fisico che accerti il cambio di possessore. Così facendo nessun altro potrà utilizzare l'asset prima del rilascio da parte del precedente possessore.

Tabella 2.4: Scheda di Elena Boschi.

2.3 Requisiti

Dall'analisi condotta mediante l'utilizzo delle personas si può subito notare come un marketplace basato su NFT possa risolvere i problemi dei protagonisti degli scenari.

Tramite questa analisi, sono emersi dei requisiti che il sitema deve soddisfare per far sì che gli utenti possano eseguire le azioni descritte. Questi sono stati riassunti nella tabella 2.5.

I requisiti sono stati volutamente divisi in funzionali e non funzionali, questo perché non solo il sistema dovrà soddisfare quelli funzionali ma nella sua implementazione dovrà cercare anche di tenere conto di tutti quei requisiti che per quanto non aggiungano funzionalità al sistema, renderanno il prodotto finale più facile da utilizzare ed efficace.

Requisiti funzionali	Requisiti non funzionali
Dimostrazione non ripudiabile di proprietà, tramite Proof-of-Control. Poter verificare l'autenticità dell'NFT. Poter consultare lo storico dell'NFT. Sicurezza. Possibilità di prestare gli NFT. Completezza delle operazioni (tipo e-commerce). Trasparenza.	Usabilità. Facilità di comprensione. Funzionante su diversi dispositivi. Performante.

Tabella 2.5: Tabella dei requisiti.

2.4 Architettura del sistema

Nella presente sezione andremo a parlare del sistema, descrivendolo ad alto livello e parlando dei componenti che ne fanno parte e come questi interagiscono tra loro.

2.4.1 Descrizione del sistema

In questa sezione iniziamo con la progettazione vera e propria, cercando di dare una visione d'insieme del sistema basandoci sui requisiti elencati nella sezione precedente (2.3).

Il sistema progettato è un marketplace NFT decentralizzato, concepito come una dapp Web3.0. L'obiettivo del sistema è offrire agli utenti una piattaforma sicura e intuitiva per l'acquisto, la vendita e lo scambio di NFT, e per essere specifici Physical NFT, sfruttando la tecnologia blockchain. La dapp si basa sullo standard ERC-6956 e prevede l'implementazione di un smart con-

tract personalizzato, che include operazioni tipiche di un marketplace NFT, con alcune innovazioni chiave.

Uno degli aspetti distintivi del sistema è l'integrazione di una verifica sotto forma di attestazione firmata da un oracolo. Questo meccanismo aggiuntivo permette di verificare la vicinanza spaziale di una transazione prima che essa venga registrata sulla blockchain, migliorando la sicurezza e l'affidabilità degli scambi.

Per implementare questa sorta di "prova di prossimità" in modo sicuro, è stata adottata la soluzione di un'ancora fisica, rappresentata da un dispositivo Arduino. L'Arduino genera un codice QR che rappresenta l'ancora stessa, il quale può essere scansionato dall'utente tramite l'interfaccia frontend dell'applicazione. L'utente, una volta vicino all'asset fisico, utilizza la fotocamera del proprio dispositivo per leggere il codice QR, che contiene le informazioni necessarie per generare l'attestazione.

Un'altra caratteristica importante è la separazione tra il proprietario dell'NFT e il possessore dell'NFT. A tal proposito, sono state previste due operazioni distinte:

- Trasferimento di possesso: Funzione che trasferisce la proprietà momentanea dell'NFT tra utenti.

- Redeem: Funzione che consente di riscattare e trasferire la proprietà effettiva dell'NFT.

2.4.2 Componenti del sistema

Per comprendere il funzionamento del sistema, è fondamentale analizzare i singoli componenti che lo costituiscono. Ogni parte del sistema gioca un ruolo fondamentale, e ogni componente del sistema è stata studiata in modo tale da facilitare la realizzazione degli obiettivi che ci siamo prefissati inizialmente, e per aderire ai requisiti raccolti nella prima fase della progettazione.

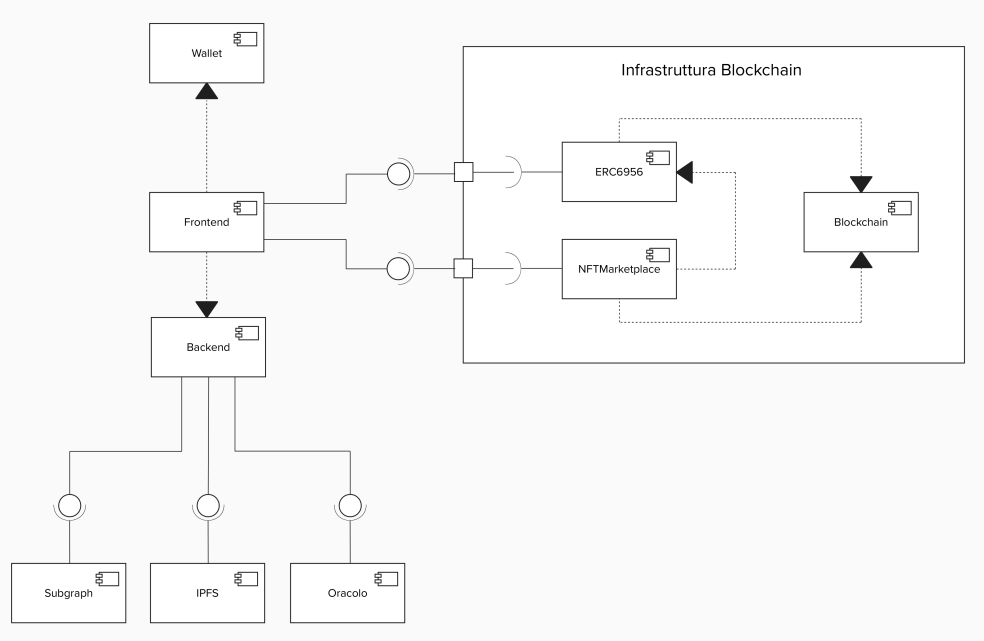

Figura 2.1: Diagramma dei componenti del sistema.

Infrastruttura blockchain

Partiamo con il descrivere in modo più approfondito la parte cruciale del sistema, ovvero l'infrastruttura blockchain.

Per questa parte è stato sviluppato lo smart contract relativo allo standard ERC-6956, arricchito con l'estensione di alcune interfacce che permettono il corretto funzionamento dello smart contract. L'interfaccia più importante che è stata implementata è sicuramente IERC6956ValidAnchors, questa permette di verificare ad esempio che l'ancora presente all'interno di una data attestazione sia effettivamente un'ancora valida. Per valida intendiamo un'ancora che sia presente nella lista di tutte le ancore che il sistma accetta, in parole povere è una lista di tutte le ancore che il sistema indica essere collegate ad un asset reale o che comunque si possono collegare ad un asset reale.

Mentre invece l'interfaccia `IERC6956AttestationLimited` non è stata implementata. Il motivo è da ricercare nella natura del marketplace, ovvero

dare la possibilità agli utenti di fare quanti scambi vogliano con qualsiasi NFT, cosa che andrebbe in contrasto con le funzionalità di questa interfaccia, che invece vorrebbe andare a porre un limite al numero di volte che un NFT può essere trasferito avendo una attestazione valida.

Ma lo smart contract ERC-6956 da solo non è sufficiente per definire tutte quelle operazioni che un utente può fare in un marketplace. Per questo motivo è stato implementato anche uno smart contract per queste operazioni, lo chiameremo d'ora in avanti NFTMarkeplace.

Le operazioni permesse da quest'ultimo contratto sono:

- Listaggio di un NFT, solo dopo approvazione
- Cancellazione di un listaggio di un NFT
- Acquisto di un NFT
- Vendita di un NFT
- Redeem di un NFT

L'operazione innovativa di questo contratto è l'operazione di redeem, questa infatti ha come scopo quello di verificare che l'utente a cui stiamo trasferendo un NFT acquistato sia effettivamente in possesso dell'asset reale corrispondente all'NFT.

Può sembrare un passaggio quasi superfluo, ma è proprio qui che riusciamo a tenere traccia di tutti i cambi di proprietà degli NFT. Mentre la funzione `transferAnchor` del contratto ERC-6956 permette un qualsiasi trasferimento di un NFT data un attestazione valida, la funzione `redeemItem` permette il trasferimento solo se il proprietario ha effetivamente fatto precedentemente l'acquisto dell'NFT e solo se ha effettivamente il controllo sull'asset collegato all'NFT, potendolo provare. Quindi, mentre la funzione `redeemItem` terrà traccia dei cambi di proprietà, la funzione `transferAnchor` terrà traccia dei cambi di possesso di un NFT.

Per finire, si vuole andare a sottolinare il fatto che questo tipo di gestione ha come risultato il fatto che l'utente dovrà avere un qualche portafoglio

Web3.0, tipo Metamask, collegato nella parte frontend. Mentre nella parte di spiegazione dello standard è stato scritto che basterebbe fornire l'address di un wallet, potendo far pagare le fee ad una terza parte. Questo è stato ritenuto necessario perchè gli utenti senza una funzione `redeemItem` ogni utente potrebbe trasferire l'NFT a suo piacimento avendo una attestazione valida. Questo porterebbe a problemi riguardo possibili utenti malevoli che potrebbe cercare di rubare l'asset reale e/o l'NFT collegato. Per questo si è reso necessario forzare una ulteriore funzione `redeemItem`, forzando però l'utente a collegarsi con un wallet al marketplace, e facendogli pagare le fee derivanti dalla transazione che si vuole eseguire.

Per quanto concerne l'implementazione dei contratti rimandiamo alla sezione di implementazione.

Oracolo

Abbiamo parlato di come utilizzare l'attestazione all'interno dei contratti per fare dei trasferimenti di NFT, ma ora vogliamo andare a vedere come effettivamente questa attestazione viene creata.

Per via della complessa natura degli oracoli, è stato scelto di simulare questa parte, e quindi di non creare un vero e proprio oracolo che abbia capacità di firmare un messaggio inviatogli.

La scelta è quindi ricaduta sul creare una API route dedicata interamente alla creazione della attestazione. Il punto cruciale è che il messaggio di ingresso dovrà contenere l'ancora che si vuole attestare, il compito dell'API sarà quello di creare l'intera attestazione e firmarla con un wallet specifico, utilizzato appositamente per firmare queste attestazioni e di cui lo smart contract ERC-6956 sia a conoscenza.

IPFS

Parte integrante del sistema è anche IPFS.

IPFS [4], o InterPlanetary File System, è un protocollo di rete peer-to-

[4]Per ulteriori informazioni su IPFS si rimanda a: https://docs.ipfs.tech/

peer progettato per creare un sistema di archiviazione e condivisione di file distribuito e decentralizzato.

Ecco alcune caratteristiche principali di IPFS:

- Decentralizzazione: A differenza dei tradizionali sistemi di archiviazione centralizzati, IPFS non si basa su un singolo server. I file sono distribuiti attraverso una rete di nodi, il che rende il sistema più resiliente e meno soggetto a fallimenti.

- Indirizzamento basato su contenuto: In IPFS, i file sono identificati in base al loro contenuto anziché alla loro posizione. Ogni file ha un hash (un identificatore unico) generato a partire dal suo contenuto. Questo significa che se due file hanno lo stesso contenuto, avranno lo stesso hash.

- Efficienza nella distribuzione dei file: IPFS permette di recuperare i file in modo più veloce ed efficiente, poiché i file possono essere scaricati da più nodi contemporaneamente, aumentando la velocità di download e riducendo il carico sui server.

- Persistenza dei dati: I file su IPFS possono rimanere disponibili finché ci sono nodi che li ospitano. Per garantire la persistenza, gli utenti possono "pin" i file, mantenendoli sempre disponibili sulla rete.

- Utilizzo in applicazioni decentralizzate (dapp): IPFS è spesso utilizzato in progetti di blockchain e applicazioni decentralizzate per archiviare dati in modo che siano facilmente accessibili e resistenti alla censura.

In sintesi, IPFS rappresenta una nuova frontiera nell'archiviazione e nella condivisione dei dati, offrendo un'alternativa più robusta e decentralizzata rispetto ai tradizionali sistemi basati su server centralizzati.

L'utilizzo di IPFS è stato previsto per via della necessità di memorizzare i dati degli NFT, e di conseguenza degli asset reali. I vantaggi sopra indicati e la possiblità di utilizzare un sistema decentralizzato anche per questa parte hanno fatto propendere per l'utilizzo di questa tecnologia.

