

Sommario

Introduzione

Requisiti.....

User Interface navigation and mock-up

Architettura

Appendice A – Termini e definizioni.....

Appendice B – Documenti di riferimento.....

Introduzione

Lottery DApp è un'applicazione decentralizzata il cui aspetto fondamentale è lo Smart Contract che viene distribuito sulla blockchain di Ethereum ed è responsabile della costruzione delle lotterie decentralizzate.

Il titolare del contratto (Owner) è il solo che può attivare la lotteria.

In fase di inizializzazione, il titolare può stabilire alcuni parametri come la durata della lotteria e il costo del biglietto mentre dati come il valore della commissione che riceverà una volta chiusa la lotteria sono impostati a livello di contratto.

Con l'apertura della lotteria si va ad attivare la finestra di gioco che consente l'acquisto dei biglietti.

I giocatori ("Giocatore1", "Giocatore2", ecc.) acquistano i biglietti della lotteria e le loro probabilità di vincita alla lotteria sono direttamente correlate alla proporzione del totale dei biglietti in sospeso che detengono per quella lotteria.

L'acquisto può avvenire solo dopo aver effettuato l'accesso sul proprio Wallet.

I giocatori nella loro area utente possono consultare i biglietti acquistati e i biglietti vinti.

Solo dopo la chiusura della finestra di gioco, può verificarsi un'estrazione.

La finestra di gioco viene chiusa con il raggiungimento di un timestamp.

L'estrazione e la chiusura effettiva della lotteria possono essere eseguite solo dal titolare.

Requisito Funzionale

Overview

Un requisito funzionale è un'affermazione di come un sistema deve comportarsi. Definisce cosa deve fare il sistema per soddisfare le esigenze o le aspettative dell'utente

Scenarios

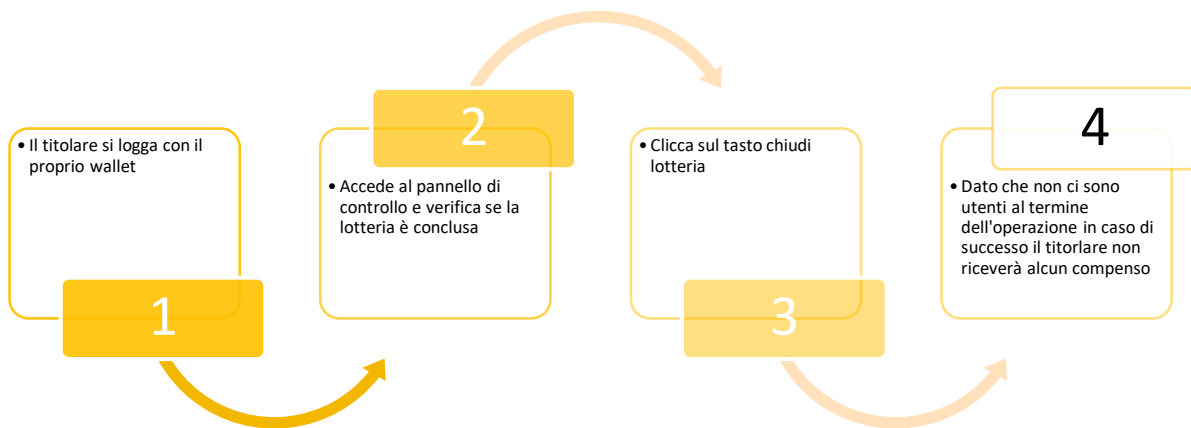
Primo Scenario Acquisto di un biglietto da parte di un giocatore



Scenario chiusura di una lotteria



Scenario Alternativo - chiusura di una lotteria senza utenti



Scenario – Apertura di una lotteria



Use Case

Nome	Attori	Pre-condition	Flusso degli eventi	Post-condition
Accesso al Wallet	User (Giocatore e Owner)	L'user deve essere in possesso di un wallet Metamask e un browser abilitato all'utilizzo del suddetto wallet	<p>Flusso degli eventi:</p> <ol style="list-style-type: none">1. L'User accede al sito2. L'User clicca sul tasto Wallet che funge da login3. Nel caso in cui è in possesso di un browser autorizzato all'utilizzo di un wallet metamask può loggarsi con le sue credenziali <p>Flusso alternativo:</p> <ol style="list-style-type: none">1. L'User non è in possesso di un wallet2. L'User si registra a un wallet e abilita l'estensione relativa al browser3. L'User può effettuare l'accesso mediante il tasto Wallet che funge da login	il giocatore e l'Owner si sono loggati mediante il loro account e possono visualizzare le proprie informazioni utente

Nome	Attori	Pre-condition	Flusso degli eventi	Post-condition
Acquista Ticket	Giocatore	<p>il giocatore deve aver effettuato l'accesso al proprio wallet per poter visualizzare il tasto acquisto</p> <p>deve possedere un saldo maggiore di 0 ETH</p> <p>la finestra di gioco deve essere aperta</p>	<p>Flow of events:</p> <ol style="list-style-type: none">1. il giocatore clicca sul tasto Compra2. viene inviata la richiesta al servizio3. a fine elaborazione sarà mostrato a schermo l'esito dell'operazione	<p>Nel caso di successo:</p> <p>mediante una modale verrà notificato al giocatore l'esito positivo dell'acquisto, sarà incrementato il contatore relativo al numero di biglietti da lui acquistati</p> <p>Nel caso di errore:</p> <p>mediante una modale verrà notificato al giocatore il fallimento avvenuto in fase di acquisto</p>
Visualizza ticket acquistati	Giocatore	il giocatore deve aver effettuato l'accesso al proprio wallet per poter visualizzare il tasto relativo ai ticket	<p>Flow of events:</p> <ol style="list-style-type: none">1. il giocatore clicca l'icona della campanella o l'icona della coccarda2. Si aprirà una modale in cui potrà visualizzare i ticket acquistati o vinti a seconda del filtro scelto	-

