Jaani Kaukonen

Marketing in Web 3.0: Leveraging Non-Fungible Tokens (NFTs)

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor’s Thesis

27 April 2024

Abstract

Author:	Jaani Kaukonen
Title:	Marketing in Web 3.0: Leveraging Non-Fungible Tokens (NFTs)
Number of Pages:	22 pages
Date:	27 April 2024

Degree:	Bachelor of Engineering
Degree Programme:	Information and Communications Technology
Professional Major:	Mobile Solutions
Supervisors:	Toni Spännäri, Senior Lecturer

Web 3.0 is the new and improved version of the internet which is based on decentralization and user ownership where blockchain technology is the facilitator. One of the main components of this technological innovation is known as non-fungible tokens (NFTs), which are digital assets that are unique and represent ownership of assets including art, music, or other collectibles. NFTs offer promising tools for marketers in Web 3.0, which enables them to engage and attract their users as the technology develops.

The purpose of this thesis is to examine the potential of NFTs in the context of different use cases, including real estate, event ticketing, and software access. Alongside elaborating on the technology behind NFTs, how they operate, the standards they use, it presents the development process of a decentralized token-gated web application on Ethereum network. Throughout the project, a selection of tools and technologies such as Next.js, TypeScript, Solidity, and Hardhat are used for the development and deployment the application.

Keywords: Web 3.0, Non-Fungible Tokens (NFTs), Marketing, Blockchain, Decentralization

Tiivistelmä

Tekijä:	Jaani Kaukonen
Otsikko:	Markkinointi Web 3.0 -ympäristössä NFT-teknologiaa hyödyntäen
Sivumäärä:	22 sivua
Aika:	27.4.2024
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja Viestintätekniikka
Ammatillinen pääaine:	Mobiiliratkaisut
Ohjaajat:	Lehtori Toni Spännäri

Yhtenä viimeisimmistä innovaatiosta internetin saralla voidaan pitää Web 3.0 -teknologiaa. Web 3.0 on käyttäjiensä omistama hajautettu järjestelmä, joka hyödyntää lohkoketjuteknologiaa turvallisuuden ja läpinäkyvyyden takaamiseksi. Lyhenne NFT tulee englannin kielen sanoista non-fungible token, mikä tarkoittaa ainutlaatuista ja korvaamatonta valtuutta tai polettia. Nämä ovat ainutlaatuisia valtuuksia, joiden avulla voidaan määrittää esimerkiksi kenelle taiteen tekijänoikeudet kuuluvat ja todentaa digitaalisen tai fyysisen asian omistajuus. NFT-teknologia tarjoaa työkaluja myös markkinointialan osaajille ja mahdollistaa sekä tehokkaiden, että kohderyhmälle personalisoitujen markkinointikampanjoiden luonnin.

Tämän tutkielman tarkoituksena on käsitellä NFT-teknologian potentiaalia Web 3.0 -ympäristössä. Tutkielma sisältää esimerkkejä aina kiinteistövälityksestä lipunmyyntiin sekä pääsyoikeuksien hallinnasta erilaisiin ohjelmistoihin. Osana tätä tutkielmaa kehitetään Web 3.0 -verkkosovellus, jossa käyttäjälle annetaan yksilöllinen NFT-valtuus, jonka avulla käyttäjä auktorisoidaan verkkosovelluksen käyttäjäksi. Kehitysprosessissa käytetään lukuisia erilaisia teknologioita, kuten Next.js, TypeScript, Solidity ja Hardhat, joiden avulla luodaan käytännönläheinen lohkoketjua hyödyntävä ohjelmisto.

Avainsanat: Web 3.0, Non-fungible token (NFT), markkinointi, lohkoketju, hajautettu tietojärjestelmä

Contents

List of Abbreviations

Web 3.0: The next iteration of the internet, characterized by decentralization, user ownership, and transparency.

NFT: Non-fungible token. A unique digital identifier stored on a blockchain that represents ownership of a digital asset.

ETH: Ether. The native cryptocurrency of the Ethereum blockchain platform.