Subgraph

Parte fondamentale del sistema è anche la gestione dei dati degli NFT e quindi degli asset reali collegati.

L'idea è quella di cercare anche in questo caso un sistema decentralizzato per essere consistenti con la natura decentralizzata del sistema, cercando allo stesso tempo di avere benefici in termini di efficienza.

Per questo motivo i subgraph di The Graph sono stati ritenuti la perfetta scelta per il sistema.

The Graph è un protocollo decentralizzato per l'indicizzazione e la query dei dati delle blockchain. Consente agli sviluppatori di costruire applicazioni decentralizzate (dapp) facilmente interrogando dati on-chain in modo efficiente.

Ecco alcuni aspetti chiave di The Graph:

- Indicizzazione dei dati: The Graph indicizza i dati provenienti da diverse blockchain e li rende facilmente accessibili tramite API GraphQL.
- Subgraph: Gli sviluppatori possono creare "subgraph" che definiscono quali dati dalla blockchain devono essere indicizzati e come devono essere strutturati. I subgraph possono essere utilizzati per estrarre informazioni specifiche da smart contract e altri dati on-chain.
- Decentralizzazione: Essendo costruito su tecnologie decentralizzate, The Graph mira a garantire che i dati siano accessibili in modo aperto e trasparente, senza necessità di affidarsi a servizi centralizzati.

In sintesi, The Graph facilita l'accesso ai dati blockchain, rendendo più semplice per gli sviluppatori costruire e scalare dapp.

Backend

Descritta la parte relativa agli smart contract, passiamo ora a parlare della parte backend.

Il backend dovrà ovviamente rendere fruibile la parte frontend. Ma la parte fondamentale del backend come visibile in figura 2.1 è quella di interfacciarsi con dei servizi esterni quali: IPFS, Subgraph e il cosiddetto oracolo.

Il backend dovrà quindi far sì che quando l'utente vuole fare determinate operazioni, il frontend possa chiamare le API messe a disposizione dal backend per ricevere alcune informazioni necessarie al funzionamento della dapp.

Le operazioni che si rendono necessarie in questo caso sono:

- Invio di informazioni a IPFS
- Invio di query ai subgraph
- Invio di richiesta di generazione e firma di una attestazione valida da parte dell'oracolo

Per finire, aspetto non meno importante sarà quello di rendere la parte front-end performante, senza avere dei ritardi inutili che compromettono l'esperienza degli utenti e la fiducia di questi nei confronti del sistema.

Frontend

Per quanto riguarda la parte frontend, la scelta è ricaduta sul tenere l'applicazione decentralizzata il più snella possibile, senza ovviamente mancare in nessuna delle funzionalità necessarie per il corretto funzionamento del sistema.

La scelta è quindi ricaduta su una single-page application con un menu che renda facilmente navigabile l'applicazione in tutte le sue parti. Questo permette di avere una applicazione che possa essere facile da usare e intuitiva nel suo utilizzo, avendo una mappa mentale delle operazioni ben chiara.

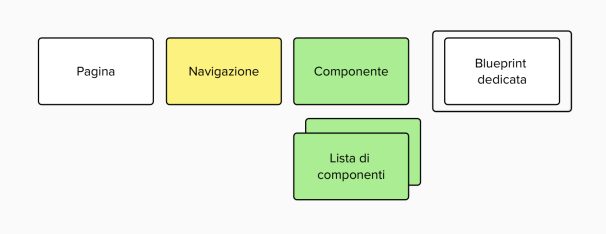

Figura 2.2: Legenda per le blueprint.

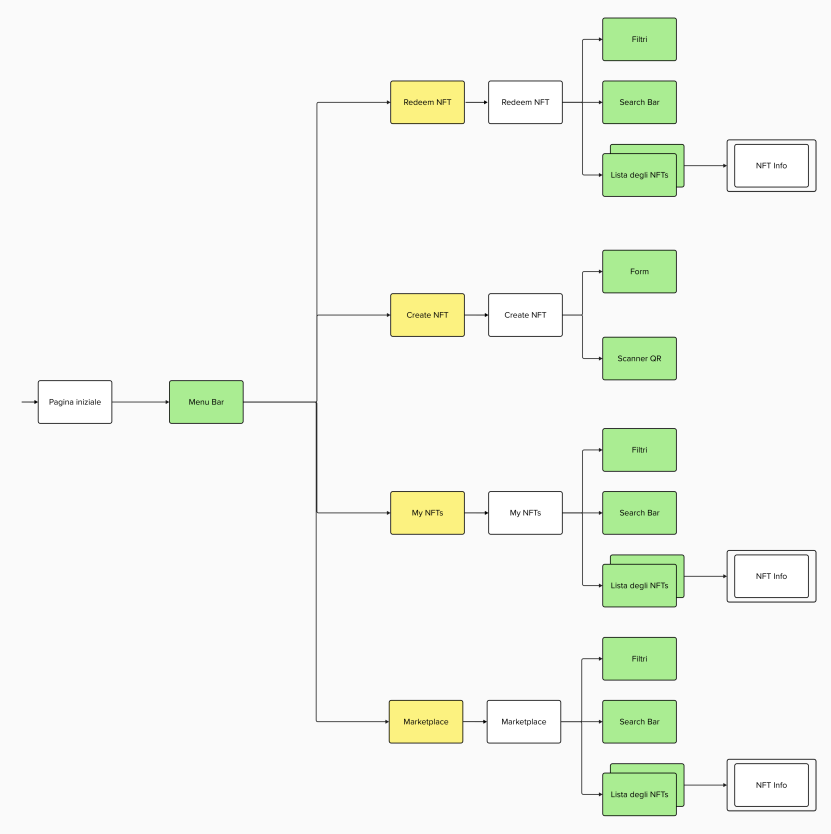

Figura 2.3: Blueprint dell'applicazione.

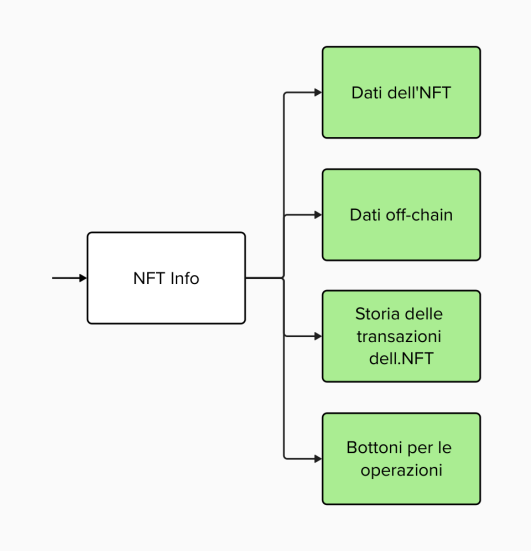

Figura 2.4: Blueprint dedicata per la pagina sulle informazioni di uno specifico NFT.

Per mostrare una prima progettazione grafica della parte frontend sono state inserite la figura 2.3 che mostra la mappa concettuale della applicazione nella sua interezza, e la figura 2.4 che mostra la pagina dedicata alle informazioni di uno specifico NFT. Per comprendere meglio queste figure, chiamate anche blueprint, è stata inserita la legenda in figura 2.2.

Tecnologia dell'ancora

Per finire, l'ultimo componente di cui andiamo a parlare è la tecnologia dell'ancora. Questo componente è parte integrante del sistema data la necessità di avere una parte fisica che riesca a mostrare l'ancora legata ad un certo asset reale.

Nel presente lavoro non ci poniamo come obiettivo quello di creare un prototipo di una nuova tecnologia che possa essere usata come ancora in questo sistema, tuttavia sarà necessario implementare un sistema di computazione fisico (come ad esempio i sistemi di edge computing) che possa agire come

tale, visto che il sistema si basa su una attestazione che deve essere firmata da un oracolo previa prova di essere in possesso dell'asset.

Per di più, come mostrato in una delle personas, si vorrebbe cercare di automatizzare il meccanismo di lettura dell'ancora.

Al giorno d'oggi non sono stati molti i tentativi di creare un modo univoco e non riproducibile di identificare un determinato asset reale tramite un qualche sistema in modo tale da poter sapere nel mondo digitale con quale asset si stia avendo a che fare.

In uno degli esempi [5] creati per mostrare le funzionalità di ERC-6956 viene mostrato come l'ancora sia un codice QR che una volta letto permette di gestire un determinato asset digitale.

La scelta di questo progetto è ricaduta sull'utilizzo di un codice QR da utilizzare come ancora.

Le motivazioni sono due. La prima è dovuta alla sua facilità di utilizzo e al suo sempre crescente inserimento nella vita quotidiana. Infatti sempre più applicazioni e progetti stanno utilizzando questa tecnologia, ed è quindi diventata mainstream, anche per persone che non hanno una grande esperienza in ambito tecnologico.

La seconda è dovuta alla natura di ERC-6956, e al cambiamento di cui si è discusso nella sezione 1.6.3. Infatti, nonostante il codice QR sia replicabile conoscendo la stringa da cui è stato creato (nel nostro caso conoscendo l'ancora), questo non è un particolare fondamentale, in quanto anche sapendo l'ancora di un asset reale, non potranno mai esistere due NFT aventi la stessa ancora; mentre anche falsificando il codice QR non potrà mai cambiare il proprietario di un NFT a meno che quest'ultimo non lo venda ad un altro utente, ed in questo caso solo colui che lo ha acquistato potrà diventare il nuovo proprietario. Ovviamente un utente malevolo potrebbe diventare il momentaneo possessore dell'NFT tramite `anchorTransfer`, ma questo è dovuto alla natura dello standard ed è anche discusso nella pagina web dedi-

[5] Esempio di integrazione di ERC6956: https://www.youtube.com/watch?v=jwTNJywnCcY&ab_channel=AuthenticVision

cata, ovvero dovranno essere previsti dei provvedimenti anche a livello della legge.

2.4.3 Interazioni del sistema

Introdotti tutti i componenti che fanno parte del sistema, passiamo ora a mostrare come questi interagiscano tra loro per garantire una esperienza fluida.

Per completezza sono state inserite delle figure che mostrano i diagrammi di sequenza di un paio di operazioni che gli utenti possono fare. La scelta delle operazioni non è casuale, sono state scelte l'operazione di acquisto di un NFT e quella di creazione di un NFT. Questo perché queste due operazioni riescono a mostrare tutte le parti del sistema e come lavorano, infatti operazioni come listaggio di un NFT e cancellazione di un listaggio di un NFT sono piú "semplici" e possono essere riassunte vedendo l'operazione di acquisto di un NFT.

Partiamo mostrando il flusso di interazione tra le diverse componenti utilizzando come caso d'uso il seguente: un utente controllando nel marketplace del sistema sceglie un particolare NFT che vuole acquistare, quindi cliccando sul pulsante "Buy" conferma la transazione e pagando partirà così la spedizione dell'asset reale collegato all'NFT acquistato. Una volta ricevuto l'asset riuscirà ad inquadrare il codice QR e potrà così far partire l'operazione di redeem nell'apposita sezione dell'app, ed una volta confermata la transazione potrà quindi ufficialmente gestire l'NFT collegato all'asset acquistato, diventandone di fatti il nuovo proprietario. La figura 2.5 mostra il diagramma di sequenza relativo a questo caso d'uso.

