

# DYNAMIC TOMATOES NFT

---

## AGRI-FOOD DIGITAL TWINS, dNFT e CHAINLINK

Nel settore agroalimentare, garantire trasparenza e qualità lungo la filiera è una sfida complessa. Problemi come la tracciabilità, l'autenticità e l'automazione limitano l'efficienza e la fiducia nel mercato.

Da qui l'esigenza di creare dei **Digital Twins** per implementare soluzioni digitali ad-hoc a questi problemi.

---



## Tracciabilità

Difficoltà nel monitorare l'intero ciclo di vita del prodotto, dalla coltivazione alla distribuzione.



## Autenticità

Rischio di frodi alimentari e certificazioni poco verificabili, con impatti su qualità e sicurezza.



## Automazione

Processi manuali e inefficienza nella gestione dei dati, con conseguente spreco di risorse.

I Dynamic NFT sono dei Non-Fungible-Token che, a differenza della loro più nota versione statica, **si evolvono nel tempo**, modificando i propri metadati al verificarsi di condizioni opportunamente codificate negli Smart Contracts. Per natura, quindi, ben si prestano all'interazione col reale mondo esterno, trovando riscontro in diversi campi di applicazione



## Generative Art and Games

Possono essere utilizzati per evolvere i tratti di personaggi o ritratti



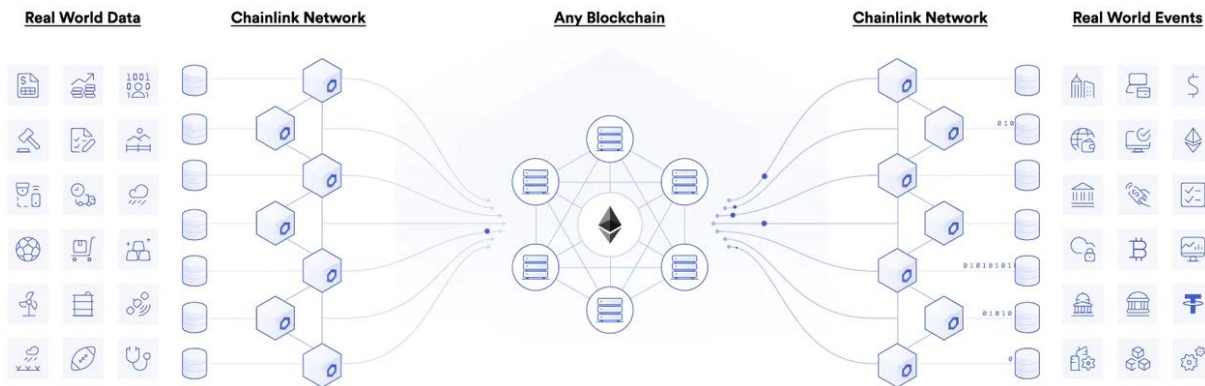
## Real-World Asset Tokenization

Trovano ampio riscontro nella digitalizzazione di asset del mondo reale di qualsiasi tipo, dall'immobiliare all'**agroalimentare**

# Chainlink: The Platform to Connect Smart Contracts

Chainlink è una rete decentralizzata di **oracoli** che permette agli smart contract di interagire in modo sicuro e decentralizzato con dati, API e sistemi off-chain

Tra i diversi servizi offerti troviamo le Chainlink **Functions**, che offrono tutti gli strumenti necessari per integrare dati ed elaborazioni dal mondo esterno on-chain, permettendo l'esecuzione di codice **JavaScript** direttamente dagli Smart Contracts



---

## DYNAMIC TOMATOES NFT



## 01. DIGITAL TWIN IMPLEMENTATION

Implementare delle repliche digitali, Digital Twins, di piante di pomodoro, attraverso l'utilizzo di dNFT, e sfruttando lo standard ERC-721.