1 Introduction

Web 3.0 is a decentralized version of the internet, and it is owned by its users. The utilisation of blockchains, cryptocurrencies and non-fungible tokens (NFTs) allow secure and transparent transactions in a peer-to-peer network.

Non-fungible tokens (NFTs) are a type of digital assets that are not only unique and irreplaceable but also a way to represent ownership of a digital or a physical asset such as art, music, or even real estate. NFTs have the potential to challenge traditional marketing by offering campaigns that are both more engaging and effective. For example, with the help of NFTs brands can create exclusive content for their customers and reward them for their loyalty.

This thesis will present the potential of NFTs with a focus on marketing in Web 3.0. It will explain the benefits of NFTs, as well as the challenges attached to them. Alongside several use cases of NFTs, it will provide a comprehensive description on the development process of a token gated decentralized web application.

2 Web 3.0

The evolution of the internet dates to 1990s when the initial protocols were developed. Ever since then it has undergone a major transformation from static websites to dynamic user-generated content, and interaction between its users. The latest innovation in this field is referred to as Web 3.0, which describes a vision of a new and improved version of the internet. Web 3.0 utilizes blockchains, cryptocurrencies, and NFTs and unlike the current centralized infrastructure, it is instead designed to be a decentralized system. This refers to Web 3.0 being constructed, operated, and owned by its users. [1.]

2.1 History of the World Wide Web

We often think of the web as a continuous pillar of modern era – it was invented and has persisted ever since. Nevertheless, the web today is vastly different from original conception. To understand the evolution of the Word Wide Web, it can be divided into the following categories: Web 1.0, Web 2.0, and Web 3.0. [1.]

Web 1.0: Read-Only (1990-2004)

In the early days the web consisted mainly of static websites owned by enterprises. There were hardly any interactions between users and individuals rarely produced content – leading it to be known as the read-only web. [1.]

Web 2.0: Read-Write (2004-present)

In the early 2000s, social media platforms began to emerge, and the web shifted towards user-generated content, known as the read-write era. Social media companies provided platforms for users to share content and engage in user-to-user interactions, which ultimately led to the development of the advertising-driven revenue model – users create the content without owning it or benefit from its monetization. [1.]

Web 3.0: Read-Write-Own (2014 onwards)

The concept of Web 3.0 was first introduced by Gavin Wood, co-founder of Ethereum, shortly after the platform's launch in 2014. It provided a solution for a problem that many enthusiasts had been struggling with. The current form of web relies on trusting small number of private companies to act in the public's best interest. By leveraging the use of blockchains, cryptocurrencies and NFTs, Web 3.0 aims to give the control back to the users in the form of ownership. [1.]

2.2 Definition

To properly describe technological innovations often proves to be a challenging task. At the time of composing this thesis, four core principles of Web 3.0 can be defined. [1.]

Decentralization

In its current form the internet is based on a centralized framework, which is mostly regulated and owned by a handful of major technology companies. By utilizing blockchain technology, Web 3.0 stands in contrast to the current paradigm by distributing ownership in decentralized manner. This form of architecture enhances both transparency and autonomy. [1.]

Permissionless

Another key aspect of Web 3.0 is its permissionless nature, wherein equal access is granted to all individuals who wish to participate. In contrast to the current centralized system, achieving this does not require the involvement of central governing bodies that may restrict access. This permissionless nature not only allows users to engage transparently but also enhances security by eliminating the need for a central governing body and the compromised actions of a few individuals. [1.]

Native payments

The full potential of the internet can only be fully harnessed with an infrastructure for transferring value. Unlike the centralized models that rely on banking infrastructure and payment processors, Web 3.0 employs cryptocurrencies for transferring value. By eradicating third-party intermediaries, Web 3.0 makes the financial services and economic incentives more accessible than the traditional system. [1.]

Trustless

Web 3.0 is a trustless platform, which means that it operates without trusted third-party intermediaries. Instead, Web 3.0 transfers the trust into decentralized mechanisms and thus, creates an ecosystem that is transparent and maintains its integrity. This fosters more accessible digital space and encourages its users for autonomy and self-governance. [1.]