Ora invece andiamo a mostrare il flusso di interazione tra le diverse componenti utilizzando come caso d'uso la creazione di un NFT: un utente vuole creare un NFT collegandolo ad un asset reale di sua proprietà che m̀unita di una ancora valida (lasciamo questa parte non dettagliata in quanto non parte del presente lavoro). A questo punto cliccando nell'apposita pagina nell'applicazione l'utente potrà compilare un form con tutti i vari dati relativi

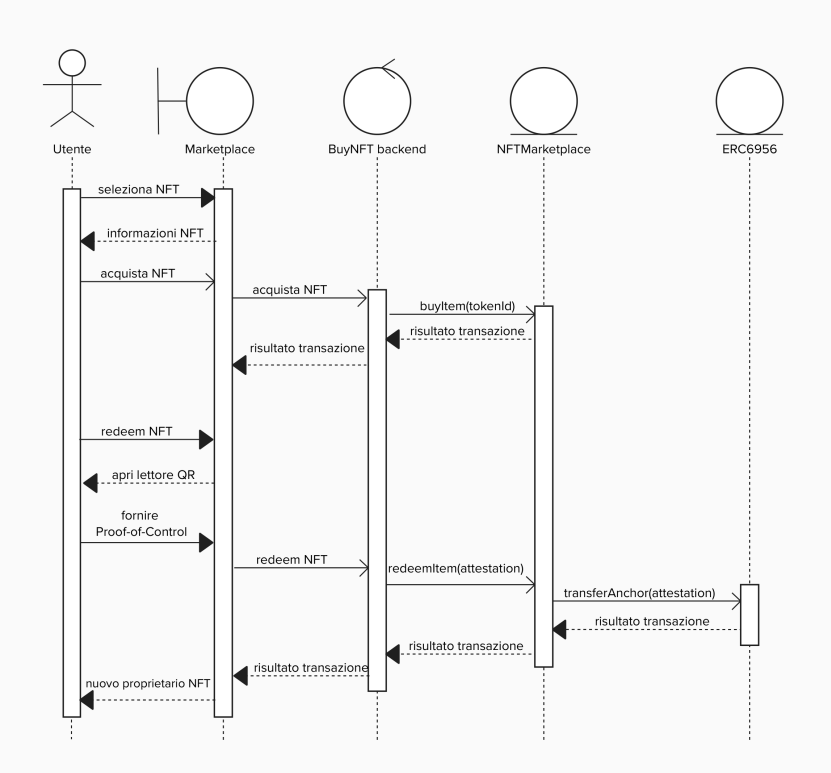

Figura 2.5: Diagramma di sequenza del caso d'uso di acquisto.

all'asset reale, e tra questi dovrà anche leggere il codice QR relativo all'ancora dell'asset. A questo punto l'utente firmerà la transazione per la creazione del nuovo NFT e alla fine lo riceverà nel proprio wallet e potrà visualizzarlo e gestirlo nella sezione My NFT. La figura 2.6 mostra il diagramma di sequenza relativo a questo caso d'uso.

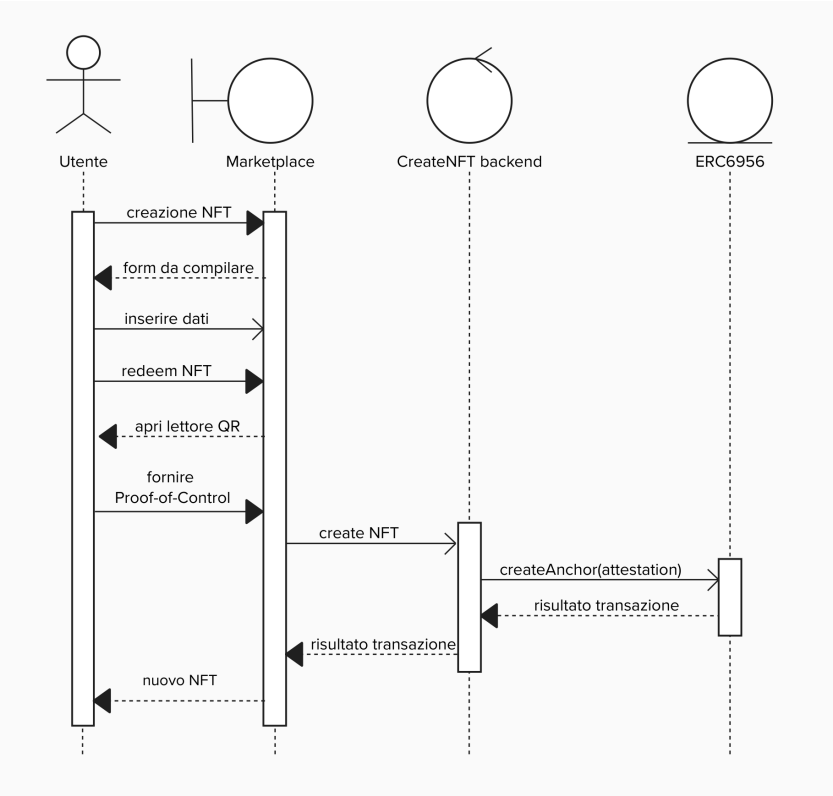

Figura 2.6: Diagramma di sequenza del caso d'uso di creazione.

Capitolo 3

Implementazione

Questo capitolo descrive il processo di implementazione della soluzione proposta, basata sulle specifiche e sui requisiti individuati nel capitolo precedente 2.3. Verranno illustrati i principali passaggi seguiti per la realizzazione del sistema, le tecnologie utilizzate e le scelte progettuali adottate. Infine verrano mostrate anche le tecniche di validazione e verifica utilizzate al fine di testare il funzionamento del sistema.

Il capitolo è suddiviso in tre sezioni principali: la prima descrive tutte le varie fasi del processo di sviluppo, cha a sua volta sarà suddivisa in più sottosezioni uno per ogni componente del progetto, la seconda esamina i problemi e soluzioni adottate per mitigarli, la terza si focalizza sui test effettuati per testare e validare il progetto.

3.1 Processo di sviluppo

Per presentare al meglio la fase di sviluppo è stato scelto di strutturare la presente sezione in più sottosezioni, una per ogni componente che abbiamo visto far parte del sistema in fase di progettazione.

3.1.1 Infrastruttura blockchain

La prima parte implementata è stata ovviamente quella relativa allo sviluppo degli smart contract, e per questo scopo è stato scelto di implementare questi contratti utilizzando il linguaggio di programmazione più utilizzato per lavorare con le blockchain, ovvero *Solidity*. Per un approfondimento su *Solidity* si rimanda al link a piè di pagina [1].

Una volta definito il linguaggio di programmazione, passiamo a vedere le parti implementative degli smart contract, specificando che è stato utilizzato *OpenZeppelin* come package di sviluppo.

ERC6956

Il primo contratto implementato è stato ovviamamente quello basato sull'interfaccia di base di ERC-6956. Vedremo a seguire che questo contratto di base sarà esteso con alcune interfacce per alcune funzionalità aggiuntive come spiegato nella sezione di progettazione 1.6.3.

Una base di contratto è stata presa da quella messa a disposizione degli autori di ERC-6956 ma è stata completamente rivista ed aggiornata utilizzando l'ultima versione di *OpenZeppelin* [2].

Ora andiamo a vedere le principali funzioni dello smart contract di cui abbiamo parlato nella sezione di progettazione. In questa sezione abbiamo parlato di una funzione `transferAnchor`, lo scopo di questa funzione è quella di trasferire un NFT se l'attestazione fornita è valida e possiamo vedere la sua implementazione nel listato 3.1.

Nell'implementazione iniziale `transferAnchor` prevedeva anche il caso di creazione di un NFT ma come vedremo nella sezione dedicata ai subgraph, serve qualcosa di più specifico, per questo motivo la funzione iniziale `transferAnchor` è stata suddivisa in due.

[1] Per un approfondimento su Solidity si rimanda al link: https://soliditylang.org/

[2] Per un approfondimento su OpenZeppelin si rimanda al link: https://docs.openzeppelin.com/contracts/5.x/

Passiamo quindi ora a vedere l'implementazione della funzione di creazione, visibile nel listato 3.2.

Come possiamo vedere la differenza tra le due funzioni è minima, viene sempre fatto un controllo sull'ancora, ma nella funzione `createAnchor` viene verificato che l'ancora non sia già presente nella blockchain, e che quindi non sia stato già creato un NFT con quell'ancora ed ovviamente viene chiamato il metodo `_safeMint`, implementato in ERC-721. Invece nella funzione `transferAnchor` il controllo sull'ancora verrà eseguito per verificare che l'ancora esista davvero, e che quindi esiste un asset reale collegato a quell'ancora, e per finire verrà chiamato il metodo `_safeTransfer`, implementato anch'esso in ERC-721.

Ciò che accomuna queste due funzioni però è il trasferimento di un token, infatti anche se non è possibile vederlo nei due listati, verrà emesso un evento `AnchorTransfer` che indicherà il successo del passaggio del token da un proprietario ad un altro. Nel caso della funzione `createAnchor`, il primo proprietario sarà indicato con il `NULL_ADDRESS`, ovvero l'indirizzo `0x00`.

```
1  function transferAnchor(bytes memory attestation, bytes
       memory data) public virtual
2  {
3      bytes32 anchor;
4      address to;
5      bytes32 attestationHash;
6
7      (to, anchor, attestationHash) = decodeAttestationIfValid(
           attestation, data);
8
9      _commitAttestation(to, anchor, attestationHash);
10
11     uint256 fromToken = tokenByAnchor[anchor];
12     address from = address(0);
13
14     _anchorIsReleased[anchor] = true;
15
16     require(fromToken > 0, "Anchor doesn't exist");
```

```
17      if(fromToken > 0) {
18          from = _ownerOf(fromToken);
19          require(from != to, "ERC6956-E6");
20          _safeTransfer(from, to, fromToken, "");
21      }
22  }
```

Listing 3.1: Listaggio della funzione transferAnchor in Solidity

```
1  function createAnchor(bytes memory attestation, string memory
       cid, bytes memory data) public virtual
2  {
3      bytes32 anchor;
4      address to;
5      bytes32 attestationHash;
6
7      (to, anchor, attestationHash) = decodeAttestationIfValid(
           attestation, data);
8
9      _commitAttestation(to, anchor, attestationHash);
10
11     uint256 fromToken = tokenByAnchor[anchor];
12
13     _anchorIsReleased[anchor] = true;releases the anchor
14     console.log("tokenID ");
15     console.log(fromToken);
16     require(fromToken == 0, "Anchor already exists");
17     console.log("giusto _safeMint()");
18     _safeMint(to, anchor, cid);
19 }
```

Listing 3.2: Listaggio della funzione createAnchor in Solidity

Parte fondamentale, oltre alle funzioni di creazione e trasferimento è la funzione di decodifica dell'attestazione per poter verificare se questa è valida. La funzione in questione è `decodeAttestationIfValid`, questa permette di estarre l'attestazione, potendo così verificare che la firma applicata sull'attestazione sia effettivamente dell'oracolo, altrimenti se così non fosse l'attestazione non viene considerata valida. Inoltre, nell'attestazione non viene solo

passata l'ancora ma anche dei dati utili a capire se questa è ancora valida, come ad esempio `validStartTime` e `validEndTime`, che sono degli orari di inizio e fine della validità dell'attestazione.

Per completezza nella funzione `decodeAttestationIfValid`, è presente la funzione `_extractSigner`, quest'ultima serve per estrarre l'indirizzo dal messaggio firmato in modo da poter verificare se l'indirizzo corrisponde a quello dell'oracolo specificato.

ERC6956Full

Nel progetto finale è presente un ulteriore smart contract rispetto ai due che abbiamo nominato fino a qui, questo perché è stato scelto di implementare le funzionalità base di ERC-6956 in un contratto e per ragioni di modularità il contratto `ERC6956Full` estenderà il contratto `ERC6956` con alcune interfacce aggiuntive. Le interfacce di cui stiamo parlando sono `IERC6956ValidAnchors` e `IERC6956Floatable`.

Mentre `IERC6956Floatable` è una interfaccia utile solo per operazioni di poco conto per le necessità di questo sistema, invece l'interfaccia `IERC6956ValidAnchors` è molto utile, come spiegato nella sezione di implementazione. Infatti permette di avere un lista di ancore valide, ovvero una lista di ancore che fanno parte del sistema e che quindi possono essere utilizzato da un asset reale.

In particolare è la funzione `updateValidAnchors`, che emette l'evento `ValidAnchorsUpdate`, che quando richiamata permette di aggiornare le ancore valide. Ovviamente può essere richiamata solo dal mantainer dichiarato al momento del deploy del contratto.

Il nocciolo di questo concetto è riassunto dalla funzione `anchorValid` che possiamo vedere nel listaggio 3.3. Come è possibile notare per verificare che l'ancora sia effetivamente nella lista delle ancore valide, gli autori di ERC-6956 suggeriscono l'utilizzo dei MerkleTree. Questo è un modo compatto ed efficiente per verificare velocemente la validità dell'ancora. Per un appro-

fondimento sul funzionamento dei MerkleTree rimandiamo al link a piè di pagina [3].

```
function anchorValid(bytes32 anchor, bytes32[] memory proof)
    public virtual view returns (bool)
{
    return MerkleProof.verify(
        proof,
        _validAnchorsMerkleRoot,
        keccak256(bytes.concat(keccak256(abi.encode(anchor)))
            ));
}
```

Listing 3.3: Listaggio della funzione anchorValid in Solidity

NFTMarketplace

Ora che abbiamo mostrato l'implementazione dei due smart contract principali, passiamo a vedere l'implementazione dello smart contract rimanente, ovvero `NFTMarketplace`.

