

# Sistema NFT per contenuti scientifici

*Autenticità, tracciabilità e valore.*

*Una soluzione blockchain per valorizzare e proteggere la proprietà intellettuale*



Progetto realizzato da  
Antonino Paternò



**DnA - Decentralized News & Articles**



## Problema

Garantire autenticità e valore ai contenuti scientifici.



Il sistema sviluppato integra NFT, randomicità verificabile (tramite Chainlink VRF) e un registro decentralizzato per gestire contenuti scientifici in modo trasparente e sicuro. Gli autori possono registrare i propri contenuti, specificando titolo, descrizione e numero massimo di copie disponibili.

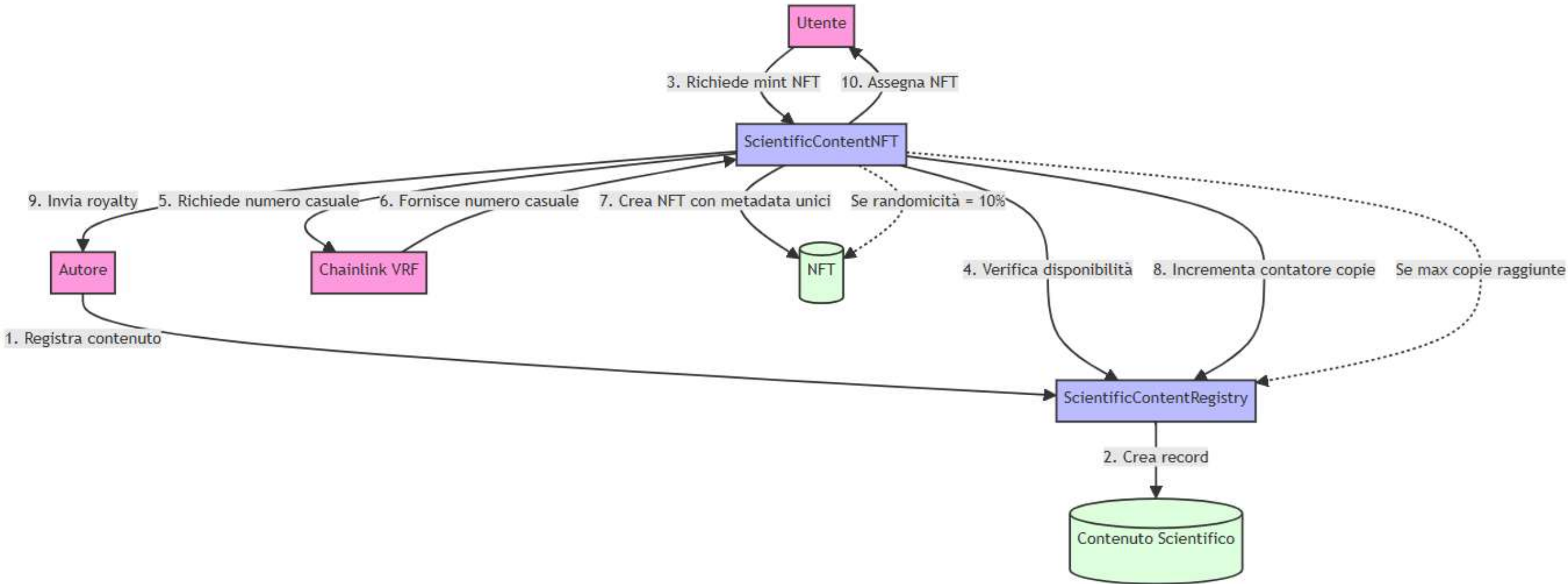
## Soluzione

Certificazione NFT e tracciabilità blockchain.



Ogni contenuto viene associato a un hash unico per garantirne l'autenticità. Gli utenti possono poi mintare NFT rappresentativi di questi contenuti, con metadati unici e una probabilità del 10% di ottenere un contenuto speciale.