Nome	Attori	Pre-condition	Flusso degli eventi	Post-condition
Apri Lotteria	Owner	L'owner deve aver effettuato l'accesso al proprio wallet La precedente lotteria deve essere stata chiusa	Flow of events: 1. L'owner clicca sull'icona dell'ingranaggio per accedere al pannello di controllo 2. Si apre una finestra relativa al pannello di controllo 3. L'owner può decidere la durata della lotteria in un range di (5 – 60 minuti) e il costo dei biglietti in un range (1-10 ETH) 4. L'owner clicca su attiva lotteria 5. Paga la commissione relativa all'avvio 6. A transazione conclusa gli sarà notificato l'esito	successo: L'owner avvia una nuova lotteria Errore: gli sarà notificato un messaggio di errore relativo al fallimento di apertura di una nuova lotteria

Nome	Attori	Pre-condition	Flusso degli eventi	Post-condition
Chiudi Lotteria	Owner	L'owner deve aver effettuato l'accesso al proprio wallet Il timestamp della lotteria deve essere scaduto	Flow of events: 1. L'owner clicca sull'icona dell'ingranaggio per accedere al pannello di controllo 2. Si apre una finestra relativa al pannello di controllo 3. L'owner clicca su chiudi lotteria 4. Viene effettuata una richiesta di chiusura allo smart Contract 5. A chiusura effettuata viene notificato l'esito dell'operazione Flusso alternativo: 1. In caso di assenza di partecipanti l'owner potrà chiudere la lotteria ma non riceverà alcun compenso	postconditions: Nel caso di successo e in presenza di partecipanti: sarà estratto un vincitore e l'owner riceverà la sua commissione

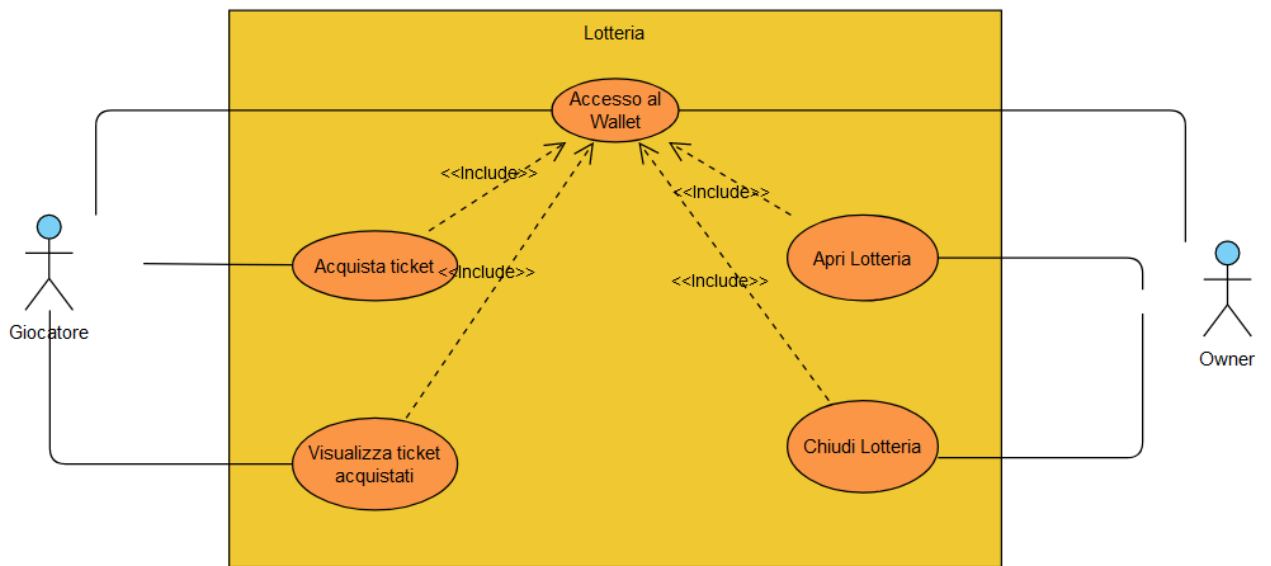


Figure 1-Use Case Lotteria

Class Diagram e Sequence Diagram

Il **diagramma delle classi** rappresenta gli elementi del sistema, le entità logiche, (dette anche **classi**) e le eventuali relazioni fra di loro. Ogni classe è corredata da un insieme di **attributi** (che ne descrivono le caratteristiche) e **operazioni** (che ne descrivono il comportamento della classe).

I **sequence diagram** sono diagrammi di **interazione** che descrivono in dettaglio come vengono eseguite le operazioni. Catturano l'interazione tra oggetti nel contesto di una collaborazione.

I diagrammi di sequenza sono incentrati sul **tempo** e mostrano visivamente l'ordine dell'interazione utilizzando l'asse verticale del diagramma per rappresentare l'ora in cui i messaggi vengono inviati e quando.