Come accennato nella sezione di implementazione, questo contratto si occuperà di tutte le operazioni che permettono la gestione di NFT, nel nostro caso NFT basati su ERC-6956.

La particolarità principale di questo contratto è ovviamente la funzione `redeemItem`, infatti come detto nel capitolo precedente, questa funzione si occuperà del trasferimento vero e proprio dell'NFT da un address ad un altro, ma solo dopo che l'utente possa provare di essere davvero in possesso dell'asset reale, quindi passando nella transazione anche l'attestazione firmata dall'oracolo. In realtà questa funzione non si occupa direttamente del trasferimento del token, ma dopo aver fatto i controlli del caso sulla validità dell'attestazione, verrà richiamata la funzione `transferAncor` vista nel listato 3.1, che si occuperà del trasferimento vero e proprio. Per finire verrà

[3] Approfondimento su Merkle Tree e la loro implementazione in OpenZeppelin: https://github.com/OpenZeppelin/merkle-tree

emesso l'evento `ItemRedeemed` che indicherà esplicitamente il nuovo proprietario dell'NFT, e quindi dell'asset reale. Possiamo vedere l'implementazione di questa funzione nel listaggio 3.4.

```
function redeemItem(address nftAddress, uint256 tokenId,
    bytes memory attestation, bytes memory data)
    public
    nonReentrant
    isToRedeem(nftAddress, tokenId, msg.sender)
{
    Transaction storage toBeRedeemed = s_toBeRedeemed[
        nftAddress][tokenId];
    require(toBeRedeemed.buyer == msg.sender, "You are not
        the buyer of this token");

    IERC6956(nftAddress).transferAnchor(attestation, data);

    delete s_toBeRedeemed[nftAddress][tokenId];

    emit ItemRedeemed(nftAddress, tokenId, msg.sender);
}
```

Listing 3.4: Listaggio della funzione redeemItem in Solidity

Ovviamente, per questo contratto ma anche per quelli precedenti, sono state applicate le cosiddette "best practices" in quanto a sicurezza dei contratti. Infatti tra i requisiti che vogliamo andare a soddisfare, non ci sono solo le funzionalità, ma anche la sicurezza dei contratti.

Per questo, abbiamo implementato nel contratto `NFTMarketplace` il modulo di sicurezza `ReentrancyGuard`, ed anche altri `modifier`, come ad esempio `isToRedeem(nftAddress, tokenId, msg.sender)` nel listato 3.4, che fanno tutti i controlli del caso, funzione per funzione, al fine di non incappare in casi sgradevoli.

Parleremo di più in seguito quando discuteremo dell'analisi della sicurezza dei contratti.

3.1.2 Oracolo

Mostrata l'implementazione degli smart contract, passiamo ora a mostrare come la parte realtiva all'oracolo sia stata simulata, come discusso nella fase di implementazione.

L'oracolo non è altro che una API route definita come specificato nel capitolo sull'implementazione, che al suo interno implementa una funzione per la firma di un messaggio data una determinata chiave privata.

3.1.3 IPFS

Nella fase di implementazione è stato fatto riferimentto ad IPFS e alla sua utilità nell'ambito della memorizzazione dei dati di un NFT, e nel nostro caso anche dell'asset reale.

Ora andremo a vedere le due principali funzioni relative alla parte su IPFS e su come queste permettano la memorizzazione dei cosiddetti metadati degli NFT.

Per prima cosa è necessario introdurre Pinata [4]. Pinata è un servizio che facilita la gestione e l'archiviazione di file su IPFS (InterPlanetary File System). IPFS, come detto in precedenza, è un protocollo di archiviazione decentralizzato che consente di memorizzare file in modo distribuito su una rete di nodi.

Pinata è quindi un servizio molto utile che ci permette di interfacciarci con IPFS senza però doverci curare di tutta la parte di gestione. Usando questo servizio quindi possiamo usufruire di API appositamente messe a disposizione da Pinata per interagire con IPFS e memorizzare dati al suo interno.

Appurata l'utilità pratica di questo servizio, è stato pensato che fosse utile utilizzarlo per memorizzare i metadati e le immagini di ogni NFT. Le API che vedremo nei prossimi listaggi sono `pinFileToIPFS` e `pinJSONToIPFS`.

[4] Per ulteriori informazioni su Pinata si ramanda a: https://docs.pinata.cloud/frameworks/next-js

Ora non rimane che capire quali siano i metadati che devono essere memorizzati per ogni NFT.

L'immagine dell'NFT sarà sicuramente fra questi, ma la soluzione migliore è sembrata quella di memorizzare l'immagine in modo separato rispetto ai metadati testuali. Quindi, verrà in un primo momento salvata l'immagine dell'NFT, e poi si dovrà in qualche modo salvare l'indirizzo in cui risiede l'immagine insieme a tutti gli altri metadati. QUindi, la scelta è ricaduta su dati con la struttura mostrata di seguito:

```
{
  title: string,
  description: string,
  imageURI: string,
  tags: String[],
}
```

Come si può notare, è presente un link ad un altro dato memorizzato su Pinata, e quindi su IPFS. Si tratta dell'immagine relativa all'NFT in questione, questo è dovuto al fatto che immagini e JSON debbano essere memorizzati in modo differente, inoltre come vedremo nella parte frontend, è molto utile avere a disposizione il link all'immagine direttamente nei metadati.

3.1.4 Subgraph

Per quanto riguarda i subgraph l'idea è stata simile alla sezione precedente. Pensare di creare e gestire un sistema decentralizzato che potesse aiutare all'indicizzazione off-chain sarebbe un'opzione, probabilmente la migliore per avere un sistema il più decentralizzato possibile, ma i costi sarebbero notevoli. Quindi la soluzione più logica è sembrata l'utilizzo di un servizio che permette tutto questo, ovvero Subgraph Studio [5].

[5]Per un approfondimento su TheGraph si ramanda a: https://thegraph.com/docs/

Subgraph Studio, permette il deploy di subgraph nella sua rete, così che i nodi all'interno possano iniziare ad indicizzare i dati, come specificato nei subgraph, rispetto al network indicato.

Prima di procedere a mostrare l'implementazione vera e propria, facciamo chiarezza su come implementare i subgraph.

Per implementare i subgraph è stato utilizzato un package messo a diposizione direttamente da Subgraph Studio: `@graphprotocol/graph-cli`. Questo permette di crare dei file chiamati `schema.graphql` che specificano come i dati da indicizzare sono organizzati e quali relazioni ci sono tra loro. Inoltre, dovranno essere implementati altri due file: il primo è `subgraph.yaml`, file che contiene le specifiche per il deploy dei subgraph, e il secondo è `mapping.ts`, questo contiene le funzioni che dovranno gestire l'indicizzazione, ovvero dovranno creare, eliminare o modificare i dati a seconda dell'evento che si sta andando a gestire.

3.1.5 Backend

Per quanto riguarda l'implementazione del backend, la scelta è stata basata sulle necessità di avere una parte frontend performante, per di più la parte backend del sistema è piuttosto snella e non richede particolari tecnologie, deve solo occuparsi delle chiamate API che abbiamo visto nella sezione sull'oracolo (3.1.2) e su IPFS (3.1.3).

Quindi dopo un'attenta revisione delle varie tecnologie utilizzate nel mondo del Web3.0, la scelta è ricaduta su Next.js [6].

Next.js è un framework per React che permette lo sviluppo di applicazioni web sia lato frontend (interfaccia utente) che backend (logica di server), facilitando lo sviluppo full-stack in un unico ambiente.

Next.js permette di generare pagine dinamiche sul server ad ogni richiesta dell'utente, il che migliora la SEO e la velocità iniziale di caricamento delle pagine. Rispetto a un'app React tradizionale, dove il rendering avviene

[6]Per ulteriori informazioni su Next.js si ramanda a: https://nextjs.org/docs

completamente lato client (Client-Side Rendering, CSR), Next.js permette di bilanciare dove viene fatto il rendering.

Inoltre, Next.js permette di creare API direttamente all'interno dell'applicazione con le API Routes, che agiscono come backend serverless. Questo significa che puoi gestire richieste HTTP e servire dati senza bisogno di configurare un backend separato, il che rende lo sviluppo più rapido.

Per via di questi motivi, è sembrato il framework più adatto per l'implementazione del marketplace.

Una volta scelto Next.js, la scelta successiva è stata fatta sull'approccio per la gestione delle route. L'approccio tradizionale chiamato Page Router e quello più innovativo chiamato App Router.

Il primo approccio è relativamente rigido in termini di flessibilità, e non permette layouts nidificati. Il secondo approccio rappresenta un cambio di paradigma per il routing e sfrutta una nuova architettura basata sui React Server Components [7], permettendo maggiore flessibilit; layouts nidificati e icomponenti vengono prima renderizzati sul server per migliorare le prestazioni e ridurre la dimensione del JavaScript che deve essere inviato al client.

Quindi, nonostante una single-page application sembrerebbe poter essere implementata in modo efficace con Page Router, è stato scelto l'approccio App Router. Questo per via della maggiore flessbilità e della struttura modulare del progetto, ma soprattutto perché il miglioramento delle performance che deriva dai React Server Components è stato il tassello che ha fatto propendere su questo approccio.

3.1.6 Frontend

Nella precedente sezione dove abbiamo parlato del backend, abbiamo illustrato Next.js, ed abbiamo specificato i motivi per cui è stato scelto questo framework. Abbiamo detto che permette la creazione di API routes server-

[7] Per ulteriori informazioni su Server e Client Components in Next.js si rimanda a: https://nextjs.org/learn/react-foundations/server-and-client-components

less facilitando quindi la creazione della parte backend, ma oltre a questo anche la parte frontend risulta di facile comprensione e modulare.

Infatti l'approccio basato su App Router permette di creare componenti riusabili e layout annidati che combinati tra loro permettono veramente uno sviluppo agevole e comprensibile.

Per quanto riguarda la creazione vera e propria dell'interfaccia del marketplace è stato pensato che avere componenti riutilizzabili e già testati fosse un plus che potesse aumentare l'affidabilità del progetto e soprattutto è stato pensato che questo portasse vantaggi notevoli anche nello sviluppo vero e proprio.

Per questo motivo la scelta non è ricaduta sui soliti framework, bensì su Shadcn/UI [8].

Shadcn/UI è una libreria di componenti UI per React, progettata per essere flessibile e facilmente personalizzabile. Utilizza Tailwind CSS come sistema di stilizzazione predefinito, ma è pensata per essere framework-agnostic, il che significa che si può integrarla facilmente con altre librerie o framework CSS.

Una delle caratteristiche principali di Shadcn/UI è che permette di costruire l'interfaccia utente senza dover dipendere da una libreria esterna, in quanto fornisce i componenti sotto forma di codice che si può copiare e modificare direttamente nel proprio progetto. Questo dà pieno controllo sull'aspetto e sul comportamento dei componenti, senza vincoli imposti da un design system predefinito.

Shadcn/UI sta prendendo sempre più piede per via di queste sue caratteristiche e il fatto di non dover scaricare e dipendere da una libreria esterna è stato decisivo per la scelta di questa libreria.

Ovviamente, meno librerie da cui dipendere significa meno vincoli per la progettazione della parte grafica, ma anche una bundle size più snella e il codice direttamente modificabile permette anche di aumentare ulteriormente la personalizzazione e le performance generali dell'app.

[8] Per ulteriori informazioni su Shadcn/UI si ramanda a: https://ui.shadcn.com/docs

L'ultimo tassello mancante riguarda le query che vengono svolte dall'applicazione verso Subgraph Studio.

Vista la necessità di avere un wallet collegato all'applicazione, per forza di cose le query devono essere svolte lato client, per questo e per via della grande notorietà del package è stato scelto di utilizzare Apollo Client [9].

Il setup di Apollo è molto facile e permette di creare un file separato per la definizione delle query, questo file è stato chiamato `subgraphQuery.ts`.

Per maggiore sicurezza lato frontend, le query sono anche state tipate, in modo che la risposta di una query si possa già sapere di che tipo sia, e quindi quali dati contenga. Sembrerebbe un passaggio superfluo, ma per evitare errori indesiderati lato utente, in realtà è un plus che migliora l'esperienza dell'utente, oltre che la leggibilità del codice scritto.