# Sinottico implementazione DnA



# Come funziona il sistema?

1

**Autore → Registry**

Registrazione contenuto

2

**Utente → NFT Contract**

Richiesta minting

3

**Chainlink VRF → NFT Contract**

Fornitura numero casuale

4

**NFT Contract → Utente**

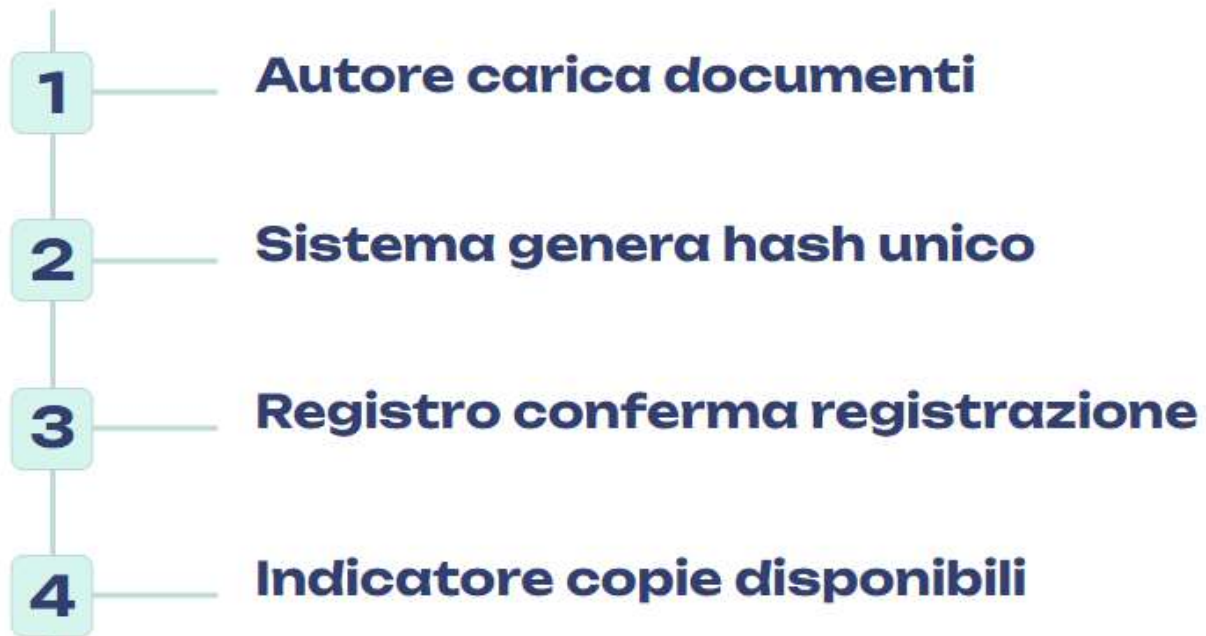
Assegnazione NFT

5

**NFT Contract → Autore**

Royalty

# Registrare un contenuto scientifico



# Creare un NFT





## ✂ Tecnologie , Framework e scelte progettuali (1/2):

- ✓ **Smart Contract** – *Sviluppati in Solidity, i contratti gestiscono la logica di registrazione, minting e distribuzione delle royalty. L'architettura a due contratti garantisce la separazione di responsabilità tra gestione dei contenuti (Registry) e tokenizzazione (NFT) migliorando modularità e facilità aggiornamento*
- ✓ **Hardhat** – *Ambiente di sviluppo e testing per SmartContract che ha permesso di eseguire test completi e simulare scenari complessi.*
- ✓ **Viem** – *Libreria per interagire con la blockchain e testare le chiamate ai contratti, scelta per la sua efficienza e facilità d'uso nello sviluppo di script e test.*

## ✂ Tecnologie , Framework e scelte progettuali (2/2):

- ✓ **Chainlink VRF** – *Utilizzato per generare numeri casuali verificabili, essenziali per il meccanismo dei contenuti speciali.*
- ✓ **Mock VRF Coordinator** – *Simulazione locale di Chainlink VRF per evitare dipendenze dalla rete*
- ✓ **Chai + Mocha** – *Framework di test per Solidity*
- ✓ **Atomicità nelle operazioni di minting** – L'incremento del contatore delle copie avviene nella stessa transazione della creazione dell'NFT, all'interno della funzione `_processMint` .  
Questo approccio:
  - Garantisce coerenza tra stato del token e registro dei contenuti
  - Previene stati inconsistenti in caso di fallimento della transazione
  - Blocca automaticamente tutto il processo se l'incremento fallisce
  - Ottimizza i costi del gas raggruppando le operazioni correlate



# Randomicità e contenuti speciali

## Chainlink VRF

Fonte di Randomicità



## Probabilità

Gli utenti possono mintare NFT rappresentativi di contenuti con una probabilità del 10% di ottenere un contenuto speciale.



**NFT Speciale**

# Royalty e pagamenti

1

**Pagamento  
utente**

0.05 ETH

2

**Calcolo  
automatico**

3%

3

**Trasferiment  
o royalty**

0.0015 ETH

4

**Restituzione  
eccesso**

All'utente

## ✂ Funzionalità chiave:

- ✓ **Registro dei contenuti** – *Gli autori registrano i contenuti nel ScientificContentRegistry, che ne memorizza i dettagli e garantisce l'autenticità tramite un hash unico.*
- ✓ **Minting degli NFT** – *Gli utenti pagano per mintare un NFT. Il sistema richiede un numero casuale tramite Chainlink VRF per generare metadati unici, inclusa la possibilità di contenuti speciali.*
- ✓ **Royalty automatiche** – *Il 3% del pagamento viene trasferito automaticamente all'autore come royalty, incentivando la creazione di contenuti di qualità.*
- ✓ **Edizioni limitate:**– *Ogni contenuto ha un numero massimo di copie, rendendo gli NFT più preziosi e collezionabili.*
- ✓ **Randomicità verificabile**– *Chainlink VRF garantisce che la randomicità sia imparziale e verificabile, aggiungendo un elemento di sorpresa e valore.*

