

Testarea unei Retele Blockchain - Alpha-Alpha Version

Magazin NFT cu abonamente

Obiectiv Alpha-Alpha

- Realizarea unei **analize state-of-the-art** pentru temă.
- Colectarea de materiale esențiale (articole, resurse, definiții).
- Identificarea aplicațiilor existente similare.
- Setup inițial pentru dezvoltarea și testarea rețelei Blockchain.

Contextul general al temei

De ce un magazin NFT cu abonamente?

- Abonamentele digitale sunt omniprezente (Netflix, Spotify etc.).
- NFT-urile permit un control real asupra abonamentului.
- Pot fi transferabile, revandute, verificate on-chain.

Arhitectura aplicației

<u>Componentă</u>	<u>Descriere</u>
Smart Contract (Solidity) ->	Creează și administrează produse sub formă de NFT
Frontend (React + Ethers.js) ->	Interfață pentru utilizatori
Hardhat ->	Framework de testare și deployment
Metamask ->	Wallet pentru conectare și tranzacții

Colectarea Materialelor Esențiale (state-of-the-art)

Articole Științifice relevante (PDF):

- [Testing Strategies for NFT Smart Contracts](#)
- [Smart Contract Security Best Practices \(Consensys\)](#)

- [Security Analysis of Non-Fungible Tokens](#)
- [Performance Analysis of NFT Marketplaces](#)
- [ERC-721 Non-Fungible Token Standard](#)
- [NFT-based Subscription Models](#)
- [Ethereum: A Secure Decentralised Generalised Transaction Ledger](#)

Pagini Web și Resurse Relevante:

- [OpenSea – cel mai mare Marketplace NFT](#)
- [Rarible – Marketplace NFT descentralizat](#)
- [Ethereum Smart Contracts Testing Guide](#)
- [ERC-721 și ERC-1155 – Standarde NFT](#)
- [OpenZeppelin Docs](#)
- [Hardhat Documentation](#)
- [Chainlink Docs](#)

Definiții Esențiale

<u>Concept</u>	<u>Definiție</u>
NFT (Non-Fungible Token) ->	Token unic pe blockchain care reprezintă un activ digital (în cazul nostru: abonamente).
Smart Contract ->	Cod care rulează automat pe blockchain, executând logică prestabilită.
Testare Unitare ->	Testarea funcțiilor individuale (ex: creare NFT, validare abonament).
Testare de Integrare ->	Testarea interacțiunii dintre contracte și aplicație web/backend.
Testare Performanță ->	Măsurarea vitezei, scalabilității și eficienței rețelei.
Testare Securitate ->	Detectarea vulnerabilităților (ex: reentrancy, overflow, frontrunning).

Analiza aplicațiilor existente

Exemple:

1. **Unstoppable Domains** – NFT ca domenii
2. **Audius** – platformă de streaming cu NFT-uri pentru acces
3. **Unlock Protocol** – oferă NFT-uri care acționează ca abonamente

Avantaje:

- Permite control complet utilizatorului
- Posibilitate de revânzare a NFT-urilor (abonamente transferabile)
- Transparent și descentralizat

Dezavantaje:

- Costuri ridicate pe rețelele Layer-1
- Dificultate în gestionarea expirării abonamentelor (necesită cron jobs sau oracole)
- Probleme de scalabilitate

Identificarea serviciilor și resurselor disponibile pentru testare

Retele:

- Ethereum Sepolia, Polygon Mumbai, Base

Framework:

- Hardhat, Truffle

Biblioteci:

- OpenZeppelin, Chainlink

Wallet:

- Metamask, WalletConnect

Testare Unitară și de Integrare

- **Truffle** (Solidity / Ethereum)
- **Hardhat** (Testare rapidă și debugging Ethereum smart contracts)
- **Ganache** (Rețea Ethereum locală pentru teste rapide și izolate)

Testare Performanță

- **JMeter** (Simulare cereri simultane pentru marketplace NFT)
- **Hyperledger Caliper** (benchmarking blockchain)
- **Apache Benchmark** (testarea performanței API-urilor marketplace)

Testare Securitate

- **Mythril** (vulnerabilități smart-contracts Ethereum)
- **Slither** (analiză statică smart-contracts NFT)
- **Echidna** (fuzzing pentru detectarea vulnerabilităților NFT)

Client App – NFT Marketplace / Product Subscription DApp

Features:

1. **Create Product** – creează un NFT ce reprezintă un produs / abonament
2. **Buy Product** – cumpără un NFT (abonament activ)
3. **Delete Product** – doar deținătorul îl poate șterge
4. **Add Review** – utilizatorii care au cumpărat pot lăsa review-uri

Setup Inițial:

1. Precondiții:

- Node.js (v18+ recomandat)
- npm (sau yarn)
- Git (opțional, dar util)

2. Inițializare proiect:

```
mkdir my-blockchain-project
```

```
cd my-blockchain-project
```

```
npm init -y
```

```
npm install --save-dev hardhat
```

```
npx hardhat
```

3. Structura proiectului:

```
my-blockchain-project/
```

```
|
```

```
├── contracts/      # Smart contracts în Solidity
```

```
|   └── Lock.sol    # Contract de exemplu
```

```
|
```

```
├── scripts/        # Scripturi de interacțiune/testare
```

```
|   └── deploy.js
```

```
|
```

```
├── test/           # Teste (Mocha/Chai)
```

```
|   └── Lock.js
```

```
|
```

```
├── hardhat.config.js  # Configurație Hardhat
```

```
├── package.json
```

```
└── README.md
```

4. Rulare blockchain local

```
npx hardhat test
```

```
npx hardhat run scripts/deploy.js --network localhost
```

```
npx hardhat node
```