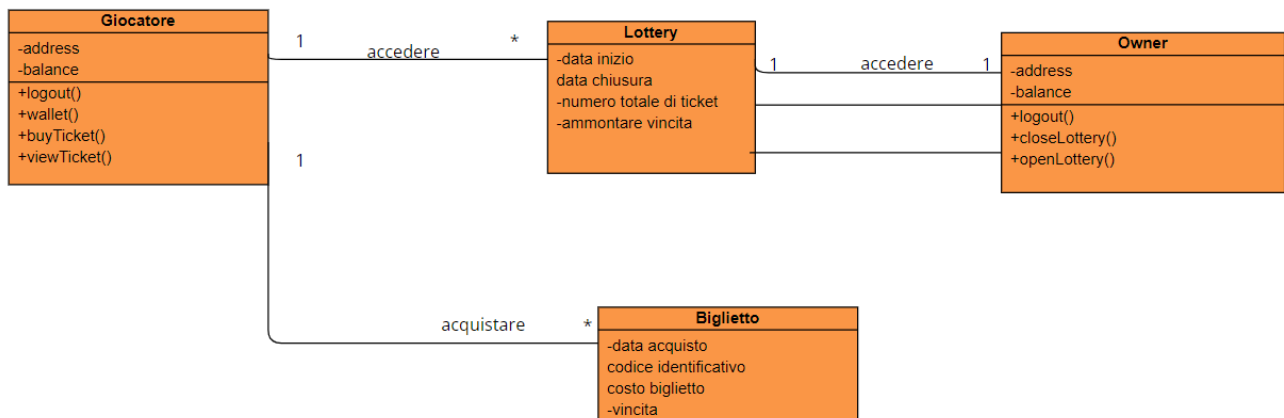


Figure 2 Class Diagram Lotteria

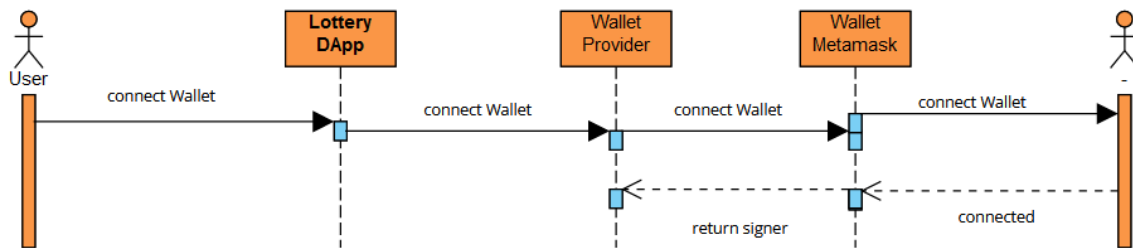


Figure 3-Connessione Wallet

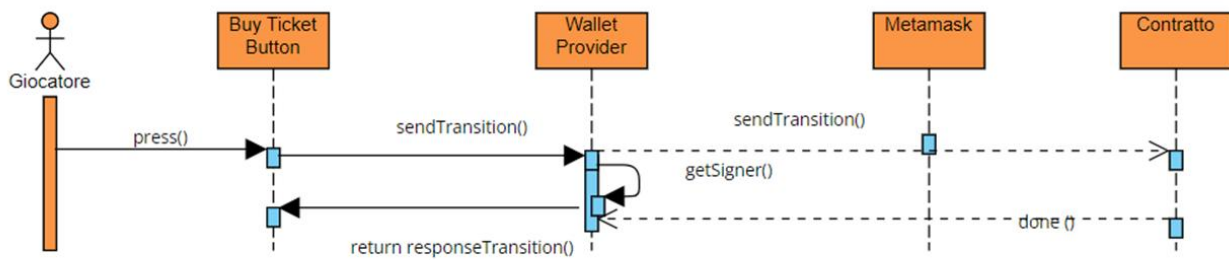


Figure 4-Acquisto Biglietto

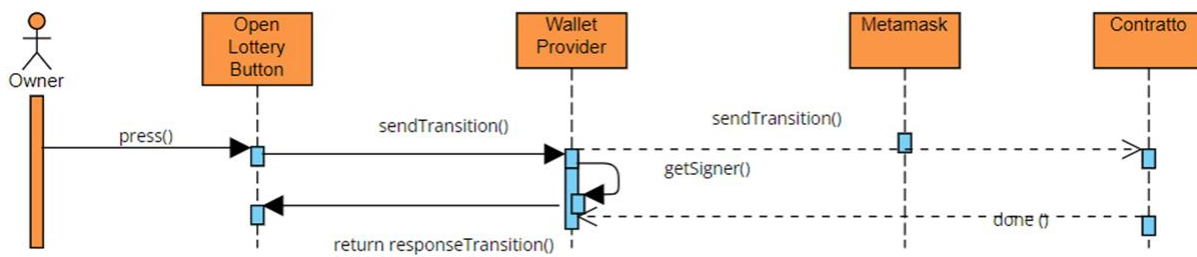


Figure 5-Apertura Lotteria

User Interface navigation and mock-up

UI/UX Mockup presenta l'interfaccia utente e l'esperienza utente di un sito Web o di un'applicazione.

Il mockup è una parte essenziale del processo di progettazione.