# Un sistema innovativo

1

Autenticità

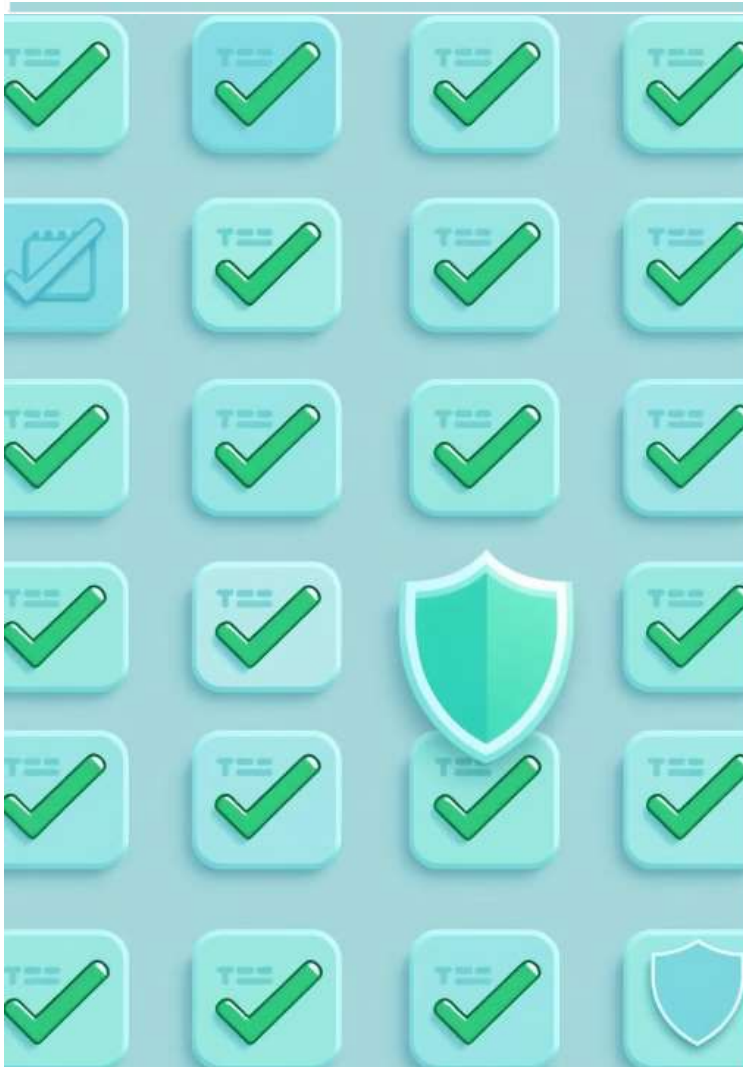
2

Tracciabilità

3

Valore aggiunto

## TEST DI VALIDAZIONE



**DeploymentTests** : Verifica che i contratti vengano deployati correttamente e che le configurazioni iniziali siano impostate come previsto

**VRFFunctionalityTests** : Verifica che il processo di minting degli NFT funzioni correttamente, inclusa la generazione di numeri casuali tramite VRF.

**SecurityAndAccessControlTests** : Verifica che solo gli utenti autorizzati possano eseguire determinate operazioni e che i pagamenti vengano gestiti correttamente

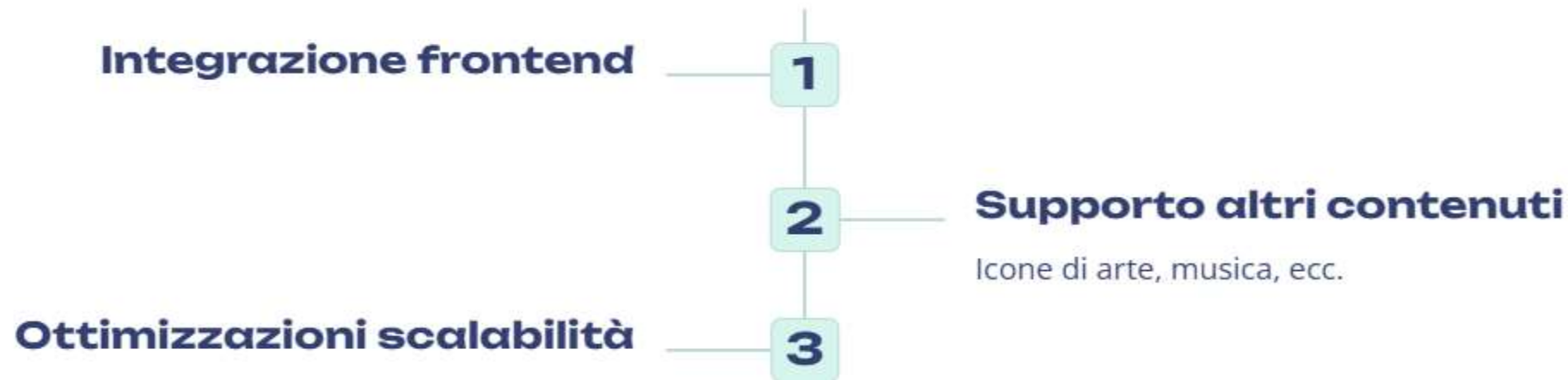
**EdgeCaseTests** : Verifica il comportamento del sistema in situazioni limite, come pagamenti insufficienti o superamento del numero massimo di copie.