3 NFT Technology

NFTs can be seen as ground-breaking innovation that has transformed the asset ownership. By utilizing blockchain technology, they enable tokenization of various items such as artwork and real estate. Unlike fungible tokens, NFTs are distinguished by their unique identifier codes and metadata. This section delves into the fundamental concepts behind NFTs and how they transform digital ownership.

3.1 Introduction to NFTs

Tokenization of items including art, collectibles, digital and physical assets is facilitated using NFTs and their concept of ownership. Immutable nature of blockchains ensures the transparency and verifiability of NFTs’ record of ownership. [2.]

3.1.1 Definition

NFTs approach to asset ownership is made possible by blockchain technology. Unlike their fungible counterparts, such as cryptocurrencies, NFTs represent unique items or assets that have been converted into digital tokens. Each NFT is equipped with unique identification codes and metadata to set them apart from other tokens within the same blockchain ecosystem. While cryptocurrencies are fungible, allowing for tokens to be interchangeable, NFTs depart from this with individual properties and characteristics. Although two NFTs originating from the same blockchain may seem to resemble each other, they are non- interchangeable, each possessing its own exclusive value and origin. This difference emphasizes the uniqueness and irreplaceability of NFTs. [3.]

3.1.2 Benefits

NFTs offer a variety of benefits, from simplifying investment processes to securing digital identities. However, one of the most notable advantages of

NFTs is their potential to significantly improve market efficiency. By tokenizing physical assets, NFTs allow for the removal of intermediaries, thus streamlining the sales process and making it much more efficient. For instance, consider an NFT representing a digital or physical artwork, such as a song or a painting. Storing these assets on a blockchain eliminates the need for agents, enabling sellers to directly connect with their audience. This direct interaction not only speeds up transactions but also empowers creators to engage more intimately with their buyers. In essence, NFTs have the power to transform traditional market dynamics, offering a more efficient and direct means of asset exchange. [3.]

Investing

In conventional markets, ownership of a stock is recorded on ledgers that include information such as the shareholder’s name, date of issuance, certificate number, and the number of shares held. In contrast, a blockchain is a distributed and secured ledger, and issuing NFTs to represent shares achieves the same objective as issuing stocks. However, leveraging blockchain and NFTs rather than a stock ledger enables the automation of ownership transfer through smart contracts. Once a share, or in this instance, an NFT representing a share, is sold, the blockchain can verify, audit, and execute the transfer of the ownership in a decentralized and transparent manner. These principles can be applied to various other assets where proof of ownership is essential, such as real estate, concert tickets, intellectual property rights, among others. The conversion of these principles to other assets is a straightforward process, as the underlying concept of tokenization remains the same. By representing ownership with NFTs on a blockchain, the need for intermediaries and centralized record-keeping is eliminated, thereby enhancing the transparency, security, and efficiency of ownership transfer for a broad range of assets. [3.]

3.2 Technical Aspects of NFTs

Blockchain is the foundational technology behind cryptocurrencies, and it serves a similar purpose in the case of NFTs. Both cryptocurrency and NFT transactions are facilitated using a distributed ledger on a decentralized network and are stored in a digital wallet. However, there is a significant difference between the two in that each non-fungible token is composed of a set of metadata that imparts unique properties to NFTs. This metadata serves as a digital fingerprint, allowing each NFT to be easily distinguished from others on the network. [4.]

This section explores the importance of token standards and smart contracts in the development of decentralized applications and ecosystems on the Ethereum blockchain. It will showcase the different types of token standards such as ERC20, ERC721, ERC777, and ERC1155, and their unique features and use cases. Additionally, it will explain how smart contracts work and they enforce the rules that enables interactions between users and applications on the blockchain.

3.2.1 Token Standards

Standards are crucial for ensuring the reliability and interoperability of technology, and this applies to NFTs as well. By defining how they should behave and how developers can interact with them, standards provide a foundation for building decentralized applications and ecosystems. The Ethereum blockchain has several NFT standards, including ERC20, ERC721, ERC777, and ERC1155, each with its unique features and use cases. [5.]

ERC20

Widely used standard for fungible tokens, which are tokens that are identical and interchangeable with each other. It is ideal for creating medium of exchange currency, voting rights, and staking tokens among others. ERC20 tokens have no unique attributes or behaviour associated with them, making them easy to manage and exchange. [6.]

