

Chain of Art

Corso di Sicurezza dei Dati

Anno accademico 2023-2024

Docenti

Alfredo De Santis
Christiancarmine Esposito



Studenti

Angela De Martino 0522501589
Marco Ciano 0522501674

La discussione verterà su

1. Introduzione

2. Realizzazione

3. Strumenti per
lo sviluppo

4. Conclusioni





Introduzione



Chain of Art

Chain of Art è una piattaforma decentralizzata basata sulla tecnologia blockchain. È progettata per gli appassionati d'arte e i collezionisti, offrendo loro un modo innovativo di acquistare e collezionare opere d'arte digitali senza intermediari.

La nostra DApp semplifica il processo di acquisto di opere d'arte, consentendo agli utenti di connettersi direttamente con gli artisti. Grazie alla tecnologia blockchain, possiamo garantire l'autenticità e la trasparenza di ogni opera d'arte presente sulla piattaforma.

Ogni transazione e ogni dettaglio sull'opera sono registrati in modo sicuro sulla blockchain, eliminando il rischio di contraffazione.

Chain of Art offre un ambiente in cui gli amanti dell'arte possono esplorare, acquistare e collezionare creazioni artistiche contemporanee, rendendo l'arte digitale più accessibile e sicura per tutti.

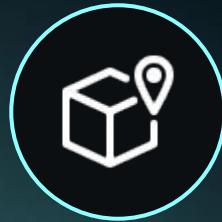


Vantaggi



Decentralizzazione

Non esiste un'autorità centrale che controlla il processo di acquisto di un'opera d'arte digitale e questo rappresenta un'enorme vantaggio in termini di trasparenza.



Tracciabilità

L'operazione di acquisto di un'opera d'arte digitale, inclusi i suoi dettagli, sono pubblici e verificabili.



Immutabilità

L'acquisto confermato e registrato nel blocco diventa irrevocabile una volta incluso nel registro delle transazioni, e non può essere modificato retroattivamente.





Realizzazione



Requisiti Funzionali

RF_1

Visualizzazione delle opere d'arte digitali

Tutti gli utenti potranno visualizzare l'elenco completo di tutte le opere d'arte digitali compresi i loro dettagli come nome dell'opera, artista, prezzo del quadro (in ETH).

Priorità Alta

RF_2

Acquisto di un opera d'arte digitale

Gli utenti avranno la possibilità di acquistare l'opera d'arte che preferiscono.

Priorità Alta

RF_3

Visualizzazione delle opere d'arte acquistate

Gli utenti devono poter accedere ad un elenco delle opere d'arte che hanno acquistato.

Priorità Alta

RF_4

Autenticazione degli utenti

Prima di poter procedere all'acquisto di un'opera d'arte digitale, gli utenti devono poter autenticarsi attraverso il loro wallet digitale Metamask, garantendo così l'immutabilità delle transazioni legate a delle identità verificate.

Priorità Alta

RF_5

Aggiornamento dello stato dell'opera d'arte

Il sistema deve aggiornare lo stato del quadro da "Disponibile" ad "Acquistato" dopo l'avvenuto acquisto da parte di un utente.

Priorità Media



Mock-up

The image shows a wireframe representation of a web browser window. The title bar at the top reads "Chain Of Art". Below the title bar is a standard browser header with back, forward, stop, and search buttons, along with a URL field containing "https://". The main content area is titled "Lista Quadri" (List of Paintings). It displays a 2x4 grid of eight identical placeholder cards, each representing a painting. Each card features a large square icon with a diagonal cross and the word "Quadro" in a small box. Below the icon, there is a summary of the painting's details: "Titolo: [REDACTED]", "Artista: [REDACTED]", and "Prezzo: [REDACTED]". At the bottom of each card is a small rectangular button labeled "Acquista" (Buy). The entire grid is centered on the page.

