A BLOCKCHAIN-BASED EDUCATIONAL ASSETS MANAGEMENT AND ACCESS CONTROL MODEL FOR THE METAVERSE

2023
MASTER THESIS
COMPUTER ENGINEERING

Muhammed HOCAOĞLU

Thesis Advisor
Assoc. Prof. Dr. Adib HABBAL

A BLOCKCHAIN-BASED EDUCATIONAL ASSETS MANAGEMENT AND ACCESS CONTROL MODEL FOR THE METAVERSE

Muhammed HOCAOGLU

Thesis Advisor
Assoc. Prof. Dr. Adib HABBAL

T.C.
Karabuk University
Institute of Graduate Programs
Department of Computer Engineering

Prepared as
Master Thesis

KARABUK
June 2023

I certify that in my opinion the thesis submitted by Muhammed HOCAOĞLU titled "A BLOCKCHAIN-BASED EDUCATIONAL ASSETS MANAGEMENT AND ACCESS CONTROL MODEL FOR THE METAVERSE" is fully adequate in scope and in quality as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Adib HABBAL .....
Thesis Advisor, Department of Computer Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a Master of Science thesis. June 15, 2023

Examining Committee Members (Institutions)	Signature
Chairman : Assist. Prof. Dr. Nehad T.A. RAMAHA (KBU)	.....
Member : Assoc. Prof. Dr. Adib HABBAL (KBU)	.....
Member : Assoc. Prof. Dr. Rafet DURGUT (BANU)	.....

The degree of Master of Science by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabuk University.

Prof. Dr. Müslüm KUZU .....
Director of the Institute of Graduate Programs

"I declare that all the information within this thesis has been gathered and presented in accordance with academic regulations and ethical principles and I have according to the requirements of these regulations and principles cited all those which do not originate in this work as well."

Muhammed HOCAOĞLU

ABSTRACT

M.Sc. Thesis

A BLOCKCHAIN-BASED EDUCATIONAL ASSETS MANAGEMENT AND ACCESS CONTROL MODEL FOR THE METAVERSE

Muhammed HOCAOĞLU

Karabuk University
Institute of Graduate Programs
The Department of Computer Engineering

Thesis Advisor:
Assoc. Prof. Dr. Adib HABBAL
June 2023, 54 Pages

After the outbreak of the COVID-19 pandemic, many schools and organizations switched over to online learning as opposed to the traditional method. Moving to Online education has many disadvantages like Lack of face-to-face interaction and isolation of students that effects their motivation. Metaverse has the ability to gather students and teachers remotely in shared virtual spaces which offer a perfect alternative for online education methods. Various modern educational resources became available as a result. The increasing popularity of online education has made the protection of educational content and the rights of the authors who create the educational material crucial. Hence, a effective model was required to ensure that unauthorized actions could not take place. This thesis proposes a decentralized model to protect and secure the educational assets in Metaverse. Our proposed model uses blockchain technology to store the educational assets as NFT. The decentralized environment of blockchain makes it impossible to modify the stored assets.The proposed model offers a time-

based access control scheme to enable owners of the asset to control who can access their access and he duration of access. The model protects both educational institutes and individuals against unauthorized modifications and fraud operations of their assets. The validation of our proposed model was done using Scyther tool. Gas cost-based evaluation was performed using Goerli test network. The obtained results of performance evaluation and validation shows the efficiency of using our proposed model in terms of performance efficiency and security that provide resistance from popular attacks providing time-based access control and cost effectiveness. The proposed model will afford a secure environment for protecting the intellectual property of users and enable them to manage their assets.

Key Words **:** Educational assets, Blockchain, NFT, Smart contract, Metaverse
Science Code : 92403

ÖZET

YÜKSEK LİSANS TEZİ

SANAL EVREN İÇİN BLOK ZİNCİR TABANLI EĞİTİM VARLIKLARI YÖNETİMİ VE ERİŞİM KONTROL MODELİ

Muhammed HOCAOĞLU

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Doç. Dr. Adib HABBAL
Haziran 2023, 54 Sayıfa

COVID-19 salgınının patlak vermesinden sonra birçok okul ve kuruluş, geleneksel yönteme kıyasla çevrimiçi eğitime geçti. Çevrimiçi eğitime geçiş, yüz yüze etkileşimin eksikliği ve öğrencilerin izole olması gibi birçok dezavantajı beraberinde getirir ve bu da motivasyonlarını etkiler. Sanal evren, öğrencileri ve öğretmenleri uzaktan bir araya getirebilen paylaşılan sanal alanlarda buluşturarak çevrimiçi eğitim yöntemleri için mükemmel bir alternatif sunar. Bu sayede çeşitli modern eğitim kaynakları da ortaya çıkmıştır. Çevrimiçi eğitimin giderek popülerleşmesi, eğitim içeriğinin korunması ve içerik oluşturan yazarların haklarının önemini artırmıştır. Bu nedenle, yetkisiz eylemlerin gerçekleşmesini engellemek için etkili bir model gerekmektedir. Bu tez, Sanal evrendeki eğitim varlıklarını korumak ve güvence altına almak için merkezi olmayan bir model önermektedir. Önerilen modelimiz, eğitim varlıklarını NFT olarak

blok zinciri teknolojisi kullanarak depolamaktadır. Blok zincirinin merkezi olmayan yapısı, depolanan varlıkların değiştirilmesini imkansız hale getirir. Önerilen modelimiz, varlık sahiplerinin kimin erişim sağlayabileceğini ve erişim süresini kontrol edebilmelerini sağlayan bir zaman tabanlı erişim kontrol düzeni sunmaktadır. Model, eğitim kurumlarını ve bireyleri varlıklarının yetkisiz değişikliklere ve sahtekarlık işlemlerine karşı korur. Önerilen modelimizin geçerliliği Scyther aracı kullanılarak doğrulanmıştır. Gaz maliyeti tabanlı bir değerlendirme ise Goerli test ağı kullanılarak gerçekleştirilmiştir. Performans değerlendirmesi ve doğrulama sonuçları, önerilen modelimizin performans verimliliği ve güvenliği açısından etkili olduğunu, yaygın saldırılara karşı direnç sağladığını ve zaman tabanlı erişim kontrolü ve maliyet etkinliği sunma konusunda etkili olduğunu göstermektedir. Önerilen model, kullanıcıların fikri mülkiyetini korumak ve varlıklarını yönetmek için güvenli bir ortam sunacaktır.

Anahtar Kelimeler : Eğitim varlıkları, Blok zinciri, NFT, Akıllı sözleşme, Sanal evren.

Bilim Kodu **:** 92403

ACKNOWLEDGMENT

First of all, Elhamdulillah for everything I have achieved. Much thanks to my parents and friends for their support during my journey. I would like to give thanks to my advisor, Assoc. Prof. Adib HABBAL, for his great interest and assistance in the preparation of this thesis.