ERC721

Non-fungible token standard that represents unique assets such as collectibles, real estate, or digital art. Each ERC721 token has a distinct ID and metadata, making it unique and irreplaceable. ERC721 is more complex than ERC20 and requires multiple smart contracts to manage the token ownership and transfer. [7.]

ERC777

Standard that builds on ERC20 and provides additional functionality for fungible tokens. The most notable feature of ERC777 tokens is the ability to receive hooks, which are functions that are called when tokens are transferred to an account or contract. This enables developers to create more complex token interactions and trigger actions in response to token transfers. [8.]

ERC1155

Multi -token standard that combines features of ERC20 and ERC721. It enables a single smart contract to represent multiple types of tokens, whether they are fungible or non-fungible. This approach provides significant gas savings and reduces the complexity of managing multiple token contracts. ERC1155 is particularly useful for gaming and collectibles projects, where multiple tokens may be required for different game items or collectibles. [9.]

3.2.2 Smart Contracts

A smart contract is a collection of functions and data that resides at a specific address on the Ethereum blockchain. It runs as a program and defines rules that are automatically enforced via the code, much like a regular contract. Smart contracts cannot be deleted by default, and interactions with them are irreversible.

Smart contracts are deployed to the network as a type of Ethereum account, but they are not controlled by a user or entity. Instead, accounts managed by users can interact with a smart contract by submitting transactions that execute a function on the contract. Smart contracts are funded by ether (ETH), and their state can be updated by the execution of these transactions. Listing 1 shows a simple example of a vending machine if it were a smart contract written in Solidity. [10.]

```
pragma solidity 0.8.7;

contract VendingMachine {

    // Declare state variables of the contract
    address public owner;
    mapping (address => uint) public cupcakeBalances;

    // When 'VendingMachine' contract is deployed:
    // 1. set the deploying address as the owner of the contract
    // 2. set the deployed smart contract's cupcake balance to 100
    constructor() {
        owner = msg.sender;
        cupcakeBalances[address(this)] = 100;
    }

    // Allow the owner to increase the smart contract's cupcake balance
    function refill(uint amount) public {
        require(msg.sender == owner, "Only the owner can refill.");
        cupcakeBalances[address(this)] += amount;
    }

    // Allow anyone to purchase cupcakes
    function purchase(uint amount) public payable {
        require(msg.value >= amount * 1 ether, "You must pay at least 1 ETH
per cupcake");
        require(cupcakeBalances[address(this)] >= amount, "Not enough cupcakes
in stock to complete this purchase");
        cupcakeBalances[address(this)] -= amount;
        cupcakeBalances[msg.sender] += amount;
    }
}
```

Listing 1. Smart contract written in Solidity that demonstrates a vending machine. [11.]

4 NFT Marketing cases

While NFTs are commonly associated with artwork and collectibles, their use cases extend far beyond that. This section explores three different scenarios of NFT marketing, highlighting the diversity of applications where NFTs can be applicable. The cases examined include the use of NFTs in real estate, event ticketing, and software access. This section provides in depth understanding of the potential benefits of NFTs with examination of different use cases in various industries.

Case 1: Real estate

NFT real estate commonly refers to programmable plots of land in virtual worlds or metaverses, allowing users to explore them with 3D avatars or purchase NFTs linked to parcels of land that represent unique ownership of their properties.

However, the impact of NFTs on real estate is not limited to virtual spaces alone. Traditional real estate relies on paper-based title deeds that are vulnerable to loss, alteration, or forgery. Even when stored in databases or other centralized systems, there is a risk of trusting third-party intermediaries to act in the public's best interest. By contrast, NFTs stored on a blockchain ledger offer greater security and transparency. Anyone can track the ownership history and previous transactions of an NFT and verify its authenticity. Moreover, the immutable nature of blockchain ledgers means that minting your title deed as an NFT could reduce the risk of malicious actors falsifying your property documents. NFTs have the potential to revolutionize traditional real estate by offering a secure and transparent alternative to paper-based systems. [12.]