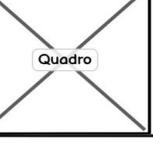


Mock-up

Chain Of Art

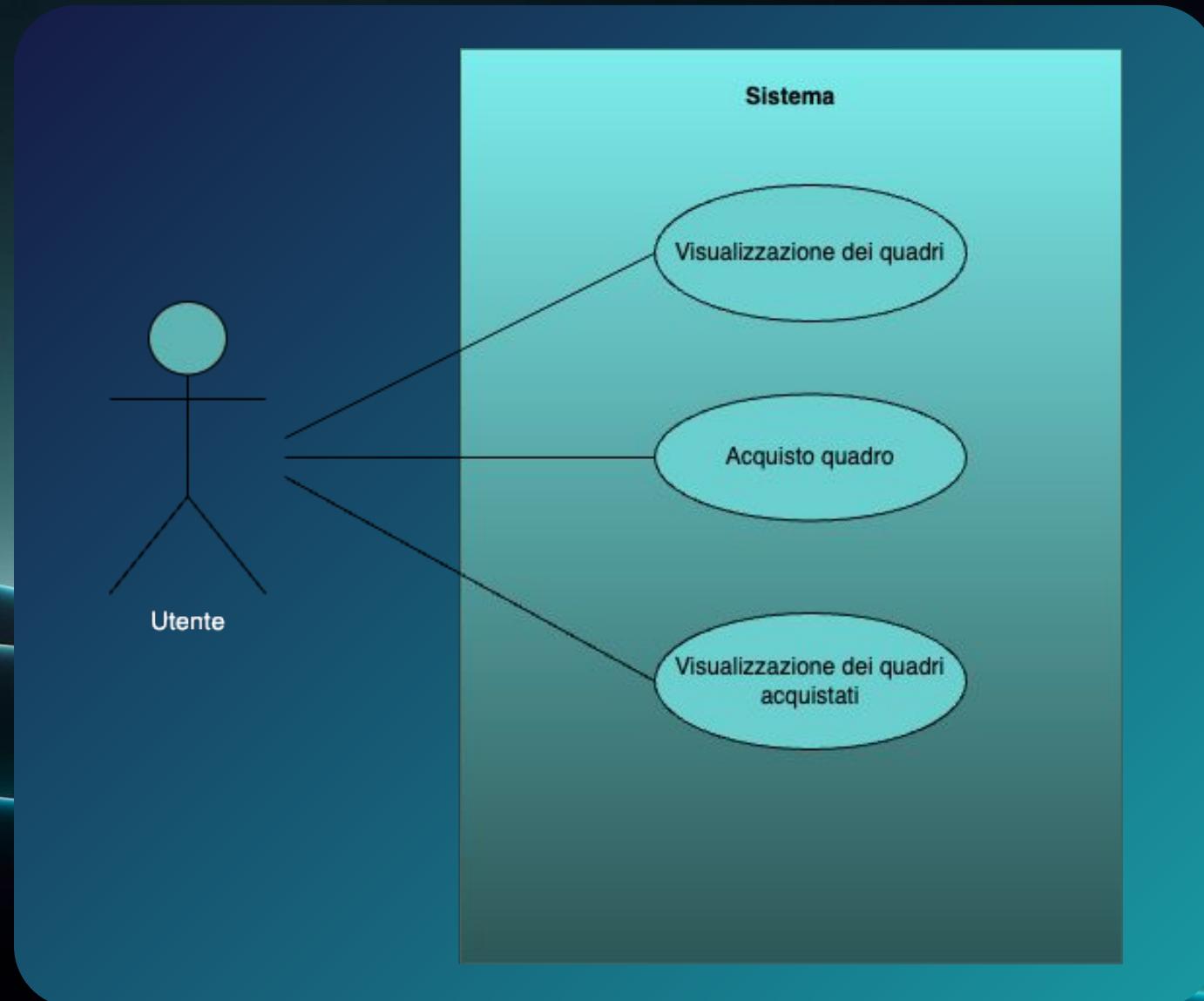
https://

Quadri Acquistati

ID	Quadro	Titolo	Artista	Prezzo	Stato
mmmm	 Quadro	mmmm mmmm mmmm	mmmm mmmm mmmm		ACQUISTATO
mmmm	 Quadro	mmmm mmmm mmmm	mmmm mmmm mmmm		ACQUISTATO



Use Case Model



Use Cases

UC_01: Acquisto di un'opera d'arte digitale

Attore: Utente

Entry condition: L'utente si trova nella homepage della dApp e si è autenticato al suo wallet digitale Metamask.