## 02. REAL-WORLD DATA INTEGRATION

Sfruttare le Chainlink Functions per ottenere i dati metereologici, dei luoghi di piantagione delle piante, da sensori IoT, per semplificazione sostituiti da API pubblici di meteo (OpenWeatherMap)

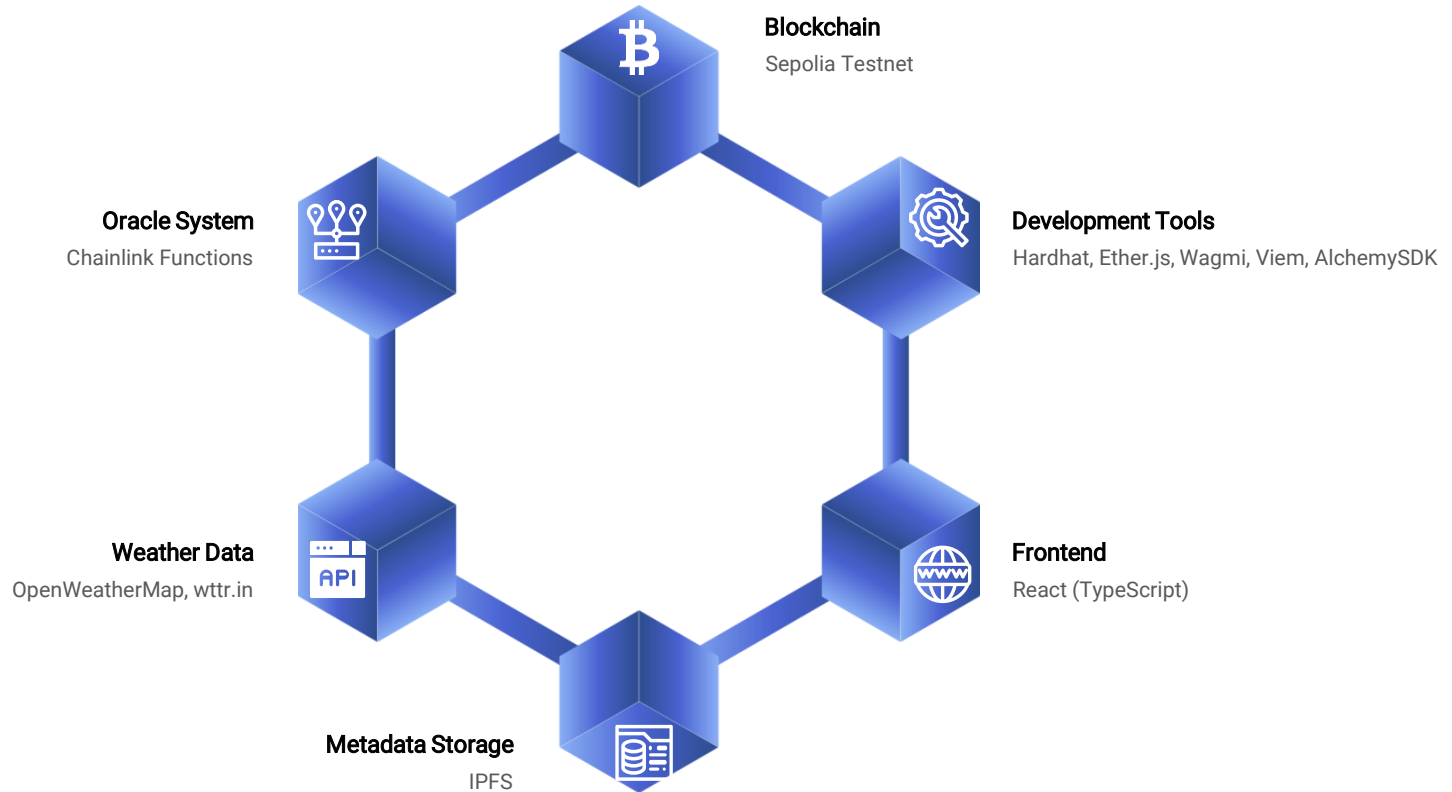
## 03. DYNAMIC GROWTH MECHANISM

Implementare un meccanismo di crescita automatico, o on-demand, delle piante di pomodoro, in modo da portarle dallo stadio iniziale di seminazione a quello finale di maturazione, in base ai dati metereologici

## 04. USER-FRIENDLY FRONTEND

Realizzare una applicazione web per l'interazione con il sistema on-chain, in modo da permettere agli utenti di «piantare», gestire e monitorare i propri digital twins.