Per concludere, è stata fatta una panoramica sulle tecnologie utilizzate per lo sviluppo della parte frontend, ma per consultare l'interfaccia grafica finale rimandiamo al capitolo successivo (4.1) sui risultati, nella sezione dedicata all'applicazione finale.

3.1.7 Tecnologia dell'ancora - Arduino

Per finire l'ultimo componente di cui andiamo a discutere l'implementazione è la parte relativa alla tecnologia dell'ancora.

La scelta del sistema di computazione da utilizzare per mostrare l'ancora è ricaduta su un Arduino munito di display, in modo da poter mostrare l'ancora.

In questo caso, avendo scelto Arduino dovremo andare a vedere quali componenti fisici sono più appropriati per mostrare un codice QR (di dimensioni quadrate).

Per prima cosa, non avendo a disposizione fisicamente un Arduino è stato scelto di simulare questa parte tramite Wokwi [10]. Wokwi è una piattafor-

[9] Per un approfondimento su Apollo Client si ramanda a: https://www.apollographql.com/docs/react

[10] Per ulteriori informazioni su Wokwi si ramanda a: https://wokwi.com/

ma online che consente di simulare circuiti elettronici, in particolare progetti basati su microcontrollori come Arduino, ESP32 e Raspberry Pi Pico, direttamente nel browser. Gli utenti possono creare e testare il loro codice senza dover utilizzare hardware fisico. La piattaforma supporta diversi componenti elettronici come LED, sensori, display OLED, motori e altri, permettendo di costruire e sperimentare progetti di elettronica in modo interattivo e facile.

Quindi, appurato che in Wokwi (comunemente usato per creare dei prototipi) si può simulare la programmazione in Arduino, si dovrà decidere tra i possibili componenti quali siano effettivamente i più consoni per l'obiettivo finale.

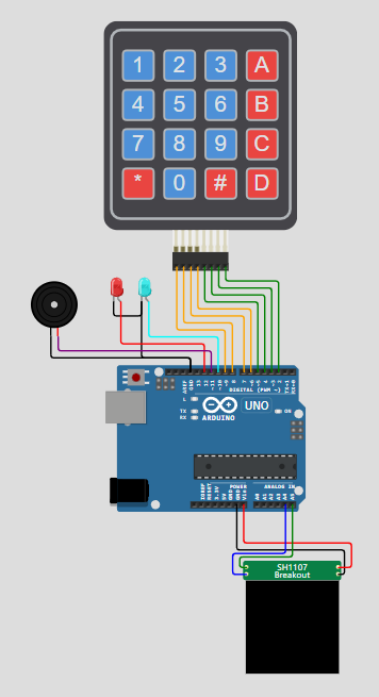

Figura 3.1: Diagramma di Arduino con display.

Dovendo mostrare l'immagine del codice QR dell'ancora, sarà necessario avere un display, sarebbe anche comodo che questo display fosse quadrato per via della natura del codice QR. Per questi motivi il componente che è sembrato più consono è stato il display 128x128 SH1107 OLED.

Per mostrare l'immagine poi è stata scelta la libreria u8g2. Il perché è derivato dal fatto che è ideale per display monocromatici con memoria limitata. Mentre, ad esempio, Adafruit è più orientata ai display a colori (TFT). Nel nostro caso avendo un display che deve mostrare immagini in bianco e nero, è sembrata un scelta logica utilizzare u8g2.

Per finire, sono stati implementati tre componenti di contorno: un keypad, un buzzer e due led. Il keypad è stato utilizzato per mostrare il codice QR solo dopo che l'utente abbia inserito la corretta password; può essere visto come un codice di sicurezza per non far mostrare il codice QR costantemente.

In aggiunta al keypad sono presenti un buzzer e due led di colori diversi. Questo perché è stato pensato potesse essere carino, a livello di esperienza dell'utente, sapere se e quando è stato premuto un pulsante nel keypad e una volta inserita la password, mostrerà una lucina colorata in base a se la password inserita sia corretta o meno. Possiamo vedere in figura 3.1 la realizzazione di questa parte in Arduino in Wokwi.

3.2 Validazione e Test

In seguito all'implementazione delle funzionalità principali, una fase altrettanto importante del lavoro ha previsto la realizzazione di test finalizzati a stabilizzare il codice e d'assicurarsi che tutto funzionasse come previsto, in modo da procedere in maniera relativamente sicura al successivo deploy dei contratti su una testnet reale.

Questa fase però non è confinata solamente ai contratti, bensì anche alla parte relativa ai subgraph, infatti prima di fare un deploy dei subgraph è utile verificare che questi si comportino come vogliamo, altrimenti potremmo avere dei dati salvati off-chain che non corrispondano effetivamente con le operazioni eseguite on-chain.

Va specificato, che quest test sono stati eseguiti in ambiente locale utilizzando quindi *Hardhat* e altri tool che vedremo in seguito. Invece per quanto

riguarda i test sperimentali fatti su testnet Sepolia, vedremo i risultati nel prossimo capitolo.

Quindi la presente sezione è organizzata in due parti: la prima relativa alla verifica e al testing degli smart contract, mentre la seconda parte è relativa alla verifica e al testing dei subgraph.

3.2.1 Test degli smart contract

Eseguire dei test è fondamentale nel processo di sviluppo degli smart contract; buona parte dell'esito del software, infatti, dipende dalla capacità di scrivere test efficaci. Assicurarsi che il codice si comporti come preventivato in fase di realizzazione, inoltre, risulta di grande rilievo specialmente nel mondo decentralizzato delle blockchain, dove gli smart contract, una volta registrati, non possono essere modificati, sono visibili a tutti ed, in linea di principio, chiunque vi può interagire, motivo per cui sono potenzialmente soggetti ad exploit. Scrivere dei test robusti, dunque, è sicuramente la prima linea di difesa.

Per prima cosa, procediamo alla verifica della sicurezza dei contratti. L'obiettivo è quello di verificare che i contratti siano sicuri rispetto agli attachi possibili su di essi. Possiamo vedere quelli più comuni nell'articolo [11].

Per fare questo utilizziamo il tool `Mythril`, comunemente utilizzato in queste situazioni. Questo tool prevede un comando `analyze` che chiamato per ogni contratto mostrerà alla fine dell'analisi eventuali criticità del contratto, se presenti.

Per un riepilogo sul perchè `Mythril` viene utilizzato e sui suoi vantaggi si rimanda all'articolo [10].

Mostriamo quindi in figura 3.2 l'analisi delle criticità eseguita sul contratto `ERC6956`, in figura 3.3 l'analisi sul contratto `ERC6956Full` e in figura 3.4 l'analisi sul contratto `NFTMarketplace`.

Ora che abbiamo appurato la sicurezza degli smart contract scritti contro i principali attacchi conosciuti, passiamo a vedere la correttezza del codice scritto.

```
matteowsl@DESKTOP-OM140T8:~/project/tesi/backend$ /home/matteowsl/
.local/bin/myth analyze ./contracts/ERC6956.sol --solc-json solc-j
son.json
The analysis was completed successfully. No issues were detected.
```

Figura 3.2: Risultato dei test della sicurezza su ERC6956.

```
matteowsl@DESKTOP-OM140T8:~/project/tesi/backend$ /home/matteowsl/
.local/bin/myth analyze ./contracts/ERC6956Full.sol --solc-json so
lc-json.json
The analysis was completed successfully. No issues were detected.
```

Figura 3.3: Risultato dei test della sicurezza su ERC6956Full.

La tecnologia utilizzata per verificare la correttezza dei contratti è sempre *Hardhat*, che attraverso uno dei suoi script predefiniti, ovvero `hardhat test` permette di eseguire il controllo attraverso degli script scritti dall'utente e inseriti all'interno della cartella di lavoro `test`.

Nelle figura 3.5 troviamo i risultati dei test realtivi al contratto `ERC6956`, in figura 3.6 troviamo i risultati dei test realtivi al contratto `ERC6956Full` e per finire in figura 3.7 troviamo i risultati dei test realtivi al contratto `NFTMarketplace`.

Queste immagini mostrano come i test siano stati completati con successo, ma non mostrano se i test effettuati abbiano verificato la totalità dei contratti, in altre parole non dimostra che tutto il codice scritto sia stato testato, mostra solo che quanto scritto nei file di test sia corretto. Quindi per mostrare che la totalità del codice dei contratti sia stato testato, utilizziamo un tool chiamato *solidity-coverage*, che una volta utilizzato permette di verificare la completezza dei test effettuati. In figura 3.8 ne mostriamo il risultato.

Per finire mostriamo in tabella 3.1 il riassunto del *gas-report* che viene fatto in automatico da *Hardhat* lanciando il comando `test`. Possiamo notare come le funzioni più dispendiose siano ovviamente quelle legate al trasferimento di un token, quindi `createAnchor` e `transferAnchor`, ed ovviamente anche `redeemItem` che richiama al suo interno `transferAnchor`.

```
matteowsl@DESKTOP-OM140T8:~/project/tesi/backend$ /home/matteowsl/
.local/bin/myth analyze ./contracts/NFTMarketplace.sol --solc-json
 solc-json.json
The analysis was completed successfully. No issues were detected.
```

Figura 3.4: Risultato dei test della sicurezza su NFTMarketplace.

```
ERC6956: Asset-Bound NFT --- Basics
  Deployment & Settings
    ✔Should implement EIP-165 support the EIP-6956 interface
  Authorization Map tests
    ✔SHOULD interpret ERC6956Authorization correctly
  Attestation-based transfers
    ✔SHOULD not allow non-trusted oracles to issue attestation
    ✔SHOULD allow mint and transfer with valid attestations
    ✔SHOULDN'T allow safeTransfer per default
    ✔SHOULDN'T allow approveAnchor followed by safeTransfer w anchor not floating
    ✔SHOULDN't allow using attestations before validity
  ERC721Burnable-compatible behavior
    ✔SHOULD burn like ERC-721 (direct)
    ✔SHOULD burn like ERC-721 (approved)
    ✔SHOULD allow issuer to burn
    ✔SHOULD burn like ERC-721 (via attestation-approved)
    ✔SHOULD burn like ERC-721 (attestation)
    ✔SHOULD use same tokenId when anchor is used again after burning
  Metadata tests
    ✔SHOULD allow only maintainer to update baseURI
    ✔SHOULD use anchor for tokenURI
```

Figura 3.5: Risultato dei test relativi al contratto ERC6956.

3.2.2 Test dei subgraph

Mostrati i test effettuati sugli smart contract, passiamo a vedere come poter testare anche la parte relativa ai subgraph.

Anche qui, la parte di test è essenziale per verificare che i file di `mapping` scritti nella parte dei subgraph, svolgano effettivamente ciò che vogliamo.

Per questa verifica è stato utilizzato un tool diverso da *Hardhat*, visto che il codice scritto in questi file è in linguaggio *AssemblyScript* [11]. Per questo motivo il tool utilizzato è stato *MatchStick-as* [12], permette la scrittura di file che simulano la creazione degli eventi emessi dalla blockchain per le

[11] Per ulteriori informazioni riguardo lo sviluppo in AssemblyScript si rimanda a: https://thegraph.com/docs/en/developing/graph-ts/api/

[12] Per un approfondimento su Matchstick-as si ramanda a: https://github.com/LimeChain/matchstick

```
ERC6956: Asset-Bound NFT --- Full
  Deployment & Settings
    ✔Should implement EIP-165 support the EIP-6956 interface
  Valid Anchors (merkle-trees)
    ✔SHOULDN't allow attesting arbitrary anchors
  Anchor-Floating
    ✔SHOULD only allow maintainer to specify canStartFloating and canStopFloating
    ✔SHOULD only allow maintainer to modify floatAll behavior w/o affecting previous tokens
    ✔SHOULD allow owner to float token only when OWNER is allowed
    ✔SHOULD only allow owner to transfer token when floating
    ✔SHOULDN'T allow owner to transfer token when explicitly marked anchored
    ✔SHOULD allow maintainer to float ANY token only when ISSUER is allowed
    ✔SHOULD allow maintainer to float HIS OWN token when OWNER is allowed
    ✔SHOULD allow approveAnchor followed by safeTransfer when anchor IS floating
```

Figura 3.6: Risultato dei test relativi al contratto ERC6956Full.

```
NFT Marketplace Tests
  ✔SHOULD allow listing and buying
  ✔SHOULD allow listing of NFT and cancel listing
  ✔SHOULDN'T allow listing of NFT when already listed
```

Figura 3.7: Risultato dei test relativi al contratto NFTMarketplace.

nostre operazioni, e una volta chiamati questi eventi riusciamo a vedere se gli `handlers` che sono stati scritti nei `mapping` svolgono ciò che devono. Mostriamo quindi in figura 3.9 i risultati relativi ai test eseguiti sui subgraph.