Mock-up dell'applicazione

Vista globale dell'applicazione

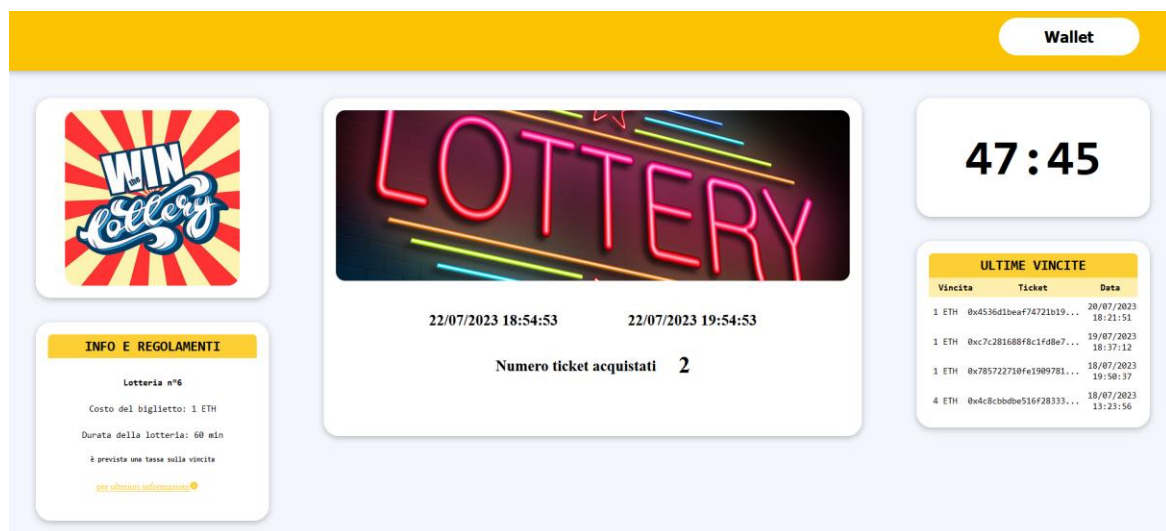


Figure 1- Lotteria attiva

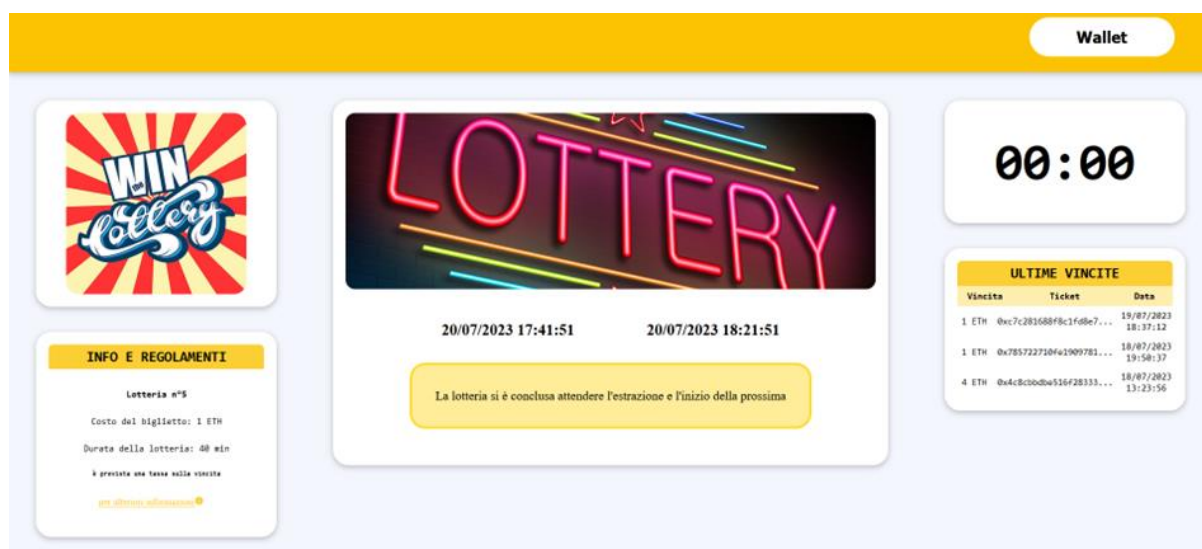


Figure 2- Lotteria chiusa

Count down

Countdown che indica la durata della lotteria. Quando la lotteria termina il countdown viene settato a 0 .

58:29

00:00

Elenco ultimi vincitori

- Valore della vincita in ETH
- Il codice associato al ticket
- Data di chiusura della lotteria

ULTIME VINCITE		
Vincita	Ticket	Data
1 ETH	0x4536d1beaf74721b19...	20/07/2023 18:21:51
1 ETH	0xc7c281688f8c1fd8e7...	19/07/2023 18:37:12
1 ETH	0x785722710fe1909781...	18/07/2023 19:50:37
4 ETH	0x4c8cbbdbe516f28333...	18/07/2023 13:23:56

Acquisto

Nel caso dei giocatori una volta loggati se la lotteria è ancora attiva si ha la comparsa del tasto “Compra “



22/07/2023 18:54:53

22/07/2023 19:54:53

Numero ticket acquistati 0

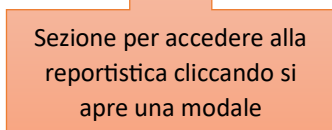
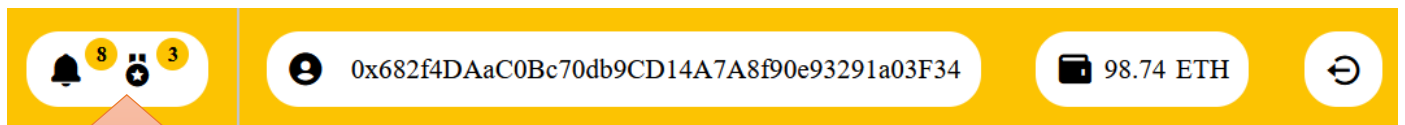
Compra

Header

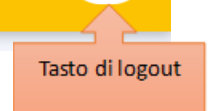
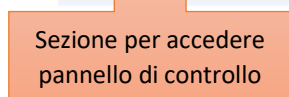
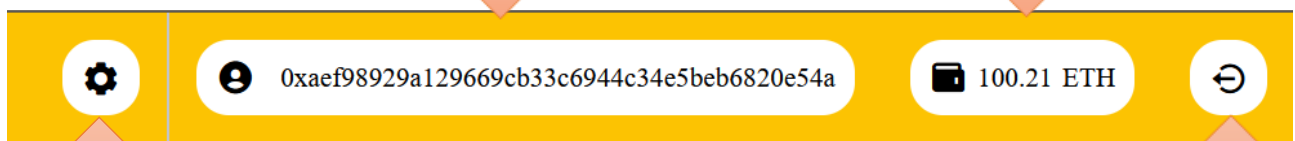
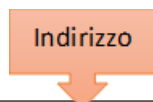
- Vista dell'header da non loggato presenta solo il tasto per loggarsi sul wallet e il logo della Dapp



- Vista dell'header da Giocatore
 - o La sezione per accedere alla propria reportistica
 - o L'indirizzo e il saldo relativi al Wallet
 - o Il tasto di logout



- Vista dell'header da Titolare
 - o Indirizzo e il Saldo relativi all Wallet
 - o Il tasto di logout
 - o La sezione per accedere al pannello di controllo



Report Biglietti Acquistati dal Giocatore

Il giocatore una volta loggatosi può visualizzare il riepilogo relativo ai biglietti cliccando nell'apposita sezione