**RoyaltyTests & RegisterContentTests**: Verifica che le royalty vengano correttamente calcolate e trasferite all'autore e che i contenuti siano registrati ed accessibili

**MintingTests & RandomnessTests**: Verifica il conio NFT con metadati corretti, e impedire conio quando si raggiunge il Max numero di copie impostato

**SpecialContentTests & TokenTransferTest**: Verifica che il contenuto speciale venga assegnato con una probabilità del 10% e che il trasferimento NFT sia effettuabile

# Futuri sviluppi



**Integrazione di un frontend:** *Per rendere il sistema accessibile a utenti non tecnici.*

**Espansione:** *Supporto per altri tipi di contenuti (es. opere d'arte, musica) e miglioramenti della scalabilità.*

**Comunità:** *Coinvolgere autori e collezionisti per creare un ecosistema vivo, dinamico e scalabile*

**REPOSITORY GIT:** <https://github.com/antopat1/ProgettoEthereumAdvancedDiAntoninoPaterno>

# ⚙️ Deploy su Blockchain pubblica Arbitrum Sepolia:

```
npx hardhat run scripts/deployContracts.ts --network arbitrumSepolia
```

🌐 Network: arbitrumSepolia

👤 Deployer Address: 0x6Db728e8A98b89C421CBb0cF23A3e9975a943B27

💰 Deployer Balance: 0.25023581777317 ETH

📁 Deploying ScientificContentRegistry...

📁 Deploying ScientificContentRegistry with args: []

📄 Bytecode length: 10742

💰 Gas Price: 0.00000000012 ETH

💰 Gas Limit: 3042449

⌚ Waiting for deployment transaction: 0xa7377b467d990a49e483ea0f71edbdd2dd7a078c11cf2eca18cca06f24e60cac

✅ ScientificContentRegistry deployed at: 0xbe3350ee3d6e476a8b4ff1b0b34daf29b580e34b

✅ ScientificContentRegistry deployed at: 0xbe3350ee3d6e476a8b4ff1b0b34daf29b580e34b

📁 Deploying ScientificContentNFT...

⌚ Waiting for deployment transaction: 0x5c700c6e1b958bfaba1376b25e01bb0b76e60d42cc0a38416a9fc0e348471a49

✅ ScientificContentNFT deployed at: 0x53300234e081e3f5da633baebfe52e3c6f0d9121

⚠️ Important: Add this NFT Contract Address to your Chainlink VRF Subscription:

PS C:\Users\Famiglia\_Paterno1\Desktop\roba\_programmazione\ProgettoEthereumAdvancedDiAntoninoPaterno>

🔗 NFT\_CONTRACT\_ADDRESS: 0x53300234e081e3f5da633baebfe52e3c6f0d9121

🔗 Adding NFT Contract as a consumer to Chainlink VRF Subscription...

🔗 Adding NFT Contract as a consumer to Chainlink VRF Subscription...

✅ NFT Contract added as a consumer to Chainlink VRF Subscription

🔗 Setting NFT Contract in Registry...

✅ NFT Contract set in Registry

✅ Deployment Summary

📁 Registry Address: 0xbe3350ee3d6e476a8b4ff1b0b34daf29b580e34b

📄 NFT Contract Address: 0x53300234e081e3f5da633baebfe52e3c6f0d9121

👤 Owner Address: 0x6Db728e8A98b89C421CBb0cF23A3e9975a943B27

🔗 VRF Coordinator: 0x5ce8d5a2bc84beb22a398cca51996f7930313d61

🔑 Key Hash: 0x1770bdc7eec7771f7ba4ffd640f34260d7f095b79c92d34a5b2551d6f6cfd2be

📄 Subscription ID: 811560626644328031001408335338184588499351602788007235688897319254018180985

*Ceck Tx EoA Deployer on  
<https://sepolia.arbiscan.io/>*

**0x6Db728e8A98b89C421CBb0cF23A3e9975a943B27**