Anche in questo caso però, per quanto possiamo notare come i test siano stati completati con successo, non dimostra che siano stati eseguiti su tutto il codice scritto per i subgraph. Per questo, anche in questo caso, è stato previsto un controllo che dimostri che tutti gli `handlers` scritti siano stati testati. Vediamo quindi in figura 3.10 i risultati del controllo sulla copertura dei test. Non è stato necessario installare nessun tool aggiuntivo in quanto il package `graph` prevede nel suo ambiente un comando ad hoc.

File	% Stmts	% Branch	% Funcs	% Lines
contracts/	98.73	77.38	94.83	97.76
ERC6956.sol	99.05	77.27	94.29	97.89
ERC6956Full.sol	100	84.38	100	100
IERC6956.sol	100	100	100	100
IERC6956AttestationLimited.sol	100	100	100	100
IERC6956Floatable.sol	100	100	100	100
IERC6956ValidAnchors.sol	100	100	100	100
NFTMarketplace.sol	96.43	72.92	91.67	95.92
All files	98.73	77.38	94.83	97.76

Figura 3.8: Risultato del controllo di solidity-coverage.

Solc version: 0.8.20		**Runs: 200**	**Block limit: 30000000 gas**
Methods			
Contract	**Method**	**Avg**	**# calls**
ERC6956Full	approve	51606	2
ERC6956Full	createAnchor	294138	6
ERC6956Full	transferAnchor	155841	8
ERC6956Full	transferFrom	111145	4
NFTMarketplace	buyItem	110194	2
NFTMarketplace	cancelListing	32128	2
NFTMarketplace	listItem	78521	6
NFTMarketplace	redeemItem	167927	4
Deployments			**% of limit**
ERC6956		2625406	8.8 %
ERC6956Full		3171269	10.6 %
NFTMarketplace		1076276	3.6 %

Tabella 3.1: Gas-report relativo ai test in locale.

```
___  ___      _        _          _   _       _
|  \/  |     | |      | |        | | (_)     | |
| .  . | __ _| |_ ___| |__  ___| |_ _  ___| | __
| |\/| |/ _` | __/ __| '_ \/ __| __| |/ __| |/ /
| |  | | (_| | || (__| | | \__ \ |_| | (__|   <
\_|  |_/\__,_|\__\___|_| |_|___/\__|_|\___|_|\_\

Compiling...

💬 Compiling ipfs-data/ipfs-data...
💬 erc-token/erc-token skipped!

Igniting tests 🔥

erc-token/erc-token
-------------------------------------------------
  NFT Marketplace events test:
    √ handleItemListed SHOULD list a Token - 0.435ms
    √ handleItemCanceled SHOULD cancel listing of a Token - 0.755ms
    √ handleItemBought SHOULD buy a Token - 0.531ms
    √ handleItemRedeemed SHOULD redeem a Token buyed - 0.726ms

ipfs-data/ipfs-data
-------------------------------------------------
  IPFS-data creation test:
    √ handleAnchorTransfer SHOULD create a new Token - 0.946ms

All 5 tests passed! 😎
```

Figura 3.9: Risultato dei test relativi ai subgraph.

Figura 3.10: Risultato relativo alla copertura dei test sui subgraph.

Capitolo 4

Risultati

In questo capitolo verranno illustrati i risultati ottenuti dall'implementazione del progetto, con l'obiettivo di rispondere alle domande e verificare le ipotesi formulate nel capitolo introduttivo.

Prima di passare a parlare dei risultati, il link realtivo alla repository del progetto implementato è https://github.com/Bona612/tesi, mentre il link relativo alla parte Arduino in *Wokwi* è possibile consultarla al link https://wokwi.com/projects/408504547254311937.

Inoltre si vuole specificare che il progetto è live al seguente link https://tesi-9t39xd7nv-bona612s-projects.vercel.app. Come si può notare è stata scelta *Vercel* [1] come piattaforma per il deploy dell'applicazione.

Il capitolo è strutturato in due sezioni principali: nella prima parte verranno mostrati degli screenshot della parte frontend, per mostrare come questa risulti intuitiva e completa, nella seconda sezione invece verranno mostrati i risultati principali relativi a valutazioni quantitative sulle transazioni eseguite sulla testnet Sepolia.

[1] Per ulteriori informazioni su Vercel si ramanda a: https://vercel.com/docs

4.1 Applicazione finale

Nella presente sezione andiamo quindi a mostrare i risultati finali dell'interfaccia grafica del marketplace.

Come detto precedentemente la parte grafica è volutamente semplicistica ma senza far mancare nessuna delle funzionalità di base che abbiamo visto nei capitoli precedenti. Per di più, l'interfaccia è stata creata in modo tale da essere responsive in ogni dispositivo possibile, partendo dagli smartphone, passando per i tablet e arrivando a computer veri e propri.

Prima di passare in rassegna tutte le varie pagine dell'applicazione, facciamo un breve accenno alla figura 4.1, possiamo vedere l'utilizzo di un componente predefinito del package `Web3Modal`, che permette di creare bottoni per collegare un wallet ad una applicazione decentralizzata.

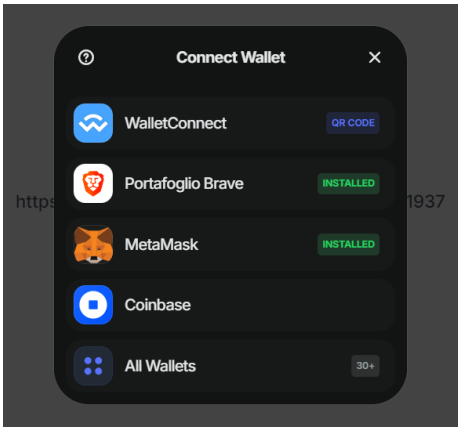

Figura 4.1: Wrb3Modal Connect Wallet.

La sezione sarà divisa in quattro parti: nella prima parte andremo a vedere la pagina reltiva alla creazione di un NFT, nella seconda parte mostreremo la sezione relativa agli NFT di un utente collegato tramite wallet, nella terza parte mostreremo la pagina relativa al marketplace e nell'ultima parte vedremo la pagina relativa alle informazioni sul singolo NFT.

4.1.1 Create NFT

La prima parte dell'applicazione che andiamo a vedere è la pagina relativa alla creazione di un NFT.

In figura 4.2 e in figura 4.3 possiamo vedere come è strutturata la pagina. In altro come prima cosa possiamo vedere il menu che è una costante in ogni pagina, visto che è da qui che si può navigare tutta l'applicazione. Si arriva a questa pagina cliccando sul bottone "Create NFT", e come è possibile vedere viene mostrata una scheda contenente un form che l'utente dovrà compilare con i dati relativi al proprio asset reale. La parte succosa in questo caso è il bottone di "Redeem" che se cliccato apre un *dialog* che permetterà all'utente di leggere il codice QR relativo all'ancora collegata all'asset. A questo punto, l'utente cliccando sul bottone "Create" vedrà aprirsi una nuovo *dialog* dove potrà confermare o meno la creazione dell'NFT. Se viene cliccato il bottone di conferma, partirà allora la transazione, ovviamente se l'utente ha collegato un *Wallet* all'applicazione, e vedrà comparire una transazione sul *Wallet* che dovrà confermare e firmare. Al completamento della transazione, l'utente riceverà una notifica di conferma da cui potrà capire se la transazione è andata a buon fine o meno. Se sarà andata a buon fine, potrà cliccare nel bottone "My NFT" nel menu, e potrà vedere il suo nuovo NFT creato.

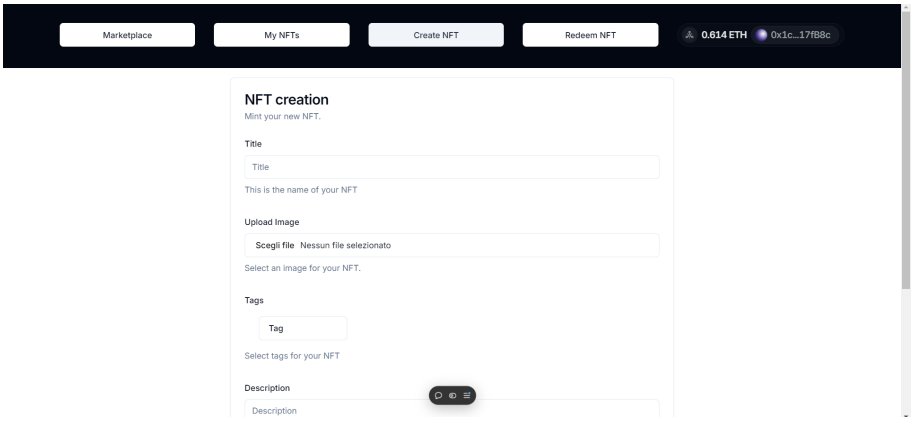

Figura 4.2: Pagina relativa alla creazione di un NFT.

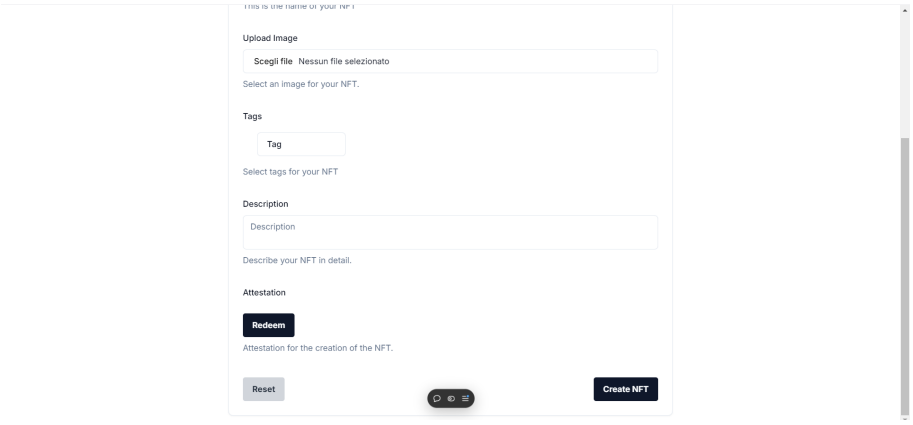

Figura 4.3: Pagina relativa alla creazione di un NFT.

4.1.2 My NFT

La parte successiva dell'applicazione che andiamo a vedere è la pagina relativa alla visualizzazione e gestione dei propri NFT.

In figura 4.4 possiamo vedere come è strutturata la pagina. Si può notare, come sempre, il menu in alto, ma questa volta non viene mostrata nessuna scheda con un form, bensì è presente un *header* con una serie di componenti, che vedremo a breve, e a seguire si può vedere il corpo della pagina dove è presente una lista di NFT, che in questa pagina corrispondono agli NFT dell'utente collegato all'applicazione. L'*header* è composto da una barra di ricerca, utile per cercare tramite delle parole l'NFT che si vuole trovare; sono presenti anche un componente per eseguire una ricerca basata sui "Tag" di un NFT, e per finire avremo due componenti che sono adibiti all'ordinamenteo della visualizzazione degli NFT.

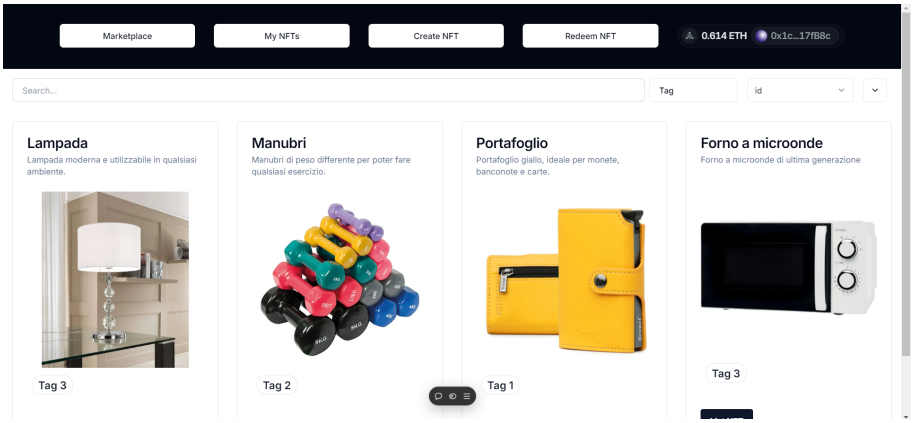

Figura 4.4: Pagina degli NFT di un utente.