Flusso degli eventi:

1. L'utente naviga attraverso l'elenco delle opere d'arte digitali disponibili sulla dApp. Ogni quadro presenta le seguenti informazioni: Immagine del quadro, titolo del quadro, nome e cognome dell'artista e il prezzo.
2. L'utente seleziona il quadro che desidera comprare.
3. L'utente clicca sul pulsante “Acquista” posizionato sotto i dettagli del quadro.
4. Il sistema controlla che l'utente non abbia già acquistato quel quadro.
5. Il sistema mostra una schermata per confermare la transazione.
6. L'utente autorizza la transazione

Exit condition (On Success): Il sistema completa l'acquisto del dipinto e informa l'utente del successo della transazione.

Exit condition (On failure): Il sistema notifica l'utente di un errore nella transazione.

Flussi alternativi:

- Se nel punto 4 il sistema nota che l'utente ha già acquistato il quadro, la transazione viene annullata e viene mostrato all'utente il seguente messaggio di errore “Errore nella transazione”



Use Cases

UC_02: Visualizzazione delle opere d'arte acquistate

Attore: Utente

Entry condition: L'utente si trova nella homepage della dApp e si è autenticato al suo wallet digitale Metamask ed ha effettuato l'acquisto di almeno un quadro.

Flusso degli eventi:

1. L'utente naviga nella homepage verso la sezione “Quadri Acquistati”.
2. L'utente visualizza una tabella riportante i quadri acquistati con i rispettivi dettagli per ogni quadro.

Exit condition (On Success): Il sistema mostra all'utente i quadri acquistati con i relativi dettagli.

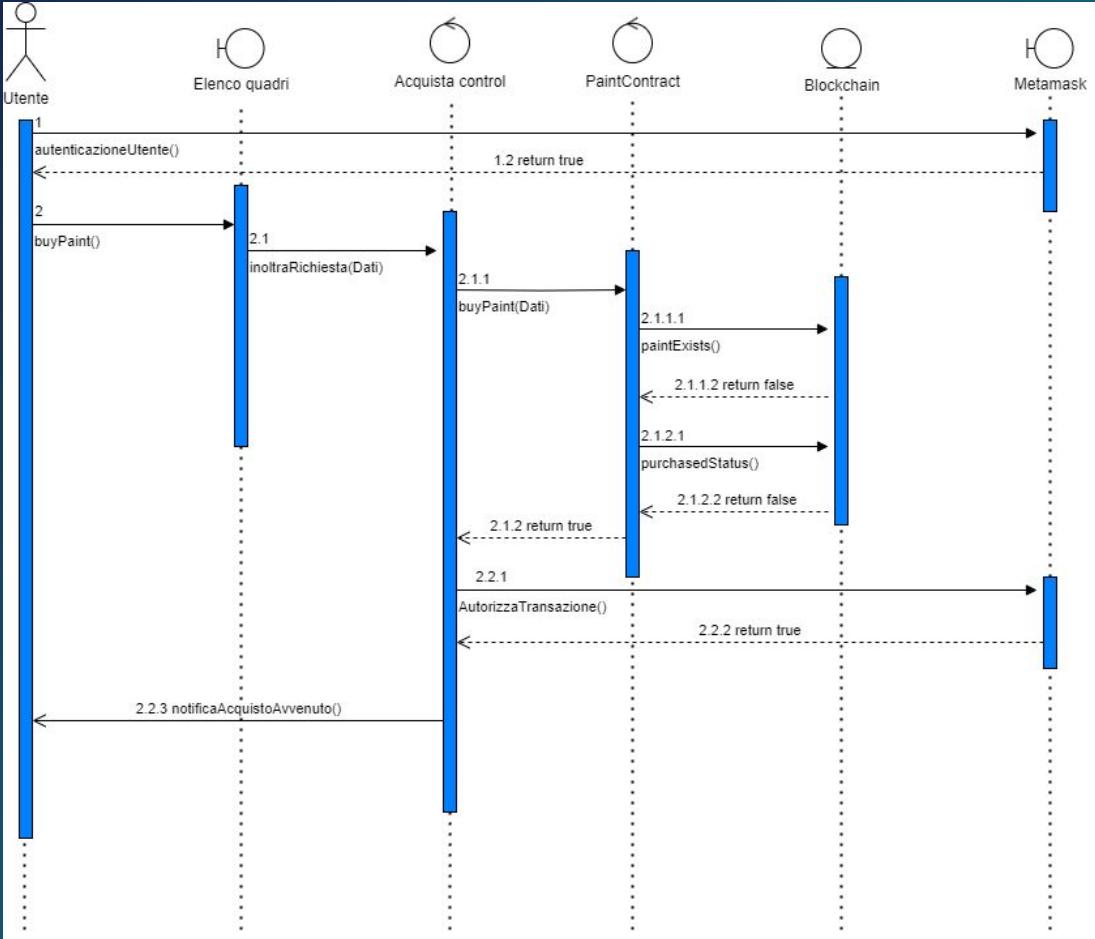
Flussi alternativi:

- Se nel punto 2 l'utente non ha effettuato l'acquisto almeno un quadro, il sistema mostra un'immagine con label di “no data” seguita dal messaggio “Non hai acquistato alcun quadro”.



Sequence diagram

Acquisto di un'opera d'arte digitale

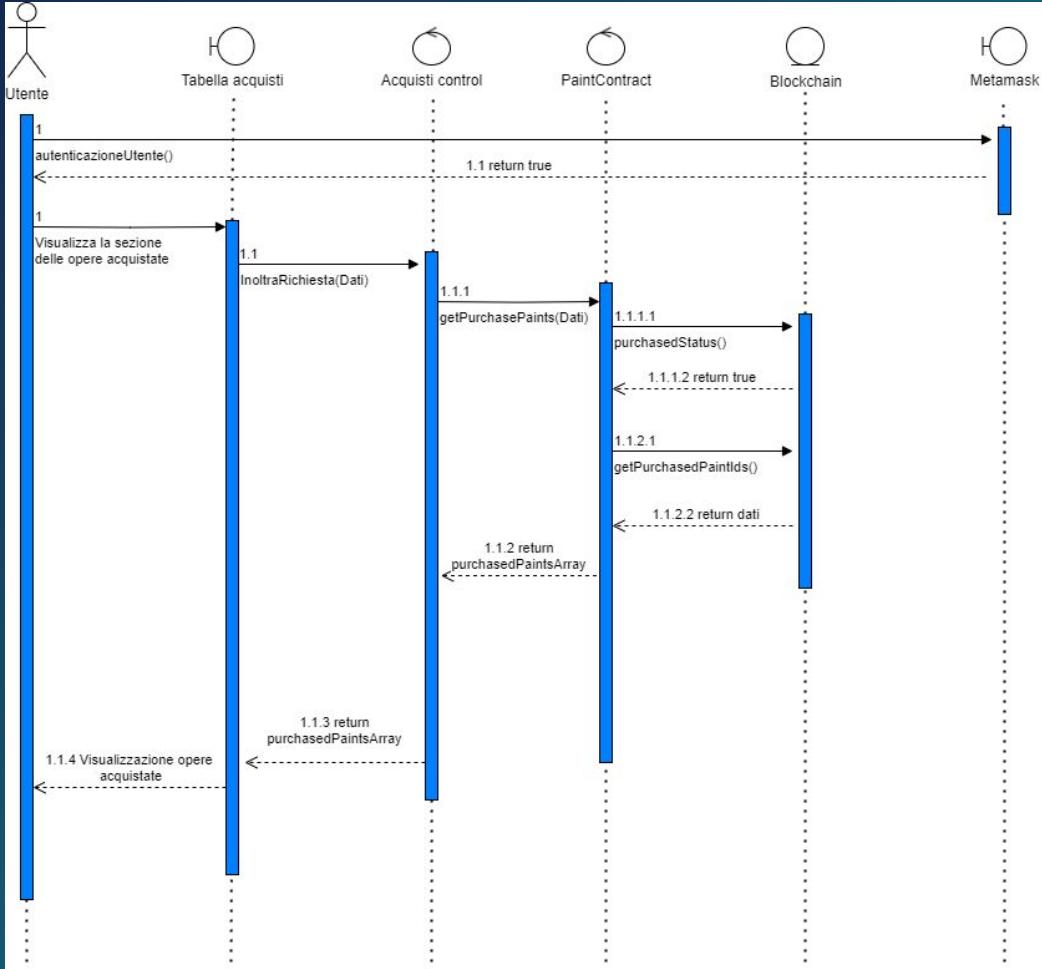


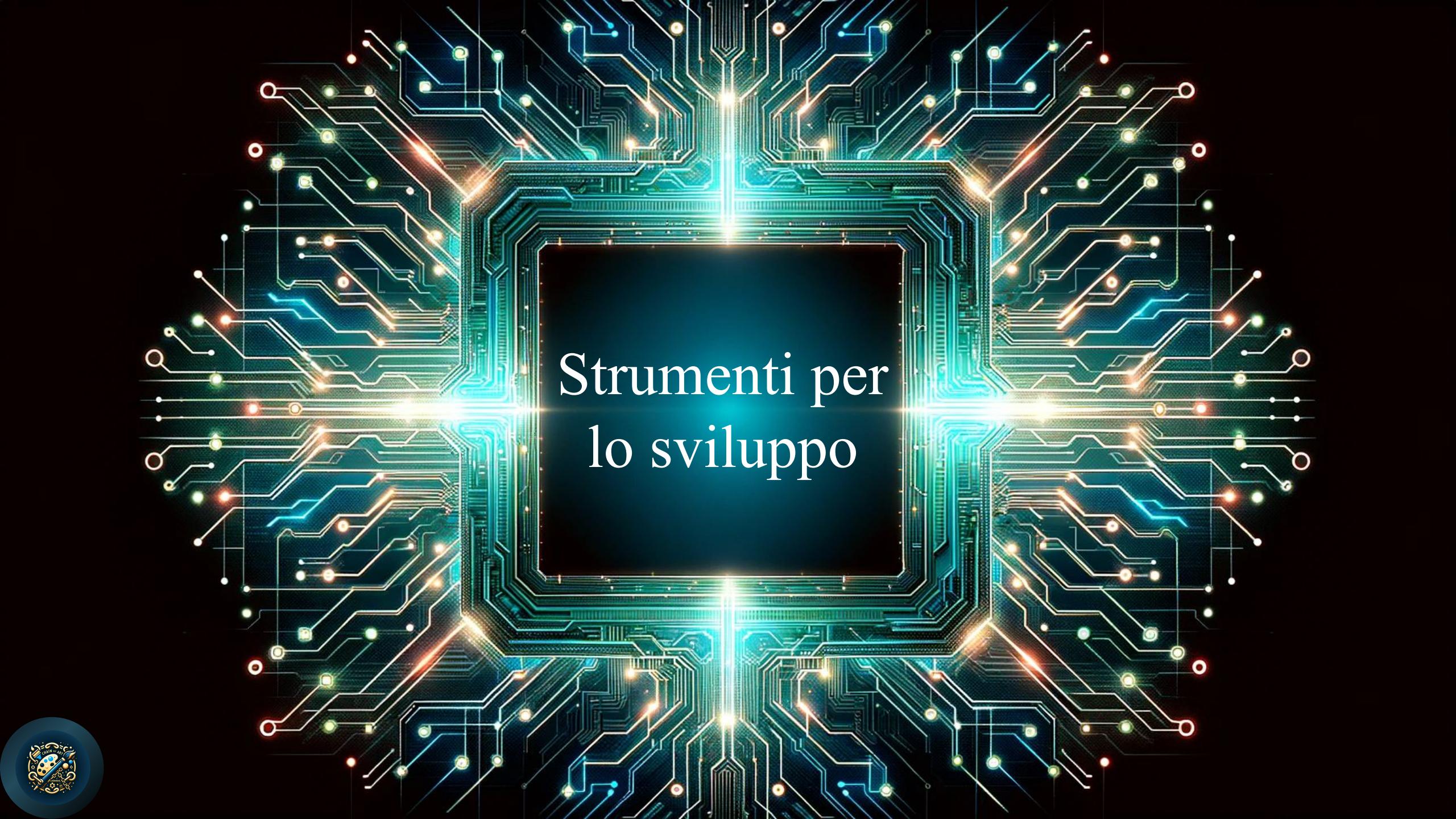
Boundary Control Entity



Sequence diagram

Visualizzazione delle opere d'arte acquistate





Strumenti per
lo sviluppo



Strumenti utilizzati

GANACHE

Simula l'ambiente blockchain
di Ethereum