4.1 Case 2: Event ticketing

The demand for tickets to popular events often exceeds the available supply, resulting in rapid sell-outs. However, the growth of automated software or "bots" that purchase these tickets at an unprecedented rate, with the intention of reselling them at inflated prices on secondary markets, has exaggerated the situation. According to Distil Networks' 2019 report *10 Popular NFT Use Cases*, bots are responsible for 39 percent of all ticketing traffic. This phenomenon not only generates frustration and increased costs for consumers, but it also leads to significant financial losses for event organizers and artists. [13.]

However, NFTs offer a solution to the problems associated with event ticketing. Like conventional physical or digital tickets, an NFT can grant access to live or virtual events. Not only does this provide a more secure and convenient alternative to traditional ticketing methods, but it also reduces the risk of scams and scalping using blockchain technology, which offers greater transparency and authenticity verification. [13.]

The benefits of NFT ticketing extend beyond preventing ticketing fraud, as it creates new opportunities for issuers to interact with customers. By collecting data associated with the holders of a specific NFT, issuers can offer perks like surprise giveaways, token-gated sites and services, and access to exclusive experiences and fan clubs. The unique features of NFTs open new possibilities for event organizers and artists to engage with their audience and offer personalized experiences that were not possible before. [13.]

4.2 Case 3: NFT token gating

Token gating is a verification method that enables projects, communities, and businesses to provide exclusive access to their spaces, events, content, or products for individuals and businesses who hold specific digital assets in their wallets. This technique is essential for success in marketing in Web 3.0, not only by generating exclusivity for token holders and building community among them but also by granting token owners access to special rewards. [13.]

Token gating creates an exclusive environment for token holders, encouraging engagement and loyalty among them. By limiting access to a particular product or event to token holders, businesses can create a sense of community around their brand, building a loyal customer base. Additionally, token gating can also offer token holders access to exclusive rewards, incentivizing them to hold onto their tokens and invest in the success of the project. [13.]

The next chapter of this thesis will focus on building a token gated software, which will allow individuals, businesses, and communities to implement token gating in their digital spaces easily.

5 Development of a token gated web application

A key aspect of this thesis was developing a token gated web application to showcase the capabilities of smart contracts. The application consists of two main parts: a smart contract for issuing tokens (NFTs) used for granting access, and a user interface for interacting with the smart contract to gain access to the token gated features.

Web 3.0 applications typically don't rely on traditional server-side infrastructure (figure 1). Instead, they focus on user interaction with the blockchain. On the client side, we have the user interface accessible via web browsers and the connections between the client and the blockchain (figure 2). [14.]

The development process of this token gated web application illustrates the integration of blockchain technology with user-friendly design principles, resulting in an intuitive and effective platform representing the potential of decentralized applications. Through smart contracts, NFTs, and user interfaces, this application demonstrates how Web 3.0 can transform access control and asset ownership paradigms.

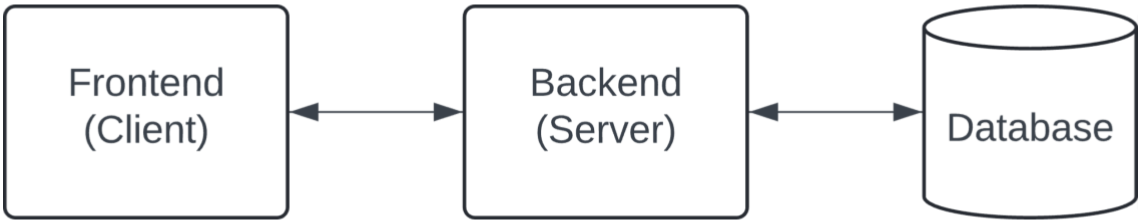

Figure 1. A conceptual mode of traditional Web 2.0 architecture.