Vediamo poi in figura 4.5, che la scheda realativa ad ogni NFT avrà in basso un bottone con la relativa operazione possibile. In questo caso possiamo vedere che il pulsante disponibile sia "List".

Per finire, mostriamo in figura 4.6, il *dialog* che si apre se si clicca nel bottone "List". Possiamo vedere come questo *dialog*, contiene anche un form al suo interno per permettere all'utente di inserire il prezzo a cui intende vendere il proprio NFT.

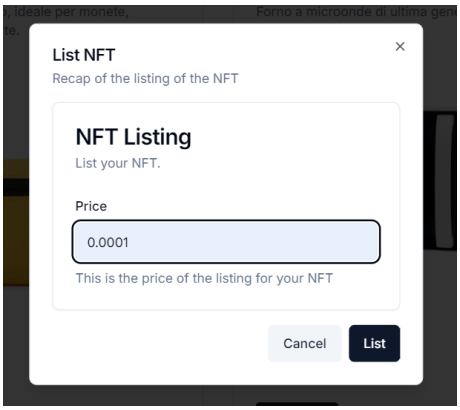

Figura 4.6: Dialog per il listaggio di un NFT.

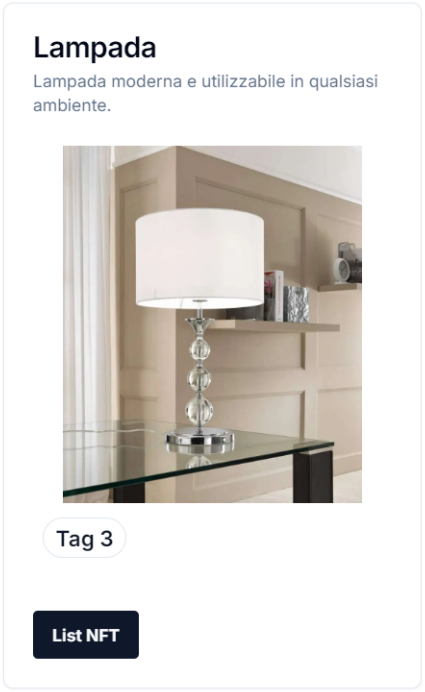

Figura 4.5: Scheda riassuntiva di un NFT.

4.1.3 Marketplace

Ora andiamo a vedere la pagina relativa al marketplace.

Per quanto riguarda, questa pagina, è esattamente uguale a quella vista nella sezione precedente (figura 4.4), l'unica cosa che cambierà saranno le operazioni possibili sugli NFT. Infatti, se nella pagina "My NFT" potremmo fare operazioni di listaggio e cancellazione, in questa potremo invece fare "Buy".

Rispetto alla figura 4.5, il bottone in basso alla scheda dell'NFT cambierà, come possiamo vedere nella figura 4.7.

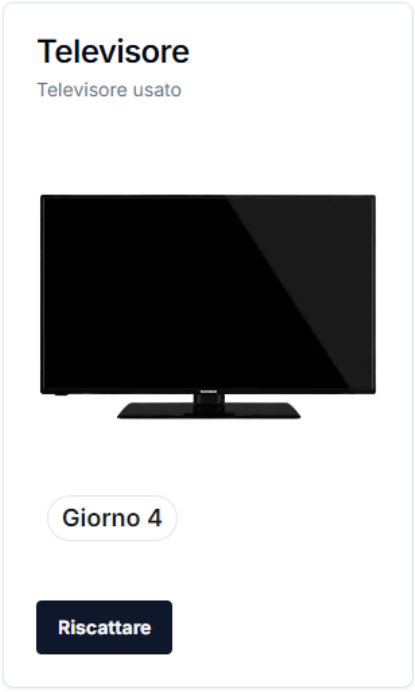

Figura 4.7: Scheda riassuntiva di un NFT, con bottone Buy.

4.1.4 NFT Info

Per finire, viene mostrato la parte più informativa dell'applicazione, ovvero quella relativa alle informazioni di un singolo NFT.

In figura 4.8 e in figura 4.9 possiamo vedere come è strutturata la pagina. Troviamo il solito menu, e a seguire una scheda con tutte le informazioni realtive all'NFT; troviamo anche un bottone "Back"che permetterà di tornare nella pagina da cui si proveniva.

La parte più interessante è visibile nella figura 4.9, dove possiamo notare delle informazioni cruciali come "Off-chain data" e "Transaction history". Come possiamo ricordare dal capitolo di progettazione, gli utenti potrebbero essere più invogliati ad usare l'applicazione se hanno a disposizione un metodo facile per verificare con chi si sta eseguendo una transazione e se l'NFT che

si sta guardando, effettivamente corrisponde con ciò che è stato scritto da colui che lo ha creato.

Ovviamente, anche qui in fondo alla scheda troviamo le possibili operazioni in base a quale NFT si sta visionando.

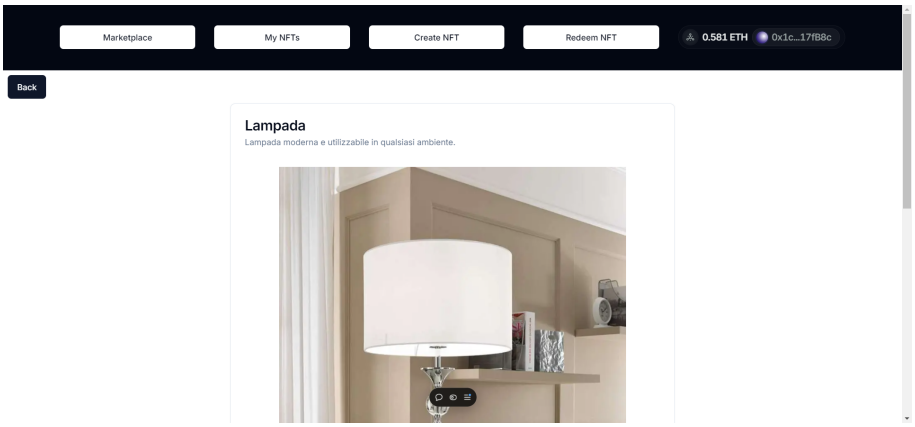

Figura 4.8: Pagina delle informazioni di un NFT.

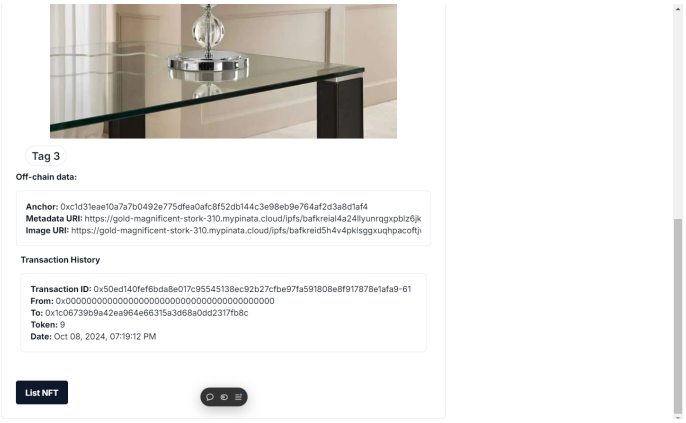

Figura 4.9: Pagina delle informazioni di un NFT.

4.1.5 Redeem

Facciamo un accenno anche alla pagina relativa al "Redeem" degli NFT.

Non mostriamo nuovi screenshot in quanto per questa pagina la grafica è esattamente uguale a quella in figura 4.4, con la sola differenza che gli NFT che vengono mostrati qui sono solamente quelli che, dopo essere stati acquistati, devono essere riscattati, facendo la verifica tramite il *dialog* in figura 4.10.

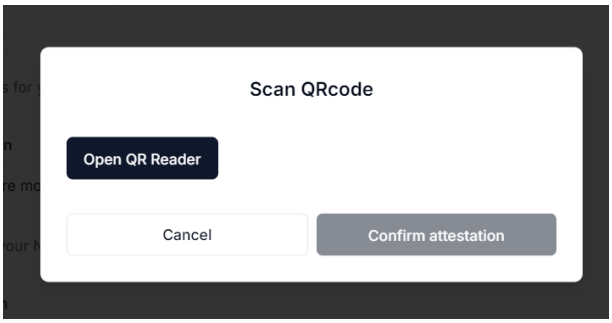

Figura 4.10: Dialog con lettore di codice QR.

4.2 Risultati quantitativi

Passate in rassegna tutte le pagine della parte frontend, mostriamo ora come sono state eseguite le prove sperimentali finali.

Tramite l'applicazione finale, sono state eseguite nella testnet Sepolia un minimo di dieci operazioni per ogni funzionalità che vogliamo andare a testare.

Ovvero: creazione di un NFT, listaggio di un NFT, cancellazione di un listaggio di un NFT, acquisto di un NFT e redeem di un NFT. In più visto che siamo in testnet dobbiamo anche consideraree che l'utente deve concedere l'approvazione al contratto `NFTMarketplace` per poter interagire con l'NFT, quindi avremo questa ulteriore funzione da chiamare prima del listaggio di un NFT.

Ricapitolando, le operazioni da testare tramite applicazione sono le seguenti:

- creazione di un NFT

- approvazione di un NFT
- listaggio di un NFT
- cancellazione di un listaggio di un NFT
- acquisto di un NFT
- redeem di un NFT

Per mostrare i dati relativi alle transazioni eseguite su Sepolia testnet è stato scelto di creare due tabelle.

La prima, tabella 4.1, indica la "Mean" (media) e la "Std Dev" (deviazione standard) relativa alla "Transaction Fee" per ogni operazione elencata precedentemente ed è visibile a seguire.

La seconda, tabella 4.2, invece indica la "Mean" (media) e la "Std Dev" (deviazione standard) relativa al "Prezzo del gas" per ogni operazione elencata precedentemente ed è visibile a seguire.

Per completezza, vengono inseriti i link alle transazioni in Etherscan eseguite dai due account utilizzati: https://sepolia.etherscan.io/address/0x1c06739B9a42EA964e66315a3d68a0Dd2317fB8c e https://sepolia.etherscan.io/address/0x8cCc6adfE4a364EdAa9C670858f7D00162639a12

Andiamo, ora a dire due parole sul risultato delle prove sperimentali.

Possiamo notare come le transazioni più dispendiose effettivamente consumano più Fee, infatti `createAnchor`, `buyItem` e `redeemItem` mediamente sono le transazioni che costano di più rispetto alle altre, anche a parità di "Prezzo del Gas".

Questo è motivato dal fatto che sia `createAnchor` sia `redeemItem` dovranno eseguire il trasferimento dell'NFT, mentre la `buyItem` è l'operazione che si occupa del pagamento del prezzo dell'NFT.

Anche `listItem` non è una operazione di poco conto, ma indubbiamente, richiede meno capacità computazionale per essere portata a termine.

Invece `approve` e `cancelListing` sono le operazioni più agevoli, devono occuparsi di impostare solo poche variabili all'interno della blockchain e quin-

di richiedono meno tempo e capacità computazionale, i quali si traducono in minore dispendio in termini di Fee.

Un appunto va fatto per l'operazione `createAnchor`, in quanto per via della memorizzazione off-chain dei metadati, dovrà ricevere in input l'URI in cui Pinata salva i metadati, questo perchè dovrà memorizzare che ad un'ancora corrisponde un determinato URI Pinata. Questo ha portato al salvataggio di un mapping di stringhe in Solidity per ogni ancora, ed essendo che lavorare con le stringhe richiede una maggiore capacità computazionale, ha portato ad un aumento delle Fee per questa operazione.

Per quanto riguarda il resto delle operazioni, tutte le funzioni sono state lasciate più snelle possibile, proprio per evitare di usare una maggiore capacità di calcolo, ma meno dell'attuale implementazione è sembrato impossibile.

In generale comunque, le "Transaction Fee" risultano essere in linea con le classiche Fee che vengono pagate al giorno d'oggi in una blockchain come Ethereum.

Ricordiamo per completezza che i test sono stati svolti in Sepolia testnet, il che porta a fare considerazioni aggiuntive. Infatti sebbene Sepolia simuli la blockchain Ethereum, sia il costo delle transazioni sia il tempo necessario per il completamento di queste risultano leggermente diverse da quelle che sarebbero in Ethereum mainnet. Problemi di congestione potebbero far aumentare notevolmente sia il costo delle Fee sia le tempistiche, come d'altro canto è visibile anche se in modo ridotto dalle tabelle. Infatti la colonna "Prezzo del Gas" indica il costo del Gas in quel momento e dipende necessariamente dalla congestione della rete.