TRUFFLE

Framework per lo sviluppo di
smart contract su blockchain



METAMASK

Wallet di criptovalute per
interagire con la blockchain



SOLIDITY
Linguaggio degli Smart Contracts

WEB3.js

Libreria JavaScript che permette
l'interfacciamento con le
blockchain

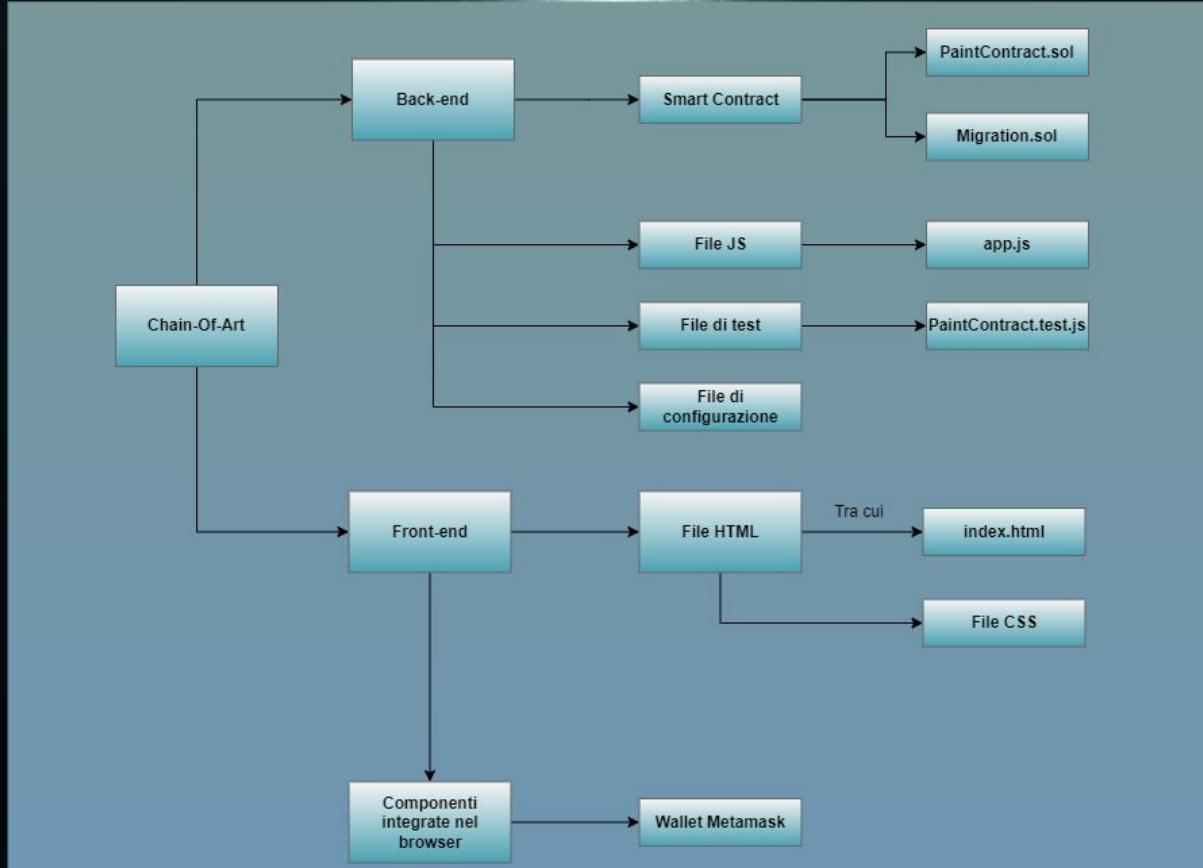


BOOTSTRAP

Framework di sviluppo per il
front-end



Strutturazione delle directory



Codice dello Smart Contract

```
contract PaintContract {
    uint public counterPaints;

    // Struttura di un singolo quadro
    struct Paint {
        uint id;
        string title;
        string img;
        string artist;
        string price;
        bool isSold;
    }

    // Mappatura degli ID dei quadri con i loro dettagli
    mapping(uint => Paint) public paints;

    // Mappatura degli indirizzi degli acquirenti con gli ID dei quadri acquistati
    mapping(address => uint[]) public purchasedPaints;

    // Evento emesso quando un quadro viene acquistato
    event purchasedPaint(address indexed buyer, uint indexed paintId);
```