And say, “My Lord, increase me in knowledge.” QURAN 20:114

CONTENTS

Page

Hiçbir içindekiler tablosu öğesine rastlanmadı.

Hiçbir içindekiler tablosu öğesine rastlanmadı.

LIST OF FIGURES

LIST OF TABLES

Page

SYMBOLS AND ABBREVIATIONS INDEX

*BC:*Blockchain

*BOLL:*Blockchain of Learning Logs

*CID:*Content Identifier

*EI:*Educational Institute

*FT:*Fungible Token

*IPFS:*Interplanetary File System

*MOE:*Ministry of Education

*NFT:*Non Fungible Token

*POS:*Proof of Stake

*POW:*Proof of Work

*SC:*Smart contract

*VR:*Virtual Reality

PART 1

INTRODUCTION

1.1. RESEARCH BACKGROUND

In recent years, problems with academic fraud and corruption have dramatically increased, according to a study by Cris Shore [1]. This make Goverments all around the world start comissions to reduce the percentages of fraud and corruption in the educational institutes by enacting strict laws. Due to the Turkish Penal Code (TCK) article 204/1, the penalty for issuing or using a counterfeit official document is imprisonment for a period of two to five years [2]. Fraud operations were also widely used to infringe on the intellectual property of authors and publishers. After COVID-19 pandemic most educational institutes started to provide online education, also some education providers started using metaverse due to it's ability to provide interactive content. That leads to the need for a model able to manage the educational content, prevent fraud operations, and preserve the intellectual property of users.

Blockchain is a distributed, immutable, and decentralized database that stores the transactions of users without the need for a third party. Blockchain was first introduced by Satoshi Nakamoto in 2008 [3]. Blockchain was first used for financial purposes as a cryptocurrency like Bitcoin and Ethereum. Blockchain has become a trend in the last few years as it started to be used in many fields like health care, tourism, and education due to its ability to store data in a peer-to-peer network, which enables all users to keep a copy of the ledger, which secures the data from fraud and modifications [4].

Non-fungible tokens (NFTs) are unique and exchangeable tokens on a blockchain network due to their scarcity and differing types and values [5]. Due to their ability to preserve intellectual property, NFTs started to be used in many industries, such as art, music, and education. The use of NFTs in the education sector presents a promising

opportunity to manage and authenticate educational assets, including certificates, diplomas, scientific articles, and degrees.

By creating an NFT for each asset, educational institutions can ensure the authenticity and ownership of the asset. It can also provide an immutable record of the asset's history. This can prevent fraud operations and make it easier for employers and institutions to verify the validity of an educational asset. Overall, the use of NFTs in managing educational assets presents a promising opportunity to improve the transparency, authenticity, and efficiency of the education sector [5].

1.2. PROBLEM STATEMENT

Educational institutions face a challenges in protecting the educational certificates and preserving the intellectual property of other educational assets from fraud operations [5]. Meanwhile Metaverse education is still using a centralized database to store the educational content which has a potential of modification while the data have single point of failure [6]. This leads to the need for a decentralized solution to manage educational assets and ensure the integrity of educational content. Based on the advantages of NFTs for managing educational assets, there is a growing need for a model that can authenticate ownership of educational assets in Metaverse using blockchain technology [5]. Therefore, there is a need for a decentralized model to manage educational assets in Metaverse, protect the intellectual property of the assets, and save them from possible fraud operations.

1.3. RESEARCH OBJECTIVES

The main objective of this study is to design and develop a decentralized model for managing educational assets in the Metaverse using blockchain technology. In particular, we aim to achieve the following specific objectives:

To design an NFT-based scheme that enables the Educational Institute (EI) to mint NFT to protect the educational content on Metaverse from fraud operations and enable

users to preserve their intellectual property. This can be achieved by smart contracts that will be developed and the immutable ledger that store the data of the minted NFT. To design a time-based access control management scheme on the Ethereum public blockchain that enables users to control their assets and identify who can access them and the period of access.

1.4. RESEARCH QUESTİONS

- How can NFTs be effectively used to manage educational assets and protect intellectual property in the educational field?
- How to enable owners of the asset to identify who can access their assets and specify access period for the viewers?

1.5. SCOPE

The study will focus on the useage of NFTs for managing educational assets in Metaverse and propose a new NFT-based model for managing educational content. While previous works were focusing only on managing certificates, our proposed model has the ability to handle all types of the educational content. The model propose a time-based access control mechanism to control the accessibility of the minted assets. The model will allow assets owners to fully control their assets and give them the ability to specify the duration of accessing the asset for each user. The model is powered by the Ethereum blockchain but due to the high cost of implementing and testing our proposed model over the public network we used Goerli test network for testing and implementing purposes.

1.6. SUMMARY

In this chapter, a brief introduction to blockchain technology is given. Followed by an introduction to NFT technology and its usage for managing educational assets. Afterward, the problems faced in preserving intellectual property in the educational sector are introduced. Then the potential of NFT to be a solution for managing the educational content and it’s ability to enable users to manage their own contents. Also

the enhancements that will be added in our model to the previous works are given.The chapter ends with a description of the scope of the research. In next chapter we will introduce a literature review for the work, mention the related works to our model, and make a vomparasion between our model and them.

PART 2

LITERATURE REVIEW

In this section, an introduction to blockchain technology will be given, blockchain architecture will be explained, and a brief view of consensus algorithms will be given. Then generations of blockchain will be discussed, followed by the advantages of using blockchain technology in several sectors. Also, we will explain about NFT technology, explaining its components, protocols, and advantages. Finally, related works that used NFT for managing the educational content will be compared with our new proposed model.

2.1.BLOCKCHAİN TECHNOLOGY

Blockchain was first proposed by Satoshi NAKAMOTO in 2008. Blockchain is a distributed and immutable ledger that stores transactions done by users without the need for a trusted third party [7]. Blockchain was first used for financial purposes, as in Bitcoin and Ethereum. Blocks are sets of transactions done over the blockchain and stored in blockchain nodes. Nodes hold identical versions of the transactions. When a block is created, it is broadcast to other nodes, so all nodes are up-to-date [8].

Blockchain has many advantages can be mentioned as below.

- Decentralization: Blockchain is a decentralized technology which means it is not controlled by any single entity, making it more secure and transparent. We mention some of the advantages of using blockchain below [9 , 10].
- Immutability: Once data is recorded on the blockchain, it cannot be modified or deleted, ensuring the integrity of the data.
- Security: Blockchain uses cryptographic algorithms to secure data, making it highly resistant to hacks and unauthorized access.

- Transparency: Transactions on the blockchain are transparent and visible to all participants, promoting trust and accountability.
- Efficiency: Blockchain can reduce transaction time and costs by eliminating intermediaries and automating processes.
- Traceability: Blockchain enables the tracking and tracing of assets and transactions, providing a clear audit trail.
- Interoperability: Blockchain can be integrated with other systems and technologies, enabling seamless data exchange and collaboration.
-

2.1.1. Structure of Blockchain

Blockchain mainly consists of five components, which are nodes, ledgers, blocks, nones, and hash codes. We will explain the components of the blockchain as shown below:

- NODES

  Nodes can be computers or servers, and they are used to store the transactions of the blockchain. All nodes on the blockchain are connected to each other and share the same ledger, so if any change occurs, it will be immediately detected [11]. Nodes are divided into full nodes and light nodes. Full nodes usually have more memory than light nodes. Stores a complete copy of the ledger and has the ability to add new blocks to the system.