Per fare un esempio numerico a riguardo, possiamo vedere nel link https://sepolia.etherscan.io/tx/0x513d9416f7533ead37251e7ab36079b3fcca54754f8c8aeea2560480e043fe2d, che per via del grande volume di transazioni che venivano eseguite in quei giorni, l'operazione `createAnchor` è arrivata a costare fino 0.08 Sepolia ETH. Immaginare di sostenere un tale costo con del reale denaro, potrebbe far cambiare idea ad alcuni utenti riguardo la creazione del

proprio NFT. Ed è anche per questo, che sempre più soluzioni di Layer 2 stanno nascendo.

Operazione	**Transaction Fee**	
	Mean	**Std Dev**
`createAnchor`	0.02798532747054334	0.0365043607297913
`approve`	0.0026380429604443605	0.003873485989693527
`listItem`	0.003217987211190908	0.00525398486209166
`cancelListing`	0.0013566410573096147	0.0011263384710036486
`buyItem`	0.0035409119336956385	0.00729602157857204
`redeemItem`	0.006294013952758512	0.01126161825026987

Tabella 4.1: Report delle Transaction Fee per ogni operazione.

Operazione	**Prezzo del Gas**	
	Mean	**Std Dev**
`createAnchor`	71.5788873944	93.37765253376654
`approve`	56.7519396235625	85.16371840237278
`listItem`	40.988768309250005	66.92207087202308
`cancelListing`	42.22612852681818	35.05784583552194
`buyItem`	33.948419054666665	65.73372026558928
`redeemItem`	30.912384520666674	54.30852066782743

Tabella 4.2: Report del prezzo del Gas per ogni operazione.

Conclusioni

Nella presente tesi è stato proposto l'utilizzo dello standard emergente, ERC-6956, per la creazione di una applicazione decentralizzata che potesse collegare il mondo reale con quello digitale.

Questo nuovo standard introduce una struttura di base che amplia le funzionalità di ERC-721, l'attuale standard per NFT, con un approccio basato sulla proprietà di un NFT basato sulla verifica del possesso dell'asset reale. Utilizzando una attestazione ed un oracolo fidato che la firmi, possiamo verificare all'interno della blockchain di essere i reali possessori dell'asset, potendo infine gestire l'NFT.

Abbiamo anche visto come questo meccanismo necessiti di una ancora fisica collegata all'asset reale per la verifica del possesso di quest'ultimo.

In seguito, è stata presentata la fase di progettazione del presente lavoro, partendo da una panoramica sui componenti e su come questi interagiscano tra loro per il corretto funzionamento del sistema.

Basandosi su questa, è stato quindi implementato il sistema, utilizzando tutta una serie di tool che ci hanno permesso di sviluppare gli smart contract relativi ad ERC-6956 e a tutte quelle funzioni che un marketplace deve poter far fare agli utenti.

Inoltre, sono state utilizzate tecnologie di ultima generazione per lo sviluppo, come IPFS e i subgraph. Queste sono due tecnologie in ascesa nel mondo dello sviluppo su blockchain e permettono di creare applicazioni efficienti e decentralizzate.

È stata quindi mostrata la parte grafica dell'applicazione decentralizza-

ta, mostrando come anche qui come lo sviluppo sia stato improntato verso l'efficienza, la sicurezza e l'esperienza utente.

Per finire è stata simulata la parte di verifica di prossimità dell'utente verso l'ancora, e quindi l'asset reale.

L'ultimo passaggio, quello più importante, è stato quello relativo ai test e alla validazione di quanto fatto. Sono stati eseguiti, in un primo momento, test in locale, per appurare che i contratti ed i subgraph si comportassero come richiesto.

Svolta la verifica in locale, si è passati al deploy dei contratti in Sepolia testnet, dove i test sperimentali sono stati finalmente eseguiti.

Nel capitolo sui risultati abbiamo discusso riguardo le prove effettuate. Si è potuto notare come il sistema implementato sia effettivamente funzionante e porti a termine tutte le operazioni dei vari contratti, anche in una testnet.

Inoltre, è stato possibile quantificare i risultati ottenuti, avendo a disposizione tutte le ricevute delle transazioni eseguite per validare il progetto. A tal riguardo è stato possibile notare come le transazioni mediamente utilizzino una quantità di Fee consona a transazioni che vengono eseguite nella blockchain Ethereum. Ma è stato anche possibile notare come la possibilità di incappare in una congestione della rete, possa portare le Fee a salire in costo anche in poco tempo. In generale, le transazioni sono state veloci e contenute nel costo, salvo casi eccezionali, ma è anche da considerare che sono state svolte su una testnet come Sepolia.

Per concludere, è bene avere a mente quali sono stati gli obiettivi prefissati, ovvero verificare praticamente che ERC-6956 è uno standard utilizzabile per collegare il mondo reale con quello digitale, e costruire una applicazione decentralizzata che potesse essere il più trasparente e facile da utilizzare possibile.

Possiamo quindi dire che l'obiettivo del presente lavoro è stato raggiunto con successo, come è stato provato da tutti i vari test eseguiti sia in locale che in testnet.

Come ultima cosa si vuole andare a definire quali possano essere dei pos-

sibili sviluppi futuri per dare una direzione a questo lavoro. Le direzioni più promettenti sembrano essere:

- Miglioramento della tecnologia dell'ancora: Per quanto nel presente lavoro si è cercato di implementare questa tecnologia nel modo più reale possibile, ovviamente questa soluzione basata su Arduino ha dei limiti. Per questo, con l'avvento di future nuove tecnologie per la verifica della prossimità rispetto all'asset reale, si potrebbe rendere ancora più sicuro e facile questo procedimento. Un esempio di progetto che utilizza un particolare chip per fare una sorta di Proof-of-Control è RTFKT [13].

- Deploy su blockchain più efficienti: Per quanto Ethereum sia la blockchain regina per lo sviluppo di smart contract, il costo elevato delle transazioni rispetto ad altre soluzioni è sotto gli occhi di tutti. Anche soluzioni su Layer 2 possono porre rimedio al costo delle Fee, anche se poi in qualche modo il costo per il passaggio tra un Layer e l'altro dovrà essere pagato.

- Possibili nuove funzionalità per ERC-6956: Essendo uno standard relativamente giovane, non possiamo ancora sapere se risolverà i limiti che abbiamo al giorno d'oggi nel campo degli NFT. Potrebbe essere un'idea, quella di pensare a nuove funzionalità.

Bibliografia

[1] Rosenfeld, M. (2012). Overview of colored coins. White paper, bitcoil. co. il, 41, 94. URL: https://bitcoil.co.il/BitcoinX.pdf

[2] Fabian Vogelsteller <fabian@ethereum.org>, Vitalik Buterin <vitalik.buterin@ethereum.org>, "ERC-20: Token Standard," Ethereum Improvement Proposals, no. 20, November 2015. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-20.

[3] William Entriken (@fulldecent), Dieter Shirley <dete@axiomzen.co>, Jacob Evans <jacob@dekz.net>, Nastassia Sachs <nastassia.sachs@protonmail.com>, "ERC-721: Non-Fungible Token Standard," Ethereum Improvement Proposals, no. 721, January 2018. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-721.

[4] Thomas Bergmueller (@tbergmueller), Lukas Meyer (@ibex-technology), "ERC-6956: Asset-bound Non-Fungible Tokens [DRAFT]," Ethereum Improvement Proposals, no. 6956, April 2023. [Online serial]. Available: https://eips.ethereum.org/EIPS/eip-6956.

[5] Agarwal, U., Rishiwal, V., Tanwar, S., Chaudhary, R., Sharma, G., Bokoro, P. N., & Sharma, R. (2022). Blockchain technology for secure supply chain management: A comprehensive review. Ieee Access, 10, 85493-85517. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9841565

[6] Alzahrani, N., & Bulusu, N. (2018, June). Block-supply chain: A new anti-counterfeiting supply chain using NFC and blockchain. In Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems (pp. 30-35). URL: https://www.researchgate.net/profile/Nirupama-Bulusu/publication/325435613_Block-Supply_Chain_A_New_Anti-Counterfeiting_Supply_Chain_Using_NFC_and_Blockchain/links/5b57b0ae458515c4b24358d3/Block-Supply-Chain-A-New-Anti-Counterfeiting-Supply-Chain-Using-NFC-and-Blockchain.pdf

[7] M. C. Jayaprasanna, V. A. Soundharya, M. Suhana, and S. Sujatha, "A block chain based management system for detecting counterfeit product in supply chain," in Proc. 3rd Int. Conf. Intell. Commun. Technol. Virtual Mobile Netw. (ICICV), Feb. 2021, pp. 253-257.

[8] Kumar, R., & Tripathi, R. (2019, January). Traceability of counterfeit medicine supply chain through Blockchain. In 2019 11th international conference on communication systems & networks (COMSNETS) (pp. 568-570). IEEE.

[9] Sahu, B., & Chandramohan Jha, A. M. (2023). NFT Marketplaces: The Future of Digital Asset Trading.

[10] Sharma, N., & Sharma, S. (2022). A survey of Mythril, a smart contract security analysis tool for EVM bytecode. Indian J Natural Sci, 13, 75. URL: https://www.researchgate.net/profile/Swati-Sharma-171/publication/366391033_A_Survey_of_Mythril_A_Smart_Contract_Security_Analysis_Tool_for_EVM_Bytecode/links/639ecbdc095a6a77743c8073/A-Survey-of-Mythril-A-Smart-Contract-Security-Analysis-Tool-for-EVM-Bytecode.pdf

[11] Tantikul, P., & Ngamsuriyaroj, S. (2020, February). Exploring Vulnerabilities in Solidity Smart Contract. In ICISSP (pp. 317-324). URL:

https://pdfs.semanticscholar.org/1f61/143148ac4c0bc62b59ab3410333702d407f6.pdf

[12] Boson Protocol. "Boson Protocol." Available: https://www.bosonprotocol.io/.

[13] RFTKT, RFTKT WM Chip, URL: https://rtfkt.com/faq/rtfkt-wm-chip

[14] Aurambout, J. P., Gkoumas, K., & Ciuffo, B. (2019). Last mile delivery by drones: An estimation of viable market potential and access to citizens across European cities. European Transport Research Review, 11(1), 1-21. URL: https://link.springer.com/content/pdf/10.1186/s12544-019-0368-2.pdf

[15] Garg, V., Niranjan, S., Prybutok, V., Pohlen, T., & Gligor, D. (2023). Drones in last-mile delivery: A systematic review on Efficiency, Accessibility, and Sustainability. Transportation Research Part D: Transport and Environment, 123, 103831. URL: https://www.morecambebaydrones.com/wp-content/uploads/2023/09/2023-Drones-in-last-mile-delivery-A-systematic-review-on-Efficiency-Accessibility-and-Sustainability.pdf

Ringraziamenti

Desidero esprimere la mia gratitudine a tutti coloro che mi hanno supportato in questo percorso universitario.

In primo luogo, un sentito ringraziamento al mio relatore, Prof. Stefano Ferretti, per avermi guidato nello sviluppo della tesi, e per avermi dato spunti preziosi che hanno arricchito il lavoro svolto, portandolo ad essere quello che è oggi. La sua disponibilità e il suo supporto costante sono stati fondamentali per superare le sfide che ho incontrato lungo il percorso.

Grazie ai miei compagni di studi, in particolar modo ad Andrea e Luca, con cui ho condiviso esperienze che mi porterò dietro per tutta la vita.

Grazie ai miei amici, capaci anche involontariamente di farmi rilassare nei momenti di maggiore difficoltà. Con loro ho passato momenti unici e indimenticabili, e se sono qui oggi è anche grazie a tutti voi. Spesso non dimostro ciò che penso, ma vorrei dirvi che vi voglio bene. La vostra presenza e il vostro supporto hanno reso questo viaggio molto più bello.

Per finire, Un ringraziamento speciale non poteva che andare alla mia famiglia, che mi ha sempre sostenuto e incoraggiato nei momenti difficili, non facendomi mancare mai nulla. Senza la loro fiducia e il loro supporto, non sarei riuscito a raggiungere questo grande traguardo. Siete una fonte di ispirazione per me.

Spero, almeno in parte, di aver reso fiero tutti voi oggi.