## 05. A DEEP DIVE INTO BLOCKCHAIN





Lo smart contract implementa l'intera logica on-chain del sistema, in dettaglio:

01. STANDARD ERC-721
02. OWNERSHIP MANAGEMENT
03. NOTIFICATION EVENTS
04. CHAINLINK FUNCTIONS INTEGRATIONS
05. GROWTH MECHANISM

```
contract DynamicTomatoes is ERC721URIStorage,
    FunctionsClient, Ownable {

    ...

    function mint() public { ... }

    mapping(uint8 => address) private _tomatoOwners;

    event TomatoMinted(uint8 tomatoId, address owner);

    function fulfillRequest( ... ) internal override { ... }

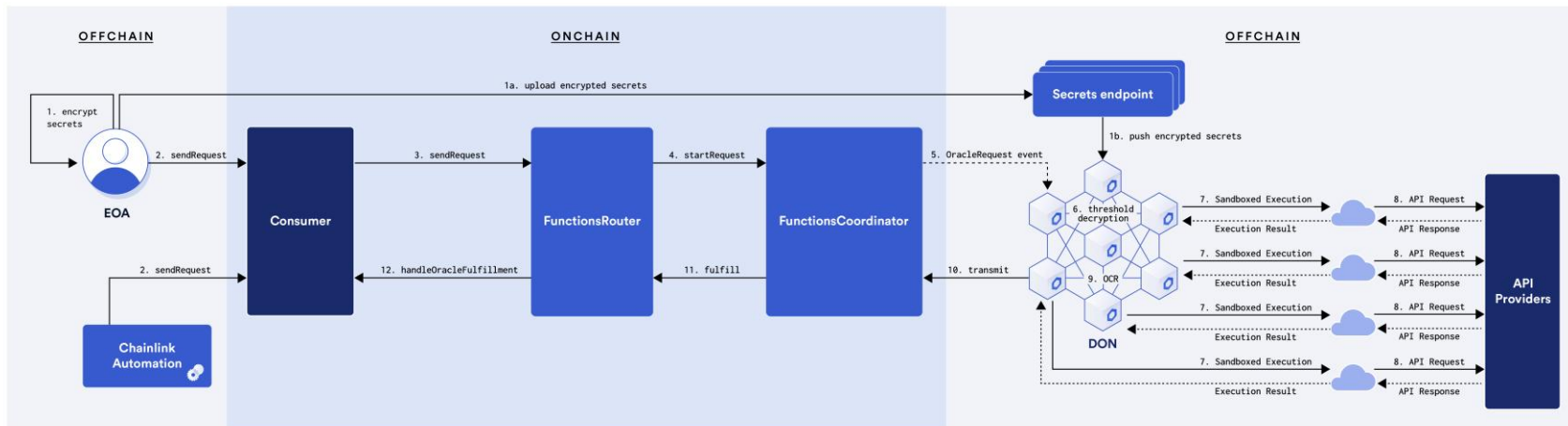
    function requestGrow(uint8 _tomatoId, ...) public { ... }

    ...

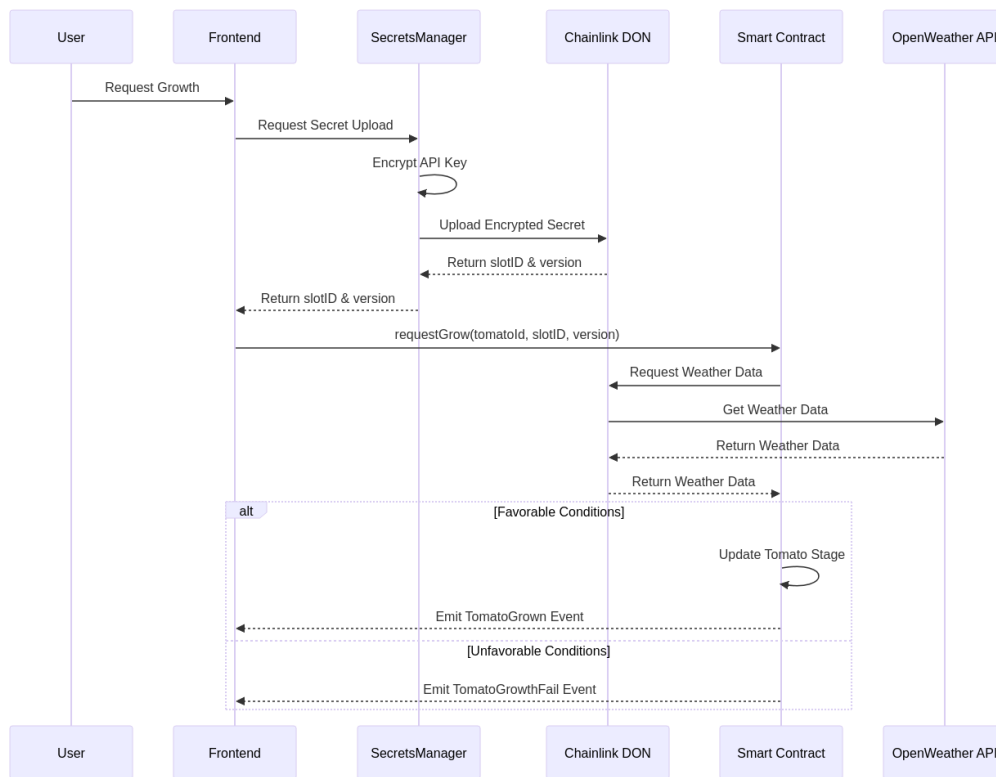
}
```