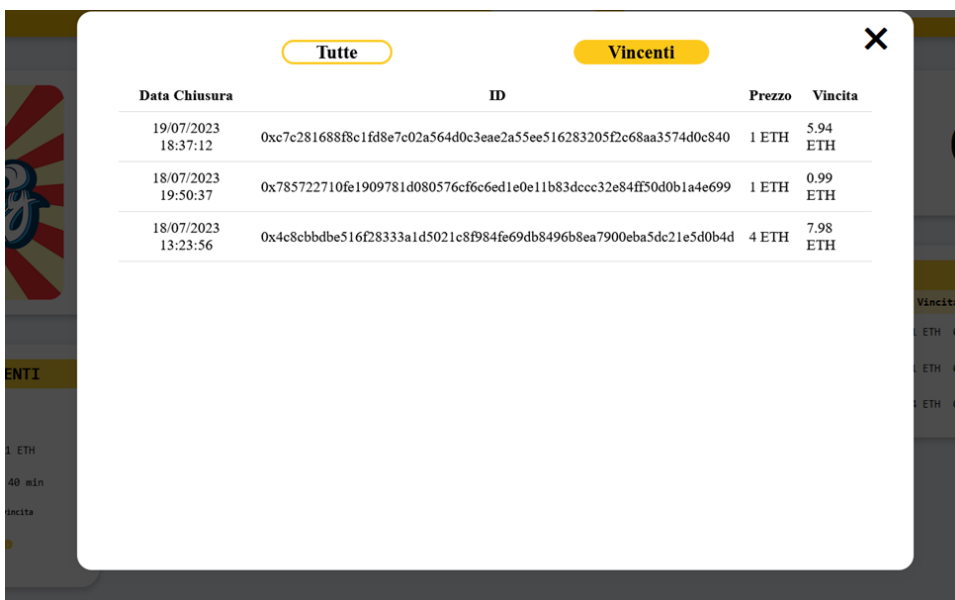
Cliccando sul simbolo della campanella si aprirà una modale con al suo interno una tabella che conterrà il riepilogo di tutti i ticket acquistati dall'utente mentre cliccando sulla coccarda si può andare a visualizzare direttamente solo i biglietti vincenti. Mediante degli appositi tab si può switchare tra le due modalità di visualizzazione

Tutte

Vincenti

X

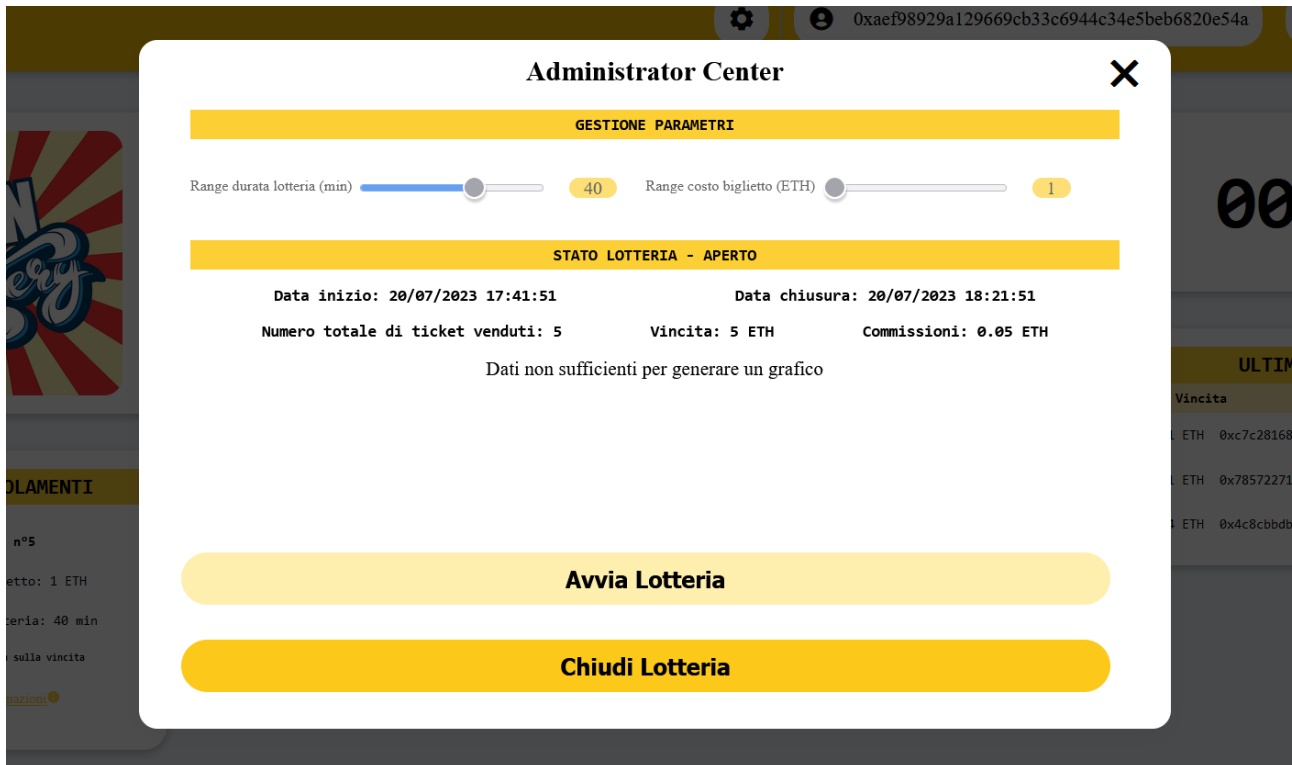
Data Chiusura Data Apertura	ID	Prezzo
20/07/2023 18:21:51 20/07/2023 18:09:53	0xe5068281292f8d7e8a2632381f82fec14f6e771d2ba3e9bb242e295e3f5b7d3e	1 ETH
19/07/2023 18:37:12 19/07/2023 18:07:41	0xa3ceaea04487226419bef82e36d088c1e9286a30b44f91f71c423324ed733e09	1 ETH
19/07/2023 18:37:12 19/07/2023 18:07:55	0xc7c281688f8c1fd8e7c02a564d0c3eae2a55ee516283205f2c68aa3574d0c840	1 ETH
19/07/2023 18:37:12 19/07/2023 18:09:02	0xde869c6e8c2e851c4e47638259e1bcd880668b468afa4d077a66e3765af09f23	1 ETH
18/07/2023 19:50:37 18/07/2023 19:22:10	0x785722710fe1909781d080576cf6c6ed1e0e11b83dccc32e84ff50d0b1a4e699	1 ETH
18/07/2023 13:23:56 18/07/2023 13:20:12	0xa745a8731848a7ab2b6f78f26c39b17087becf4872d20df54976fc46a884f844	4 ETH
18/07/2023 13:23:56 18/07/2023 13:21:59	0x4c8cbbdbe516f28333a1d5021c8f984fe69db8496b8ea7900eba5dc21e5d0b4d	4 ETH



Tutte		Vincenti			
Data Chiusura		ID	Prezzo	Vincita	
19/07/2023 18:37:12	0xc7c281688f8c1fd8e7c02a564d0c3eae2a55ee516283205f2c68aa3574d0c840	1 ETH	5.94 ETH		
18/07/2023 19:50:37	0x785722710fe1909781d080576cf6c6ed1e0e11b83dccc32e84ff50d0b1a4e699	1 ETH	0.99 ETH		
18/07/2023 13:23:56	0x4c8cbbdbe516f28333a1d5021c8f984fe69db8496b8ea7900eba5dc21e5d0b4d	4 ETH	7.98 ETH		

Pannello di Controllo del titolare

Il titolare cliccando sul simbolo della “rotellina” può accedere al pannello di controllo per la gestione della lotteria.