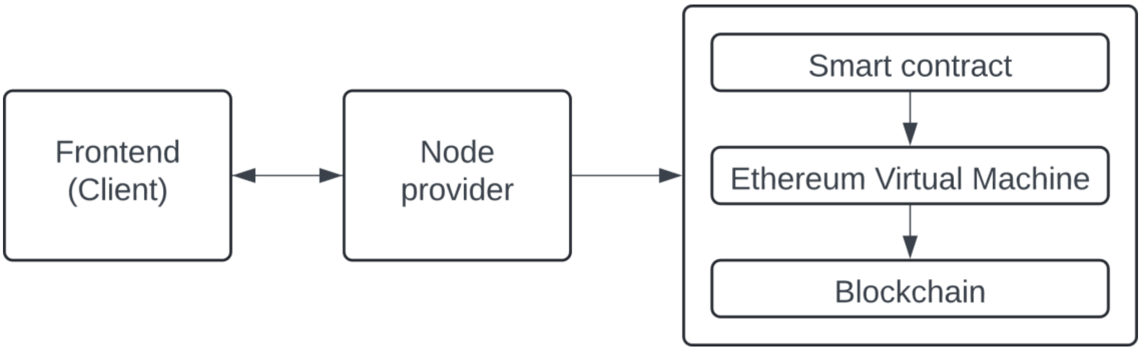

Figure 2. A conceptual model of Web 3.0 architecture.

The development utilizes a stack of modern tools and technologies to ensure efficiency and effectiveness throughout the implementation process. Development process followed test-driven model to ensure desired functionality of the smart contract, as they are immutable after deployment.

5.1 Tools and Technologies

The tools and technologies chosen for this project followed the industry's best practices and are widely used in decentralized application development.

Next.js

The project utilizes Next.js as a frontend framework due to its modular architecture which allows for code reusability and maintainability. As it is built on React and with its server-side rendering capabilities, it has become widely adopted framework in the industry.

TypeScript

TypeScript was chosen for its static typing system to enhance the code quality and maintainability. It allows for the detection of potential bugs in the source code during the development process which, plays an important role in ensuring more stable and reliable application.

Solidity

Solidity is the main programming language for smart contract development on the Ethereum blockchain and was therefore chosen for the project. The smart contract that implemented the token-gating logic and minting process of the NFTs was developed using Solidity.

Hardhat

Hardhat is the development environment for Ethereum smart contracts that offers tools and functionalities such as testing and deployment. Since it is widely used in the Ethereum ecosystem and ensures the reliability of the smart contract, it was chosen for the project.

In summary, Next.js, TypeScript, Solidity, and Hardhat were selected to ensure well tested, functional, and reliable application, following the best practices in software development.

5.2 Implementation

The implementation phase involved several steps including the development of the client-side application, development of the smart contract, and its integration with the frontend, and thorough testing as well as the deployment of the contract itself.

Smart Contract Development

A key part of the application is the smart contract written in Solidity. It is responsible for token gating and ownership on the Ethereum blockchain. The contract serves as the main access point of the application, allowing user interaction, access control mechanism, and ensures the integrity and transparency of the token transactions. Listing 2 shows the source code of the smart contract for token gating application.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/utils/Counters.sol";

contract AccessToken is ERC721 {
    using Counters for Counters.Counter;
    Counters.Counter private _tokenIdCounter;

    constructor() ERC721("AccessToken", "ATK") {
        _tokenIdCounter.increment();
    }

    // Mapping to store which addresses have access
    mapping(address => bool) private _hasAccess;

    // Mint a new access token (NFT)
    function mint() public {
        require(!_hasAccess[msg.sender], "Wallet already owns an NFT");
        uint256 tokenId = _tokenIdCounter.current();
        _mint(msg.sender, tokenId);
        _hasAccess[msg.sender] = true;
        _tokenIdCounter.increment();
    }

    // Check if an address has access
    function hasAccess(address user) public view returns (bool) {
        return _hasAccess[user];
    }
}
```

Listing 2. Solidity smart contract that handles the token generations as well as token gating logic.

Integration with Next.js Frontend

The interaction between the client-side and the smart contract utilizes functionalities from Hardhat, such as connection functions for the Metamask wallet, as well as Sepolia test network with Alchemy acting as the node provider. This integration allows the Next.js frontend to interact with the smart contract and allow user to generate the NFTs. Listing 3 shows how the frontend of the application interacts with the blockchain.