L'integrazione delle Chainlink Functions, per recuperare i dati meteo, è stato un punto focale del progetto. Nel dettaglio, è stato necessario dapprima includere nel contratto una funzione JavaScript per l'interazione con l'API provider dei dati meteo, eseguito on-demand dal Decentralized Oracle Network di Chainlink, responsabile dell'effettivo recupero dei dati.

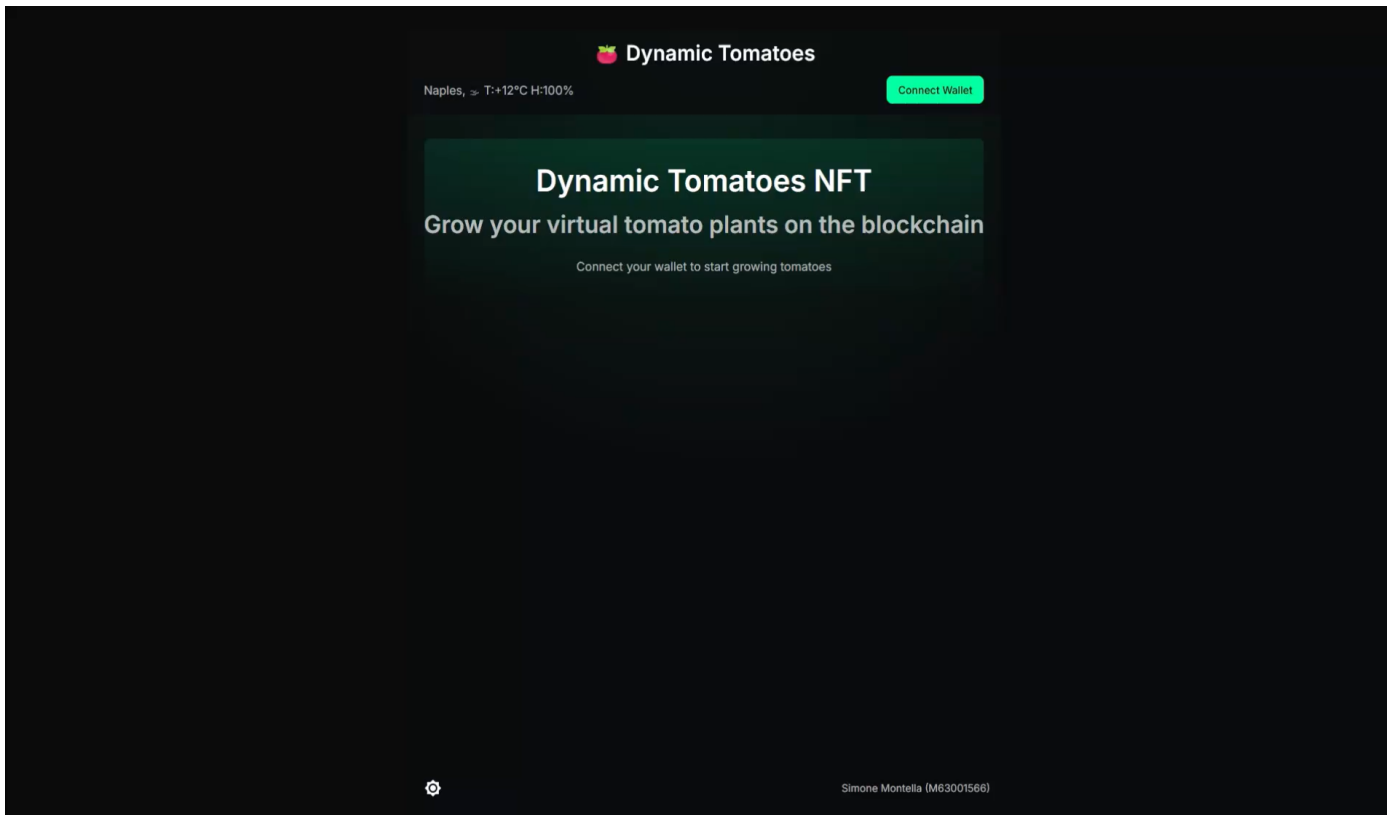
Dal momento che per l'interazione con l'API di OpenWeatherMap era necessario l'utilizzo di un API Key, è stato necessario inviare allo stesso DON le credenziali, criptate, per l'accesso all'API stesso. È stato perciò realizzato un WebService, off-chain, proprio a questo scopo.



01. Interagendo col Frontend, l'**utente** proprietario di un dNFT ne richiede l'evoluzione
02. Il **Frontend** richiama il WebService, il **SecretsManager**, per notificare l'inizio di una richiesta verso il DON
03. Il WebService cripta l'API Key e la invia al DON, restituendo al Frontend l'ID delle credenziali criptate ricevuto dal DON stesso
04. Il Frontend interagisce con lo **Smart Contract**, che richiede a sua volta al **DON** l'esecuzione dello script JavaScript fornendo anche l'ID delle credenziali appena ricevute