Attraverso il pannello di controllo il titolare può gestire e monitorare la lotteria.

1. Il titolare può visualizzare alcune informazioni come:
 - a. lo stato della lotteria
 - b. il numero totale di biglietti venduti
 - c. il valore della vincita e della commissione ¹
 - d. la data di apertura e chiusura della lotteria
 - e. Nel caso di un numero di biglietti venduti maggiore di 20 un grafico riassuntivo
2. Il titolare può effettuare delle operazioni come:
 - a. decidere la durata della lotteria (in un range che va dai 5 ai 60 minuti) e il costo del biglietto (in un range 1 a 10 ETH)²
 - b. Allo scadere della lotteria terminarla mediante il tasto “chiudi lotteria” che porterà all'estrazione di un vincitore e successivamente può decidere di aprire una nuova lotteria

¹ La commissione sulla vincita massima viene calcolata num di ticket * parametro della commissione (valore fisso dello smart contract)

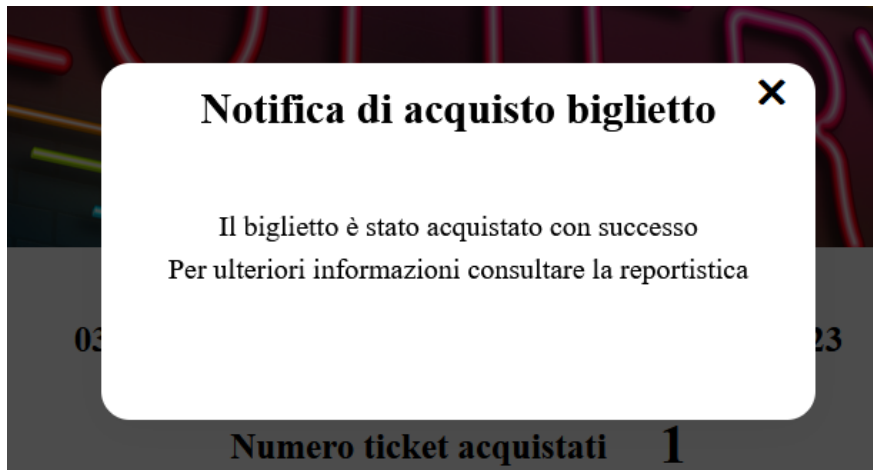
² Nel caso in cui l'utente non effettua una scelta verranno usati i valori della lotteria precedente

Messaggistica

Mediante modali viene implementata una messaggistica che informa l'utente di determinati avvenimenti

Alcuni esempi sono:

- la modale post acquisto



- la modale che informa il titolare dell'esito della chiusura della lotteria



Architettura

Overview

L'architettura di un'applicazione decentralizzata o DApp si discosta da quella delle applicazioni tradizionali dato che non esiste un database centralizzato che memorizza lo stato dell'applicazione.

A differenza delle applicazioni tradizionali, l'applicazione lato client non comunica con il database ma comunica direttamente con la blockchain, la quale consente di elaborare i dati attraverso reti distribuite ed eseguire transazioni.

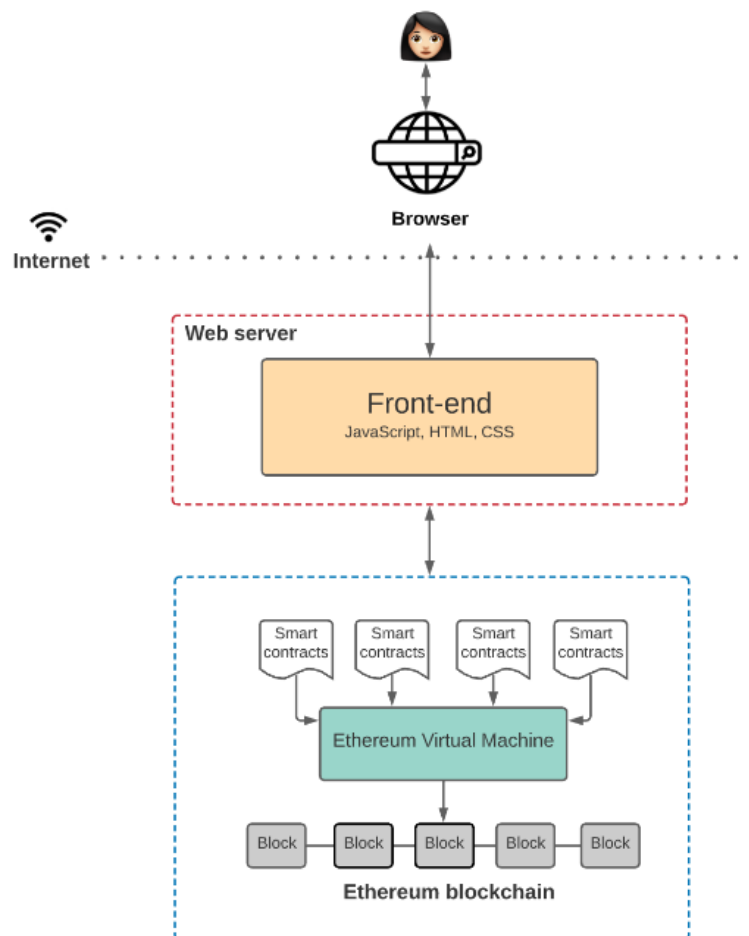


Figure 1 Struttura di base di un'applicazione decentralizzata

Struttura di un applicazione decentralizzata

L'architettura di una Dapp è costruita su una rete decentralizzata che combina uno smart contract e l'interfaccia utente di un front-end.