- LEDGER

  Ledgers can be considered as the database of the blockchain. There are three types of ledgers which are public ledgers, distributed ledgers, and decentralized ledgers. Public ledgers are accessible to everyone while it is open to all the users of the network. Transactions in public ledgers can only be done after authorizing the identity of the node that will perform the transaction. Distributed ledgers enable all participants to have a copy of the ledger, but only specific nodes have the ability to add blocks and verify transactions over the network. A decentralized ledger doesn't require any type of trust between the participants,

enables accessing real-time data from the ledger, and reduces the dependence on specific authorities to manage the network.

- BLOCK

  Blocks can be considered as the backbone of the blockchain. Blocks contain the data of multiple transactions that occur over the network. When a block is filled with transactions, it will be linked with the previous block by storing the previous hash code inside it. Every block consists of two sections. Header, which contains the hash code of the previous block, timestamp, nonce, and Merkle root of the block. Transaction section, which contains a set of transactions stored in the block. When a block is created, its data are verified by the network, and then it is added to the blockchain.

- HASH CODE

  Hash code is the way blockchain uses to guarantee the security of the network. Hash codes ensure that no one can modify the transaction data stored inside the block. Blocks can't be added to the blockchain unless they are provided with the hashcode.

- NONCE

  It is a 32-bit number randomly used in cryptographic communication while creating the block. It can be used only once on the blockchain. When a nonce is created, it is added to the hashed block and then rehashed again with the block. The advantage of nonces is that it makes the transactions secure while verifying the transactions along with other data in blocks.

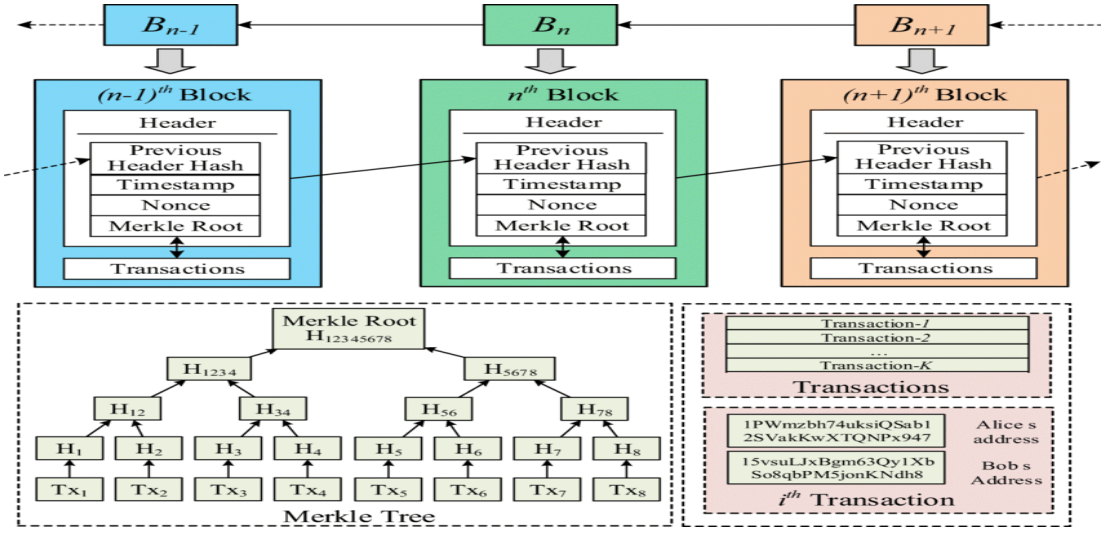

Figure 2.1. Blockchain architecture [12].

2.1.2. Consensus Algorithms

The consensus algorithm is an agreement among blockchain participants that is responsible for ensuring the security and integrity of the network [13]. There are many consensus mechanisms used in different blockchain networks, including Proof of Stake (POS) and Proof of Work (POW). Each consensus algorithm has its own advantages and disadvantages in terms of security, scalability, and energy efficiency, and the choice of algorithm depends on the specific use case and requirements of the blockchain network [13]. While we will use the Ethereum network, we will use POS as a consensus protocol in our research.

Table 2.1. Comparison between Consensus Protocols.

Consensus Mechanisim	Needed Resource	Implementation	Reward
POW	High computational power	Bitcoin	✓
POS	Stake	Ethereum	×
POSpace	High memory	Permacoin	✓
POI	Node significance	NEM	✓
Minimum Block Hash	No resources needed	Bitcoin extension	✓
PBFT	No resources needed	Hyperledger	×

2.1.3. Generations of Blockchain

Blockchain technology can be broadly classified into four generations [14]. We will provide details about each generation in the next sub-sections.

- FIRST-GENERATION BLOCKCHAIN

  Satoshi NAKAMOTO was the first one who converted the theoretical consept of blockchain into a real technology. The first generation introduced the the consept of Cryptocurrency which aims to create a Peer-To-Peer electronic cash system that performs the transactions without the need to a third trusted party like banks.The first generation is based on the proof-of-work (PoW) consensus algorithm. PoW requires computational work to be done to validate transactions and add them to the blockchain [15].
- SECOND-GENERATION BLOCKCHAIN

  The second generation of blockchain technology, which emerged in 2013, introduced the concepts of smart contracts and decentralized applications (DApps). Smart contracts are self-executing scripts depending on a predefined conditions.Smart contracts afford new use cases to use the blockchain network by enabling users to perform more complicated transactions in addition to the Cryptocurrency transactions [15]. The most popular second-generation blockchain is Ethereum, which uses the proof-of-stake (PoS) consensus algorithm.
- THIRD-GENERATION BLCKCHAIN

  The third generation of blockchain, also known as the enterprise blockchain, aims to make blockchain technology more practical for real-world applications. It is designed for business use cases. It study the integration of blockchain technology with other technologies like Internet Of Things (IOT) and the Artificial Intelligence (AI). It features higher scalability, better privacy, and interoperability between different blockchain networks. Enterprise blockchain platforms include Hyperledger Fabric, Corda, and Quorum [14 , 15].
- FOURTH-GENERATION BLOCKCHAIN

the fourth generation of blockchain applications can be divided into three verticals which are Metaverse, Web 3.0, and industry 4.0. The consept of the fourth generation aims to close the gaps in the previous generations. It aim to achieve high transaction throughput, improved privacy models, and reduced energy consumption of executing smart contracts [15].