Codice dello Smart Contract

```
// Funzione per ottenere i dettagli di un quadro dato il suo ID
function getPaintDetails(
    uint paintId
)
public
view
returns (
    uint id,
    string memory title,
    string memory img,
    string memory artist,
    string memory price,
    bool isSold
)

{
    Paint storage paint = paints[paintId];
    return (
        paint.id,
        paint.title,
        paint.img,
        paint.artist,
        paint.price,
        paint.isSold
    );
}

function paintExists(uint paintId) public view returns (bool) {
    return !paints[paintId].isSold;
}
```

```
// Funzione per acquistare un quadro
function buyPaint(uint paintId) public payable {
    Paint storage paint = paints[paintId];
    require(paintExists(paintId), "Il quadro non e' disponibile");

    // Verifica se il quadro è disponibile
    require(!paint.isSold, "Il quadro non e' disponibile");

    // Verifica se l'utente ha già acquistato questo quadro
    require(
        !purchasedStatus(msg.sender, paintId),
        "Il quadro non e' disponibile"
    );

    // Aggiorna lo stato del quadro come venduto
    paint.isSold = true;

    // Aggiungi l'ID del quadro agli acquisti dell'utente
    purchasedPaints[msg.sender].push(paintId);

    // Emetti l'evento per registrare l'acquisto
    emit purchasedPaint(msg.sender, paintId);
}
```



Codice dello Smart Contract

```
// Funzione di supporto per verificare se un utente ha già acquistato un quadro
function purchasedStatus(
    address buyer,
    uint paintId
) internal view returns (bool) {
    uint[] storage purchasedIds = purchasedPaints[buyer];
    for (uint i = 0; i < purchasedIds.length; i++) {
        if (purchasedIds[i] == paintId) {
            return true;
        }
    }
    return false;
}

// Funzione per ottenere gli ID dei quadri acquistati da un utente
function getPurchasedPaintIds(
    address buyer
) public view returns (uint[] memory) {
    uint[] memory purchasedIds = purchasedPaints[buyer];
    return purchasedIds;
}

// Funzione per ottenere i dettagli dei quadri acquistati da un utente
function getPurchasedPaints(
    address buyer
) public view returns (Paint[] memory) {
    uint[] memory purchasedIds = purchasedPaints[buyer];
    Paint[] memory purchasedPaintsArray = new Paint[](purchasedIds.length);

    for (uint i = 0; i < purchasedIds.length; i++) {
        purchasedPaintsArray[i] = paints[purchasedIds[i]];
    }
    return purchasedPaintsArray;
}
```



```
Compiling your contracts...
=====
> Everything is up to date, there is nothing to compile.

Starting migrations...
=====
> Network name:      'development'
> Network id:        5777
> Block gas limit:   6721975 (0x6691b7)

1_initial_migration.js
=====
Replacing 'Migrations'

> transaction hash: 0x590ff9052e47257206fa0f5b820dcc81fae5a6575ca44f543f3fc18e31fde9
> Blocks: 0           Seconds: 0
> contract address: 0x5E3D03797DA89104d3E3D9cF6Ba6f7419B0C9658
> block number:       1
> block timestamp:   1705945117
> account:            0x19B98084ec124D80f5F0423dc297092E129a5A37
> balance:             99.99915573025
> gas used:            250154 (0x3d12a)
> gas price:            3.375 gwei
> value sent:          0 ETH
> total cost:         0.00084426975 ETH

> Saving migration to chain.
> Saving artifacts
=====
> Total cost:        0.00084426975 ETH

2_deploy_contracts.js
=====
Replacing 'PaintContract'

> transaction hash: 0x94ab42acf48caf4b913d370cee35f97747e73f115dc363ae6691e0a80b066d94
> Blocks: 0           Seconds: 0
> contract address: 0xBfDc0c6d36beB2d67be69a0Fe58903DEc3E71999
> block number:       3
> block timestamp:   1705945120
> account:            0x19B98084ec124D80f5F0423dc297092E129a5A37
> balance:             99.98777821868133265
> gas used:            3532362 (0x35e64a)
> gas price:            3.178366198 gwei
> value sent:          0 ETH
> total cost:         0.011227139979899676 ETH

> Saving migration to chain.
> Saving artifacts
=====
> Total cost:        0.011227139979899676 ETH

Summary
=====
> Total deployments: 2
> Final cost:        0.012071409729899676 ETH
```