La rete decentralizzata è realizzata dalla blockchain di Ethereum che stabilisce una rete peer-to-peer che esegue e verifica in modo sicuro il codice dell'applicazione, chiamato contratti intelligenti.

Le applicazioni decentralizzate hanno il loro codice back-end in esecuzione su una rete decentralizzata e non su un server centralizzato. La blockchain di Ethereum viene utilizzata anche per l'archiviazione dei dati.

Gli smart contract invece si occupano di definire i cambiamenti di stato che si verificano sulla blockchain e rappresentano il codice back-end

Come qualsiasi tipo di contratto, i contratti intelligenti specificano le condizioni di un accordo tra parti diverse. Si tratta di una raccolta di codice e dati che risiede in un indirizzo specifico sulla blockchain di Ethereum e viene eseguito sulla blockchain di Ethereum

Una volta distribuiti i contratti intelligenti sulla rete, non è possibile modificarli.

Le DApp possono essere decentralizzate perché sono controllate dalla logica scritta nel contratto, non da un individuo o da un'azienda.

Tra la blockchain di Ethereum e gli Smart contract è necessario un intermediario.

Questo ruolo viene ricoperto dalla Ethereum Virtual Machine (EVM).

Un EVM è un ambiente di runtime che esegue gli smart contract di Ethereum.

Viene progettato come una sandbox, ciò lo rende isolato dalle altre parti del sistema.

Nell'ecosistema Ethereum, EVM svolge un ruolo fondamentale fornendo una piattaforma per applicazioni decentralizzate (DApp) da costruire su di esso.

Ciò garantisce che tutte le transazioni e i contratti intelligenti effettuati sulla blockchain di Ethereum siano eseguiti nel modo corretto e previsto, come desiderato dal codice del contratto intelligente. Serve come piattaforma per l'esecuzione delle applicazioni.

In fine per poter interagire con l'utente è necessaria un GUI. Nel caso delle DApp l'interfaccia grafica non si occuperà solo di visualizzare a schermo le informazioni ma anche di comunicare con la logica dell'applicazione definita negli smart contract.

Per richiamare le funzioni, il front-end deve comunicare con contratti intelligenti.

Ma Ethereum è una rete decentralizzata. Ogni nodo della rete Ethereum conserva una copia di tutti gli stati sulla macchina virtuale Ethereum, inclusi i dati e il codice associati a ogni contratto intelligente.

È necessario quindi interagire con uno di questi nodi per interagire con i dati e il codice sulla blockchain.

Questo perché qualsiasi nodo può trasmettere una richiesta per l'esecuzione di una transazione sull'EVM. Quindi il compito di un minatore è eseguire la transazione e propagare il cambiamento di stato risultante alla rete.

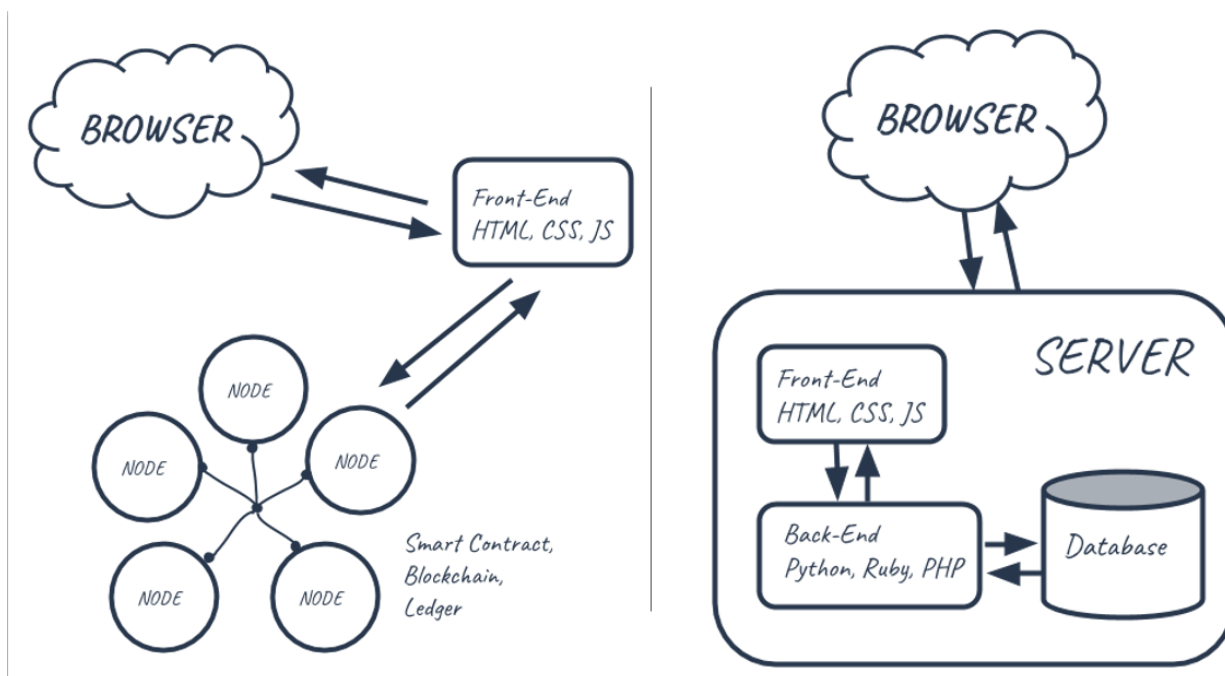


Figure 2 Confronto architettura applicazione centralizzata con applicazione decentralizzata

Autenticazione con Metamask

Blockchain o Web3.0 forniscono un accesso decentralizzato, il che significa che nessuno ha accesso alle informazioni personali dell'utente, mitigando quindi il rischio della privacy dell'utente e rendendo gli utenti proprietari delle proprie informazioni.

Per autenticare gli utenti su portafogli Blockchain come Metamask, vengono utilizzati metodi come la crittografia a chiave pubblica, rendendola così una piattaforma sicura e protetta per l'accesso ai servizi su Internet.