2.2. NFT TECHNOLOGY

NFTs are the non-fungible part of the Ethereum blockchain network, which has its unique value. NFTs are unique digital assets that contain unique identification address and metadata stored over the blockchain network. NFTs are programmed due to predefined standards like ERC-721, which was introduced in 2018 as the first standard that supports NFTs [16]. NFTs can be used to represent both physical and digital assets. The process of representing physical assets in NFT is called tokenization. Tokenized assets can be divided into two categories:

Fungible, which are identical and can be replaced with each other's [17], and non-fungible, which are different in type and value, so they can't be replaced with other non-fungible tokens [18]. Most of NFT's characteristics are derived from blockchain technology, which is the basis of NFT.

While NFT depends mainly on blockchain, it has become the core component of NFT [19]. Each NFT is minted through a smart contract that has a unique token contract address stored in the minted NFT. NFTs also store the unique address of the creator, which makes it possible to identify the creators of NFTs. Each token has a unique NFT ID to distinguish it from other NFTs. All NFTs also have metadata, which is the actual content of the NFT. It can be media, pictures, art, or anything that can be stored. While metadata may has huge volume, which leads to high expenses to store it over blockchain, it is stored off-chain, and then a link to it is hashed and stored over blockchain. Storing the metadata of the NFT off-chain, hashing its link, and attaching it to the NFT is an effective way to reduce the cost of minting NFT for educational assets. All minted tokens store the transaction history, which enables the viewer to follow the ownership of the NFT until the creator[18 , 19].

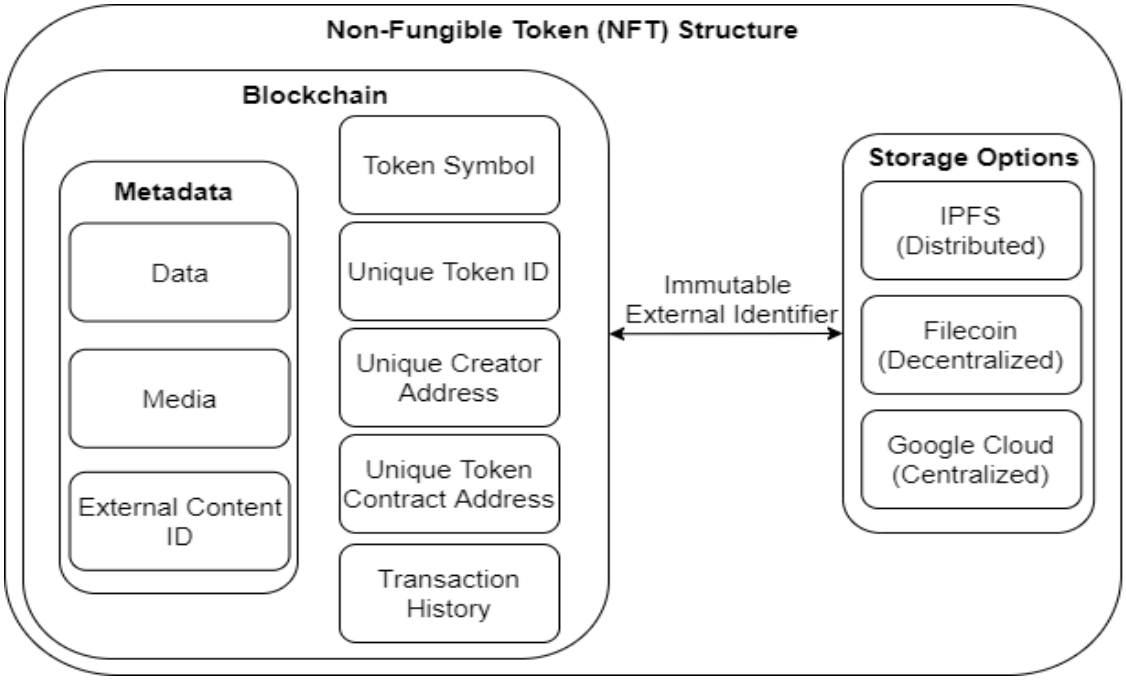

Figure 2.2. Components of NFT [18].

2.2.1. Advantages of Nft Technology

NFT depends on blockchain technology, which uses a distributed network to store data. Data and transactions among blockchains are tamper-resistant, which means that when a transaction is added to the chain, it cannot be deleted or modified [5 , 20]. There are many advantages to using NFT, and some of them are provided as follows:

- Proof of ownership: NFT assigns the ownership of an asset to a specific owner. When an NFT is registered over the blockchain, it cannot be modified or deleted, enabling owners to guarantee their ownership of assets.
- Easy ownership transformation: Owners of an asset can easily transfer ownership of their NFTs over the blockchain network.
- Authenticity: When an NFT is minted, it is recorded over the blockchain network with a unique contract address, which enables owners to ensure their NFTs while they are recorded over the blockchain.

2.2.2. Tokens Protocols and Standards

token protocols and standards provides guidelines for developers, users, and platforms for creatinig tokens interacting with it over blockchain platforms. standards aim to enhance the overall functionality of tokenized ecosystems by specifying how to create, transfer, and manage the minted tokens. In the sub-sections below we will discuss the standards of tokens produced for Ethereum network and specify which standard we will use and why.

- ERC-20

  ERC-20 is a standard proposed by Fabian Vogelsteller in November 2015 for fungible tokens (FTs). The standard assumes tokens to be exactly the same in type and value. ERC-20 can be used to represent a vote in elections, a stake in a market, a copy of a book, and many other applications. The standard provides basic functionality to transfer tokens and allow tokens to be approved, which enables third parties to spend the tokens [21].
- ERC-721

  ERC-721 was proposed by William Entriken, Dieter Shirley, Jacob Evans, and Nastassia Sachs in January 2018. The ERC-721 is an NFT standard that assumes every token is unique in terms of type and value from the other tokens in the same smart contract due to its rarity, quantity, and age. ERC-721 was first used in the Cryptokitties project, which is a game that enables players to buy, sell, and breed crypto cats as NFTs. For any ERC-721 contract, the contract address and uint256 token ID must be unique. The ERC-721 standard provides basic functions to transfer tokens from one account to another, get the current token balance of an account, and identify the owner of a token [16].
- ERC-1155

  ERC-1155 was proposed by Witek Radomski, Andrew Cooke, Philippe Castonguay, James Therien, Eric Binet, and Ronan Sandford in July 2018. The standard aims to enable smart contracts to represent and control both FTs and NFTs at the same time. While ERC-20 and ERC-721 require a separate contract to be deployed for each token type, ERC-1155 enables smart contracts to check the balance of many owners, transfer multiple token types at once, and approve

a third party to control and manage all tokens in a single contract, which reduces the transaction cost [22]. In the table below we provide a comparison table among the standards of tokens in terms of supporting FT, supporting NFTs, and support multiple token types in single smart contract.