Deploy dello smart contract

Utilizzando Truffle effettuiamo il deploy dello smart contract, digitando i seguenti comandi nel prompt:

- **truffle compile:** per verificare la presenza di errori sintattici
- **truffle migrate:** per deploysare lo smart contract sulla Blockchain Ganache



Testing

Sono stati eseguiti dei test in JavaScript utilizzando il framework di testing per gli smart contracts di Ethereum chiamato **Mocha**, insieme alla libreria di assert di Node.js per le asserzioni.

Attraverso il seguente comando digitato sul prompt dei comandi, possiamo far partire i test.

```
truffle test
```

I cui risultati hanno dato esito positivo:

```
Contract: PaintContract
```

- ✓ should create a new paint (190ms)
- ✓ should allow buying a paint (161ms)
- ✓ should prevent buying the same paint twice (216ms)

```
3 passing (2s)
```



Demo Chain Of Art



[Chain Of Art - Landing page](#)

[Visualizza le opere](#)

[Visualizza acquisti effettuati](#)

[Chi siamo](#)

**Chain of Art : l'Arte Digitale tra
Blockchain & Web 3.0**



Selezione del quadro destinato all'acquisto

Selezione del quadro

Seleziona il quadro che desideri acquistare



Distese Stellate

Alessandro Vibrante

0.33

ETH

Acquista



Sinfonia di Spirali Cosmiche

Marco Tessitore

0.43

ETH

Acquista



Il giorno

Vittorio Formale

0.26

ETH

Acquista



Equilibrio Astratto

Silvio Baratta

0.47

ETH

Acquista



Notifica di acquisto di Metamask

Selezione del quadro

Seleziona il quadro che desideri acquistare



Distese Stellate
Alessandro Vibrante
0.33
ETH
[Acquista](#)


Sinfonia di Spirali Cosmiche
Marco Tessitore
0.43
ETH
[Acquista](#)


Il giorno
Vittorio Formale
0.26
ETH
[Acquista](#)

MetaMask Notification

Chain-of-Art

Account 3 → 0xFaD5A...307aE

http://localhost:3000

0xFaD5A...307aE : INTERAZIONE CONTRATTO

c 0.33 ETH

DETtagli HEX

Market >

Gas (estimated) 0.000485 0.000485 ETH
Likely in < 30 seconds Max fee: 0.000485 ETH

Totale 0.330485 0.330485 ETH
Amount + gas fee Max amount: 0.330485 ETH

[Annulla](#) [Conferma](#)

[Acquista](#)



Notifica di acquisto del quadro

localhost:3000 dice
Quadro Acquistato

OK

Seleziona il quadro che desideri acquistare



Distese Stellate
Alessandro Vibrante
0.33
ETH
[Acquista](#)



Sinfonia di Spirali Cosmiche
Marco Tessitore
0.43
ETH
[Acquista](#)



Il giorno
Vittorio Formale
0.26
ETH
[Acquista](#)



Equilibrio Astratto
Silvio Baratta
0.47
ETH
[Acquista](#)



Visualizzazione del quadro acquistato

Quadri Acquistati

ID	QUADRO	TITOLO QUADRO	ARTISTA	PREZZO	STATO
0		Distese Stellate	Alessandro Vibrante	0.33 ETH	ACQUISTATO





Conclusioni



Conclusioni

Chain of Art si pone come punto di riferimento nel mondo dell'arte digitale, dimostrando come la tecnologia blockchain possa essere un alleato prezioso in un contesto del genere.

Possibili sviluppi futuri



Gestione account multipli

Offrendo un'implementazione specifica per la gestione dell'account del venditore.



Introduzione di nuove funzionalità

Come ad esempio l'implementazione di un'asta virtuale per l'acquisto di opere digitali.



Impiego della Virtual Reality

Per dare la possibilità agli utenti di ammirare da "vicino" le opere esposte prima di acquistarle.





Grazie per
l'attenzione