```
useEffect(() => {
    if (!isConnected) {
        return
    }

    const walletHasAccess = async () => {
        const provider = new ethers.BrowserProvider(window.ethereum)
        const signer = await provider.getSigner()
        const contract = new ethers.Contract(
            accessTokenAddress,
            accessToken.abi,
            signer
        )

        try {
            const response = await contract.hasAccess(signer)
            setHasAccess(response)
        } catch (e) {
            console.log("error: ", e)
        }
    }

    walletHasAccess()
}, [accounts, hasAccess, isConnected])
```

Listing 3. React Hook that connects to the blockchain to verify if the user possess the NFT required for access.

Testing

Immutable nature of the blockchain requires comprehensive testing throughout the development process to ensure the reliability and security of the application. Unit test for validating the contracts functionality and robustness were implemented and executed using Hardhats extensive testing features. Listing 4 shows how unit testing for the smart contract ensures its reliability.

```
const { ethers } = require("hardhat")
const { expect } = require("chai")

describe("AccessToken contract", () => {
    let token
    let owner
    let addr1
    let addr2

    beforeEach(async () => {
        const AccessToken = await ethers.getContractFactory("AccessToken")
        token = await AccessToken.deploy()
        await token.waitForDeployment();

        [owner, addr1, addr2] = await ethers.getSigners()
    })

    it("should allow a wallet to mint one NFT", async () => {
        await token.connect(owner).mint()
        expect(await token.balanceOf(owner.address)).to.equal(1)
    })

    it("should not allow a wallet to mint more than one NFT", async () => {
        await token.connect(owner).mint()
        try {
            await token.connect(owner).mint()
        } catch (error) {
            expect(error.message).to.contain("Wallet already owns an NFT")
        }
    })

    it("should allow a wallet to check ownership of an NFT", async () => {
        await token.connect(owner).mint()
        const hasAccess = await token.hasAccess(owner.address)
        expect(hasAccess).to.be.true
    })

    it("should not allow a wallet to check ownership of an NFT it does not
own", async () => {
        const hasAccess = await token.hasAccess(addr1.address)
        expect(hasAccess).to.be.false
    })

    it("should allow a different wallet to mint one NFT", async () => {
        await token.connect(addr1).mint()
        expect(await token.balanceOf(addr1.address)).to.equal(1)
    })

    it("should allow a different wallet to check ownership of its NFT", async
() => {
        await token.connect(addr1).mint()
        const hasAccess = await token.hasAccess(addr1.address)
        expect(hasAccess).to.be.true
    })

    it("should not allow a wallet to check ownership of another wallet's NFT",
async () => {
        await token.connect(addr1).mint()
        const hasAccess = await token.hasAccess(owner.address)
        expect(hasAccess).to.be.false
    })
})
```

Listing 4. Unit tests for testing the logic of the smart contract.

5.3 Deployment

The deployment phase was orchestrated using Hardhat, with Alchemy as a node provider, and the Sepolia test network as the target.

The decision to deploy the application on the Sepolia test network, rather than the main Ethereum blockchain, was made for several reasons. Foremost among these was the deployment costs during the development and testing phases of process.

Hardhat played a key role in the deployment process, and it was chosen because of its comprehensive set of features and ease of deployment. Listing 5 shows the deployment script of the smart contract.

```
import pkg from "hardhat"
const { ethers } = pkg

async function main() {
    const accessToken = await ethers.deployContract("AccessToken")

    await accessToken.waitForDeployment()