05. Ottenuti i dati meteo dal DON, il contratto determina se le condizioni meteorologiche sono favorevoli o no all'evoluzione del dNFT



Per l'interazione con lo Smart Contract è stata predisposta una WebApp, sviluppata in **React** e **Typescript**, in modo da offrire agli utenti tutte le funzionalità necessarie per l'interazione col contratto, in modo da poter gestire e monitorare i propri dNFT.





### RISULTATI CHIAVE

01. Implementazione di un diffuso caso d'uso dei dNFT
02. Integrazione on-chain di dati recuperati dal mondo reale
03. Sviluppo di un'interfaccia utente semplice ed efficace
04. Starting point per sviluppi futuri



### PROSPETTIVE FUTURE

01. Integrazione IoT
02. Memorizzazione di più dati riguardo la coltivazione e la filiera
03. Migration su Layer 2



### A PERSONAL BLOCKCHAIN JOURNEY

# Grazie per l'attenzione !



[linkedin.com/in/simone-montella](https://www.linkedin.com/in/simone-montella)



[github.com/simonemontella/DynamicTomatoesNFT](https://github.com/simonemontella/DynamicTomatoesNFT)



[simone.montella2@studenti.unina.it](mailto:simone.montella2@studenti.unina.it) / [simonemontella8@gmail.com](mailto:simonemontella8@gmail.com)