Table 2.2. Comparison between token's standards

Standard	Year	Support FT	Support NFT	Support Multiple Token Types In Single SC
ERC-20	2015	✓	×	×
ERC-721	2018	×	✓	×
ERC-1155	2018	✓	✓	✓

Due to our needs for implementing the assets and while the cost of deploying NFTs over ERC-1155 standard is higher than ERC-721 we decided to use ERC-721 in our proposed model.

2.3. METAVERSE

In recent years, there has been a surge in the release of Metaverse-related applications by entertainment and social networking companies, including Meta (formerly Facebook) [23]. The Metaverse represents a computer-generated environment that is linked to the physical world. It is considered the next generation of the internet, after the web and mobile internet generations, due to its ability to provide an alternative life for users within a virtual realm.

The Metaverse primarily refers to the virtual reality (VR) space where users can interact with each other through computer-generated environments in real time. VR technology allows users to replace their perception of the real world with digitally produced environments using software and headgear devices. Additionally, augmented reality (AR) technology enables users to engage with a mix of digital and physical worlds, by using techniques such as object detection, plane detection, and

motion tracking to recognize real-world objects. Metaverse enables users to grant access through VR headsets, smartphones, and computers.

The Metaverse offers users a broad spectrum of activities, ranging from work and gaming to learning, all within a virtual environment. Particularly after the advent of the COVID-19 pandemic, the utilization of the Metaverse for educational purposes has gained significant traction. Its ability to facilitate virtual classes for teachers and students has made it an attractive solution for remote learning. However, the growing use of the Metaverse in education necessitates the protection of educational content created by educational institutions and the preservation of intellectual property rights for content creators.

Currently, the storage infrastructure employed in the Metaverse depends on centralized servers, rendering it vulnerable to attacks that may compromise the rights of content creators. Consequently, there is a pressing need to store Metaverse data in a decentralized environment to ensure the integrity and security of the content. To address this challenge, we aim to merge blockchain technology, non-fungible tokens (NFTs), and the Metaverse, thereby providing a platform for the creation of secure educational content and safeguarding the intellectual property rights of its creators.

By combining these technologies, we aim to establish a robust and transparent ecosystem that enhances the educational experience within the Metaverse while ensuring the authenticity and protection of educational assets. This integration will enable users to explore and engage in educational activities with confidence, leveraging the benefits of blockchain's immutable nature, NFTs' uniqueness and ownership verification, and the immersive capabilities of the Metaverse.

2.4. BLOCKCHAIN-BASED SOLUTIONS FOR MANAGING THE EDUCATIONAL ASSETS

This part will explore how NFT can solve some of the issues faced in managing educational assets. We will give a brief explanation of each of the previous works and compare them with our proposed model.

2.4.1. Nftcert

NFTCert is a framework to issue NFT-based certificates instead of paper format. Due to this framework, companies and international universities can authenticate the educational certificates of students, depending on NFT. The framework uses private blockchain architecture, which forces educational institutes and universities to get permission to join the network before accessing the certification services [24]. The framewrok uses an online payment gateway as a payment method for certification services instead of cryptocurrency. NFTCERT has it's own digital certificate data format which is mandatory to follow to issue new certificates. it adds the signature of student, student's personal information, and certificate information to issue the certificate and transfer it to student's wallet.

Due to NFTCERT only members of their private blockchain have the ability to access their certification services. While EIs and companies can't verify the ownership of certificates unless they ask to join the private blockchain of the framework the accessing process take long time because of the need to apply to join the network before.

2.4.2. Blockchain For Education

Blockchain for education is an educational certificate-issuing perspective based on blockchain. It guarantees the protection and verification of certificates without the need to return them to the issuing authority. The model uses the InterPlanetary File System to store the profiles of certification authorities. The model allows to identify the identity of authorities who can issue certificates, allow the authorities to issue the certificate, and allow third parties to verify the issued certificates [25].

The model contains users, educational institutes, and certification authorities. Certification authorities import the data of student's exams from the system, create the digital certificate, and revokate the crtificate as well as confirming the validity of the issued certificates. While the ability of the framework to enable other nodes on the blockchain to issue certificates distinct it from NFTCERT, it still doesn’t have the

ability to issue other types of educational content and need the permission to access the network which affect the users base of the model.

2.4.3. Decentralized Educational Assets Storage

In this paper, they integrate the blockchain into the educational sector by proposeing a decentralized educational model to authenticate the educational certificates of students and provide blockchain-based academic certification system for the issuers. The model validates the certificates of students through blockchain and IPFS (interplanetary file system). The model use IPFS to securely store educational assets in form of non-fungible tokens instead of storing it directly to the blockchain network. Ethereum network is used to store the Content Identifier of retrieved from IPFS. The integration between blockchain and IPFS enable the model to reduce storage fees for minting educational certificates over blockchain [26].

Although the model used Ganache as a public blockchain and used IPFS to reduce the storage fees of the proposed model, it doesn't give support for other educational contents like videos, pictures, PDF, and other multimedia. The model doesn't give details about the issuers of the certificates which allow any node to be able to perform minting process while there is no mechanism that identify the nodes which are able to mint new certificates and that is covered by the pre-registration stage in our proposed model.

2.4.4. Blockcerts

Blockcerts is an open standard that was designed to work over any blockchain network. The standard allows to verify and issue official records like course licenses and academic credentials. The standard allows the educational institutes to issue educational certificates based on the credentials given by the students and make it easy for certificate owners to reveal their certificates only to the desired third parties. In addition to issuing certificates for users, issuers of the certificates also have the ability to revoke certificates from recipients. The credentials of certificates are hashed into

the blockchain and then sent to the recipient, who can give it to the verifier, who will check the blockchain to verify the certificate [27].

The standard only focusses on issuing licenses based on the credentials provided by the applicants and doesn't pay attention to the management of other educational assets over the network. The standard also doesn't take into considration the potential of unofficial Educational institutes to use it to issue certificates over blockchain which enable any user to create their own certificate without an authority that organize issuers identity.

2.4.5. Boll

Blockchain Of Learning Logs (BOLL) provides an implementation of a private blockchain platform for keeping track of learning achievements like certificates and transcripts. The framework has two main user groups, which are institutions and learners. At least one institution must serve as the host of the blockchain. The framework enables connecting learning records among different institutions, which will help learners to move their diplomas and records from one institute to another in a secure and verifiable way [28].

The model is only used to store the learning achievements and doesn't cover the whole educational contents that can be produced by the EI like videos, pictures, and PDF files. Also the model has difficulties to deal with logs from different systems while there is no standard to unify logs that came from different learning logs formats . The model also use private blockchain which affect the number of Educational institutes and the stdents that can use the model negatively.

2.4.6. TTECCDU

The model provides an authorization mechanism for large scale distributed systems over Ethereum network using Ganach for deployment. The approach depends on role based access control model which enables it's owner to use several smart contracts to grant or deny the access to the asset. The model provided five different access control

mechanisms over Ethereum network which are trust based, cost based, temporal based, cardinality based, and usage-based access control contracts. The model doesn't talk specifically about the educational content, but provides a good access control mechanism choices to follow for other works [29].