    console.log("AccessToken deployed to: ", accessToken.getAddress())
}

main().catch((error) => {
    console.error(error)
    process.exitCode = 1
})
```

Listing 5. Script that deploys the smart contract using Hardhat.

Hardhat proved to be versatile tool in the decentralized application development from compiling smart contracts to debugging and deploying them onto the blockchain.

5.4 Results

The development process resulted to a decentralized web application, introducing an alternative approach to access control through token gating mechanisms. Users can connect their Metamask wallet to the application, initiate a simple process of minting an NFT, which act as digital keys, granting access to exclusive areas of the application.

This application serves as a proof of concept, demonstrating the potential of token gating for access control. Looking ahead, there are opportunities for future improvements. One is enhancing the NFTs by incorporating content into them which can increase the depth and value of digital assets.

Additionally, expanding support to various wallet solutions beyond Metamask has the potential to enhance the accessibility of the application.

6 Conclusions

NFTs, alongside various other blockchain technologies, have the potential to revolutionize multiple industries, including marketing in Web 3.0 ecosystem. They offer several unique benefits, from more engaging and effective marketing campaigns to digital identity and ownership. However, as with every emerging technological innovation, there are also challenges that need to be overcome, such as the complexity of blockchain technology and the lack of awareness of the benefits that NFTs can provide.

This thesis has explored various use cases of NFTs, ranging from event ticketing to software access and even real estate. However, it is important to acknowledge that the use cases extend far beyond the scope of this thesis, and they are constantly evolving. By emphasizing the potential of NFTs in marketing within the Web 3.0 ecosystem and demonstrating the development of a token-gated web application, this thesis has shown the practical implementation of NFTs in creating more engaging and effective campaigns, as well as how smart contracts can revolutionize access control and asset ownership.

The development process utilized a stack of modern tools and technologies, including Next.js, TypeScript, Solidity, and Hardhat. These tools were selected to ensure the application's functionality and maintainability, following best practices in software development.

Overall, NFTs are a promising tool for marketers in Web 3.0 as well in other industries, and smart contracts offer variety of use cases ranging from one domain to another. As the technology matures and becomes more widely adopted, we can expect to see even more innovative and effective marketing campaigns that leveraging NFTs.

References

1 Introduction to Web3. Online material. Ethereum < https://ethereum.org/en/web3/> Updated 17.2.2023. Read 23.3.2.2024.

2 Non-fungible-Token (NFT). Online material. Ethereum < https://ethereum.org/en/nft/> Read 23.3.2024.

3 Non-Fungible Token (NFT): What It Means and How It Works. Online material. Investopedia < https://www.investopedia.com/non-fungible-tokens-nft-5115211> Read 27.3.2023.

4 NFTs and the blockchain. Online material. Inmind < https://inmindsoftware.com/2021/05/27/technical-aspects-of-nfts/> Updated 27.2.2021. Read 4.11.2023.

5 OpenZeppelin Documentation. Online material. OpenZeppelin <https://docs.openzeppelin.com/contracts/3.x/tokens#different-kinds-of-tokens> Read 10.12.2023.

6 OpenZeppelin Documentation. Online material. OpenZeppelin <https://docs.openzeppelin.com/contracts/3.x/tokens#ERC20> Read 4.4.2024.

7 OpenZeppelin Documentation. Online material. OpenZeppelin <https://docs.openzeppelin.com/contracts/3.x/tokens#ERC721> Read 4.4.2024.

8 OpenZeppelin Documentation. Online material. OpenZeppelin <https://docs.openzeppelin.com/contracts/3.x/tokens#ERC777> Read 4.4.2024.

9 OpenZeppelin Documentation. Online material. OpenZeppelin <https://docs.openzeppelin.com/contracts/3.x/tokens#ERC1155> Read 4.4.2024.

10 Ethereum documentation. Online material. Ethereum < https://ethereum.org/en/developers/docs/smart-contracts/> Read 4.4.2024.

11 Ethereum documentation. Online material. Ethereum < https://ethereum.org/en/developers/docs/smart-contracts/> Read 4.4.2024.

12 What Is NFT Real Estate and How Is It Changing the Real Estate Industry. Online material. Binance < https://www.binance.com/en/blog/nft/what-is-nft-real-estate-and-how-is-it-changing-the-real-estate-industry-2253735085697354496> Updated 15.9.2022. Read 5.4.2024.

13 10 Popular NFT Use Cases. Online material. builtin < https://ethereum.org/en/developers/docs/smart-contracts/> Updated 19.12.2022. Read 5.4.2024.

14 Web 2.0 architecture Vs Web 3.0 architecture. Online material. Medium < https://medium.com/web3-magazine/web-2-0-architecture-vs-web-3-0-architecture-c87ff647231Updated 7.3.2022. Read 27.4.2024.