Il ruolo di Metamask nelle applicazioni decentralizzare

MetaMask è un'estensione o un plug-in per i browser che consente agli utenti di interagire facilmente con le applicazioni decentralizzate sulla blockchain di Ethereum. Questo è possibile perché MetaMask funge da ponte tra le DApp e i browser, facilitandone l'utilizzo e la fruizione, funziona grazie all'utilizzo di web3.js, una libreria che fa parte dello sviluppo ufficiale di Ethereum.

MetaMask è stato creato per essere un wallet per Ethereum e uno strumento per interagire con le DApps. Per cui funge sia da provider che da signer.

Come provider stabilisce un canale di comunicazione tra l'estensione e l'applicazione. Una volta che l'applicazione riconosce che MetaMask sia presente, viene abilitato e può essere utilizzato dall'utente.

L'utente può eseguire varie azioni, dall'acquisto o vendita di token, all'accesso alle risorse o a qualsiasi servizio da esse fornito.

Ognuna di queste azioni presenta un costo, che deve essere pagato in Ethereum o nel token ad essa indicato.

In entrambi i casi, MetaMask dispone degli strumenti necessari per gestire tale interazione. Va a ricoprire il ruolo di Signer, non solo genera un wallet di criptovaluta, ma controlla anche ogni interazione dell'utente con l'applicazione ed esegue le operazioni necessarie affinché tali operazioni vengano eseguite.

Tutto questo avviene in un mezzo di comunicazione sicuro e utilizzando una forte crittografia.

MetaMask ha la capacità di generare le proprie chiavi asimmetriche, salvarle localmente e gestirne l'accesso. Grazie a questo, è un'estensione altamente sicura.

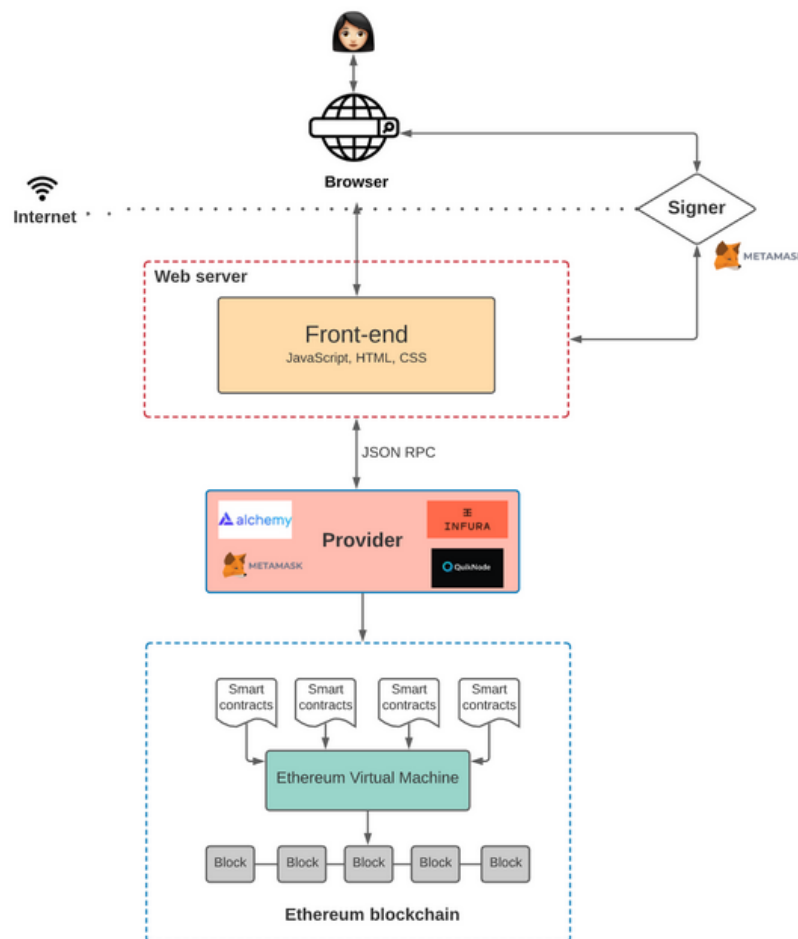


Figure 3 Architettura di un'applicazione decentralizzata con l'inserimento di metamask

Appendice A – Termini e definizioni

Blockchain: la blockchain è un database distribuito o libro mastro condiviso tra i nodi di una rete di computer. Sono meglio conosciuti per il loro ruolo cruciale nei sistemi di criptovaluta per mantenere un registro sicuro e decentralizzato delle transazioni, ma non sono limitati agli usi di criptovaluta. Le blockchain possono essere utilizzate per rendere immutabili i dati in qualsiasi settore, il termine usato per descrivere l'impossibilità di essere modificati.

Poiché non è possibile modificare un blocco, l'unica fiducia necessaria è nel punto in cui un utente o un programma immette i dati. Questo aspetto riduce la necessità di terze parti fidate, che di solito sono revisori o altre persone che aggiungono costi e commettono errori.

Smart contract: Uno smart contract è un programma auto eseguibile che automatizza le azioni richieste in un accordo o contratto. Una volta completate, le transazioni sono tracciabili e irreversibili. I contratti intelligenti consentono di eseguire transazioni e accordi affidabili tra parti disparate e anonime senza la necessità di un'autorità centrale, un sistema legale o un meccanismo di applicazione esterno.

Applicazioni decentralizzate (dApp): sono programmi software che vengono eseguiti su una rete di computer blockchain o peer-to-peer (P2P) invece che su un singolo computer. Le DApp (chiamate anche "dapps") sono quindi al di fuori dell'ambito e del controllo di un'unica autorità. Le DApp sono spesso costruite sulla piattaforma Ethereum.

Ethereum: Piattaforma decentralizzata per la creazione e pubblicazione peer to peer di contratti intelligenti creati in un linguaggio di programmazione Turing-completo. La criptovaluta è Ether.

Appendice B- documenti di riferimento

Sitografia

<https://www.geeksforgeeks.org/architecture-of-a-dapp/>

<https://ethereum.org/it/developers/docs/dapps/>

<https://www.coinbase.com/it/learn/crypto-basics/what-is-a-smart-contract>

<https://www.investopedia.com>

<https://docs.metamask.io/>