Table 2.3. Comparison of existing blockchain-based management systems.

Method	Year	Objective	Support educational certificates	Cover all educational content	Privacy of personal info	Applicable with Metaverse
Blockcerts	2016	Allow educational institutes to issue certificates based on student's credentials.	✓	×	✓	×
BOLL	2022	enables connecting certificates and transcripts among different institutions	✓	×	✓	×
Blockchain for education	2018	Guarantee the protection and verification of certificates without the need for a third party.	✓	×	✓	×
NFTCERT	2021	Issuing NFT-based certificates instead of paper format.	✓	×	✓	×
Decentralized storage of educational assets using Blockchain technology	2022	Authenticate student's certificate using blockchain.	✓	×	✓	×
TTECCDU: a blockchain-based approach for expressive authorization management	2023	Proposes an authorization framework that comprises of multi access control models	×	✓	✓	×
Our proposed model	2023	Protect the educational assets from fraud operations and preserve the intellectual property.	✓	✓	✓	✓

PART3

PROPOSED ASSET MANAGEMENT SCHEME

In this section, full details of our model to manage the educational assets will be introduced, starting with the architecture of the model, explaining the pre-registration and asset approval stages, before moving on to asset management schemes, which consist of the registration phase, the asset minting phase, and the assigning roles phase. Lastly, we introduced our time-based access control mechanism. The chapter ends with a conclusion to the work.

3.1. MODEL ARCHITECTURE

Our model aims to manage educational assets in the Metaverse with a system architecture that consists of five main actors as follows:

- User: This can be an author or a student applying to the educational institute to register or grant access to an asset.
- Trusted Authority: Receives authorization requests from the educational institutes and mints a yearly NFT that contains the public keys of the authorized Educational Institutes in its metadata, which will allow users to check the authenticity of the Educational Institutes. In our model we consider the Minstry Of Education as the trusted authority for our model.
- Educational Institute (EI): Which apply to Ministry of Education for joining the model. EIs handle the requests from users, and have the authority to mint new tokens.
- IPFS: Which provide a decentralized off-chain storage service for data that enable the minter to store the metadata of NFT without the need to pay huge cost by providing a link stored over blockchain instead of the actual metadata.

- Blockchain: Ethereum blockchain which is used to hold the information of both the EI and users, it is also used to store the tokens, and managing the accessibility of the tokens. Using public blockchain makes it easier for both EI and users to use the model and get benefits from it's features.

The architecture of our model is given as in figure below:

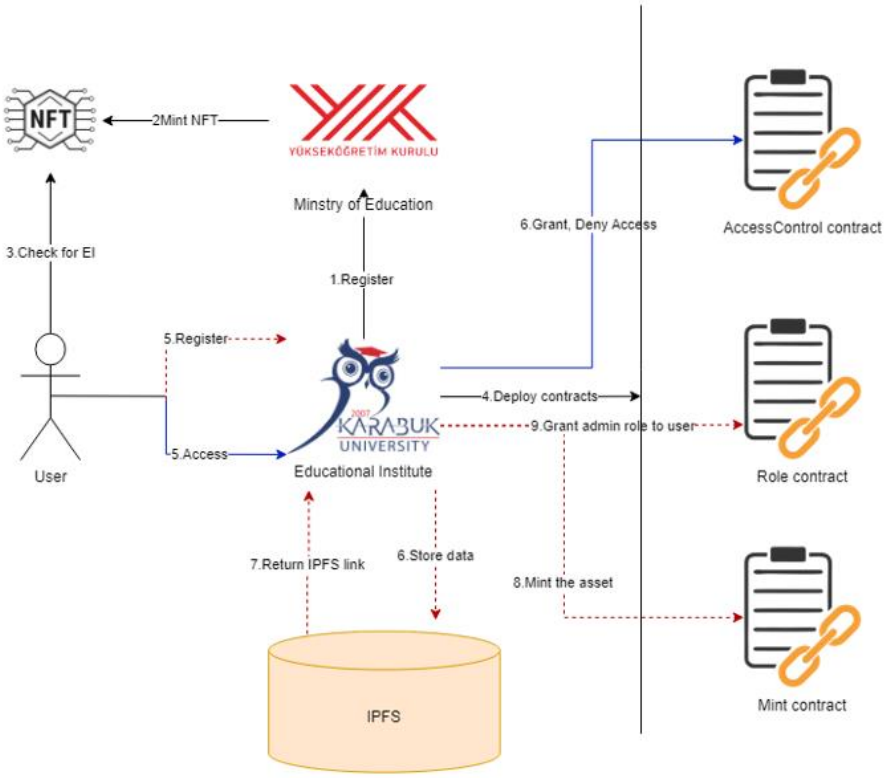

Figure 3.1. Architecture of our proposed model.

Model architecture can be divided into two stages, which are pre-registration and asset registration and approval stages.

3.1.1. Pre-Registration Process

It starts with the application of EIs to the Ministry of Education (MOE) to get authorization for using the model. MOE collects the public keys of the accepted EI applications and stores them in a file on IPFS. MOE will then use the Mint contract to mint the NFT over the Ethereum network and define the metadata of the minted NFT

as the CID received from the IPFS server. Applicants who want to register or access an asset will be able to check the MOE token and choose the appropriate EI to apply for.

In the figure below, we show the pre-registration stage of our model architecture.

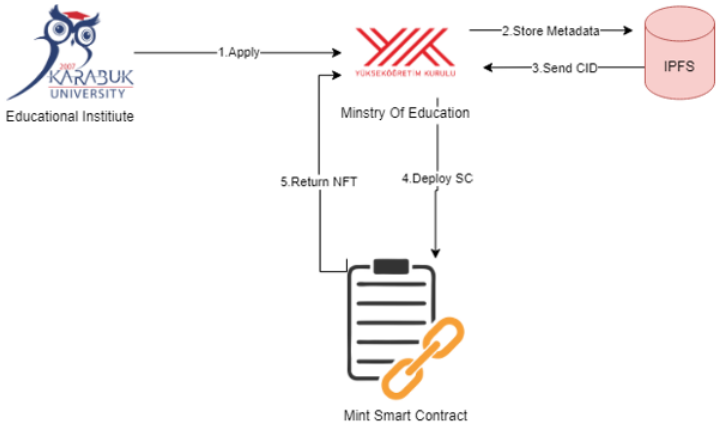

Figure 3.2. Pre-registration stage of our model architecture.

The process starts when MOE mint an NFT that contains the public keys of all authorized EIs and grant access for all of users among the network. Users who want to apply to an EI will be able to check for an appropriate institute to apply for. When applicant apply for registering or accessing a content, the EI evaluate the application and deploy SC of the system. If the evaluated request of registering is accepted, EI mint the metadata of the content as an NFT and grant ADMIN_ROLE to the applicants to enable them to add and remove viewers of the content using the access control mechanism. When applicant apply for accessing a resource, the application is evaluated by the EI and in case of acceptance, the EI uses Access Control contract to enable the applicant to view the asset for a limited period of time.

3.1.2. Asset Registration and Approve Process

The metaverse user starts the stage by checking the token minted by the MOE in the first stage and checking for an appropriate EI to apply for. The EI deploys the Mint, Role, and AccessControl smart contracts which was given by the MOE and starts receiving applications from users. Mint contract is responsible to mint new assets and

transfer it to user's wallet. Role contract is used to assign admins as well as revoke admins for the minted assets using grant role and revoke role functions. Role contract also can add minters to the system which have the ability to mint new tokens using Mint contract. Access control contract can be used by the admins of the assets to add or remove viewers of their content. The contract has two functions which are Addviewer and Removeviewer used to enable or deny the access to minted assets. When a user applies for registration or accessing an asset, the EI evaluates the application and starts the next step up to the application. There are two use cases in the asset registration and approval stages, which are user requests to access an asset or user requests to register an asset.

In the figure below, we show the second stage of our proposed model architecture.

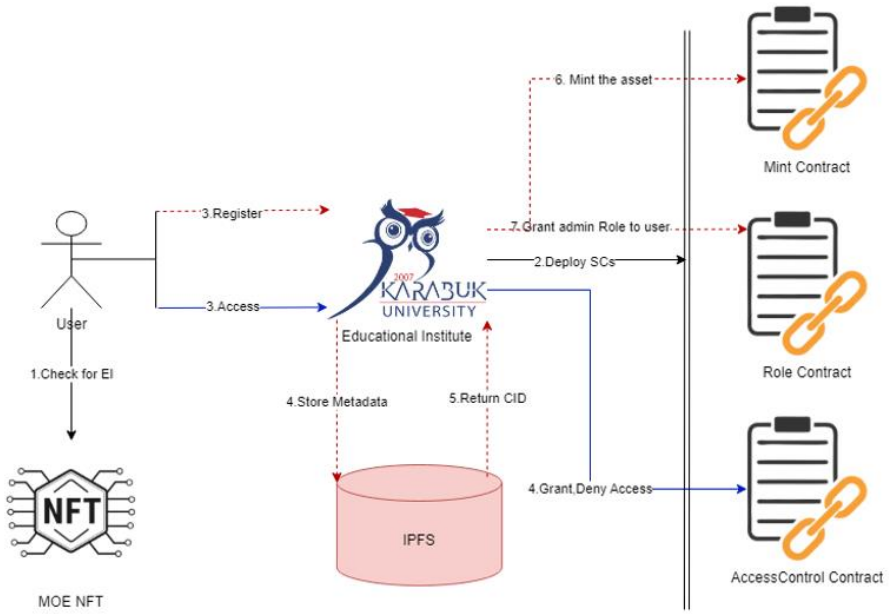

Figure 3.3. Asset registration and approval stage.

- USER REQUESTS TO REGISTER AN ASSET

  The Educational Institute (EI) will deploy the smart contract of the model and receive the applications for registration and the assets from the users. EI will evaluate the request of the user, and if it is accepted, EI will start the minting process of the asset. To add an educational asset as an NFT, the EI stores the desired asset's file to IPFS and gets a CID of the asset, which will be attached to

the JSON file and contain the information of the asset owner and considered as a digital educational asset that proves the ownership of that asset. Then, while EI has MINTER_ROLE, which enables admins and viewers to add the minted token, The EI will grant an admin role to the applicant using Role contract to enable them to add and remove viewers of their assets.

- USER REQUESTS TO ACCESS AN ASSET
  When the EI receives a request for viewing an educational asset, it evaluates the application, and if the application is accepted, the address of the applicant will be added as a viewer for the minted token for a limited time, so the applicant will be able to access the metadata of the minted token.

3.2. ASSET MANAGEMENT SCHEME

In this section, we will talk about the phases of our asset management scheme, which are divided into 3 phases: registration phase, asset minting phase, and assigning roles phase.

3.2.1. Registration Phase

The registration process is done by Role contract, which is used to specify the user's roles on the model. Role contract has mainly three roles, which are DEFAULT_ADMIN_ROLE, MINTER_ROLE, and ADMIN_ROLE. Role contract has two functions, which are grantRole() and revokeRole. Both functions take the address of the assigned user and a bytes32 variable of the desired role to be given or revoked to a user. The roles given by this contract specify the responsibilities of users. DEFAULT_ADMIN_ROLE is given automatically for the address of the person who deployed the contract and enables its holder to run all functions in the smart contracts. MINTER_ROLE enables the holder of it to use the mint contract for minting new NFTs. ADMIN_ROLE can be given to enable its holder to use the AccessControl contract, which enables adding and removing viewers of a specific token but doesn't have the authority to use the Mint contract to add new NFTs to the network.

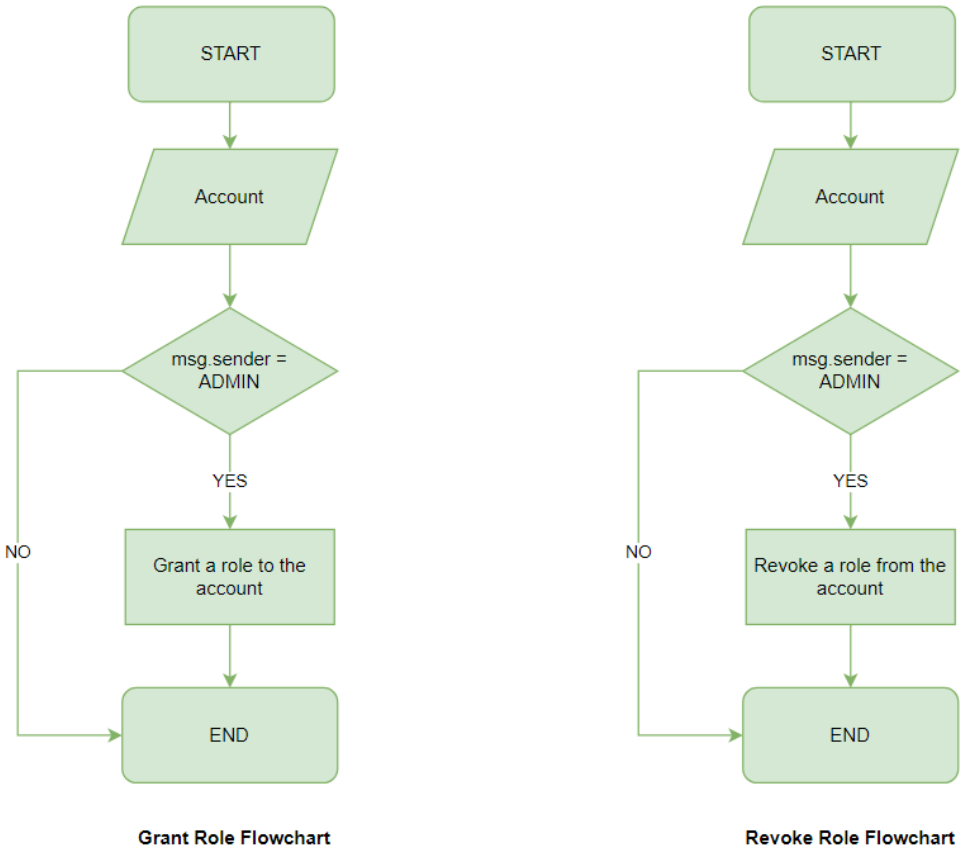

Figure 3.4. Flowchart of GrantRole and RevokeRole functions.

3.2.2. Asset Minting Phase

The asset minting phase is done using a Mint contract by addresses who granted MINTER_ROLE by entering the name and symbol of the token for deploying the contract. The mint contract contains a mint function that needs three parameters to be executed, which are to, tokenID, and metadata, as explained below:

To: is the address where the token will be transferred.

$TokenId$: which is the ID that will be given for the token.

$Metadata$: which is a string filled with the IPFS link of the JSON file that stores the multimedia of the educational asset.

After filling the parameters the SC checks the address if has MINTER_ROLE or not. If the address has MINTER_ROLE the metadata associated to the token is added and token is minted and transferred to the owner's address.

The figure below show the flowchart of mint contract used for our model.

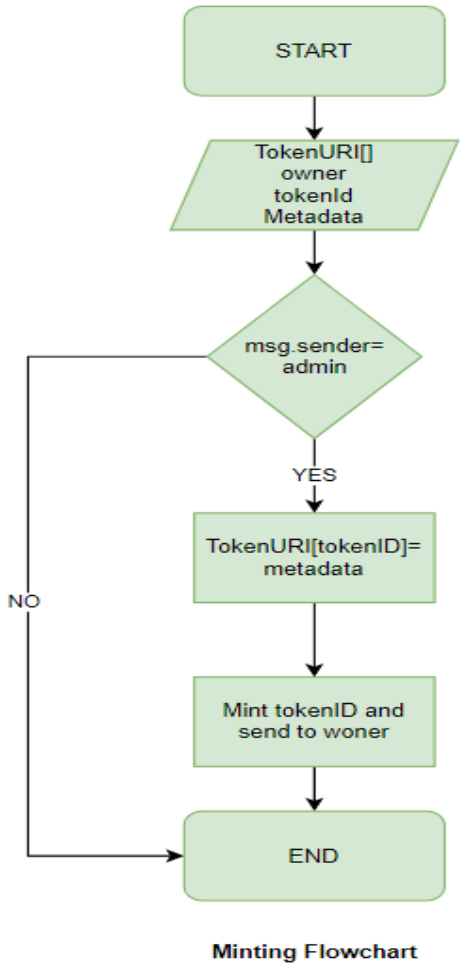

Figure 3.5. Flowchart of mint function.

We also provided a sequence diagram for phase in our proposed model, as in the figure below where the author sends the data to the EI as an application. After that the educational institute evaluates the application and store the desired asset to IPFS server then receives CID of the stored asset. The EI then start the minting process to represent the asset as an NFT and the metadata of the NFT will be the CID received from IPFS. Finally the minted asset is transferred to the author's wallet.

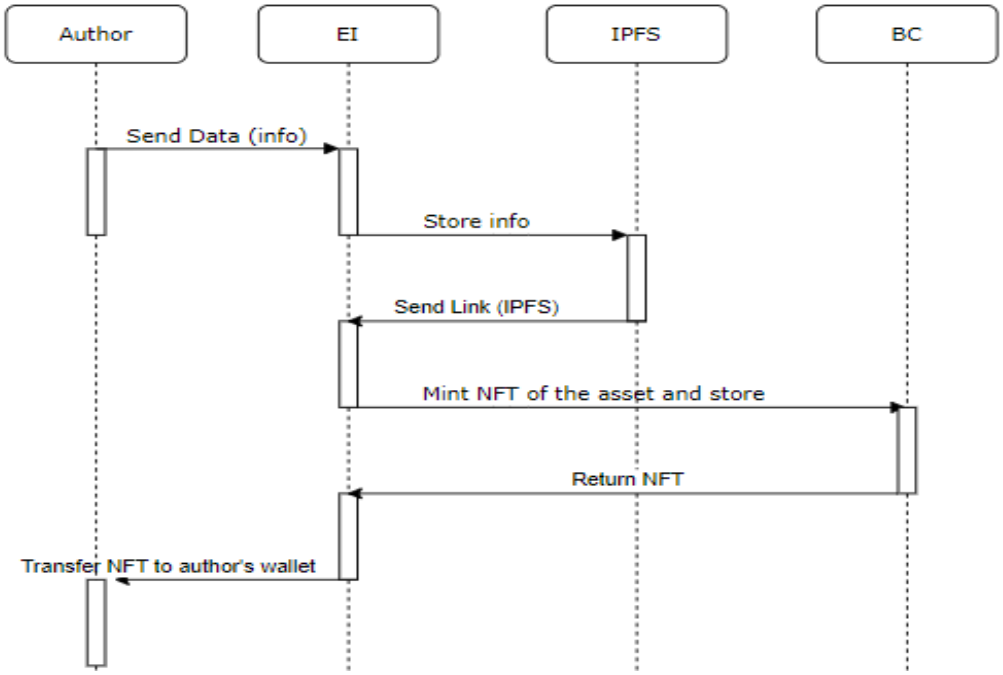

Figure 3.6. Sequence diagram of minting an asset.

3.3. TIME-BASED ACCESS CONTROL SCHEME

We will introduce the scheme of our access control to the minted asset, introducing who has the ability to access the asset and adding or removing viewers of the minted asset. The access control process is done by the AccessControl contract, which is a time-based access control contract that manage the admins of tokens to specify who can grant the access to their tokens and the duration of access. Functions in the AccessControl contract can be executed by addresses granted ADMIN_ROLE using the grantRole() function in Role contract. The AccessControl contract manage the viewers of the tokens using two functions, which are AddViewer() and RemoveViewer(). AddViewer() takes three parameters, which are tokenID, the address of the desired viewer for a specific asset, which will be stored in a list that contains the addresses of viewers which we represent as Token Viewers[ ] [ ], and the period that the viewer is allowed to access the token per hour.when the user call AddViewer(), the function checks if the caller address has ADMIN_ROLE then if user is an Admin the function will convert the access period to seconds via multiplying it by 3600 and add the viewer address and the token Id to the Token viewer list and assign the value of it as the time of the transaction + the access period by seconds

which will enable the viewer to view the token for the time given. Addresses listed in the viewers list will be able to view the metadata of the token for a specific time. On the other hand, the RemoveViewer() function takes two parameters, which are tokenID and the address of the viewer. When the function is called it check if the caller has ADMIN_ROLE then if yes the address of the viewer will be removed from the Viewers list.

In the figure below, we show the Access control chart for our model.

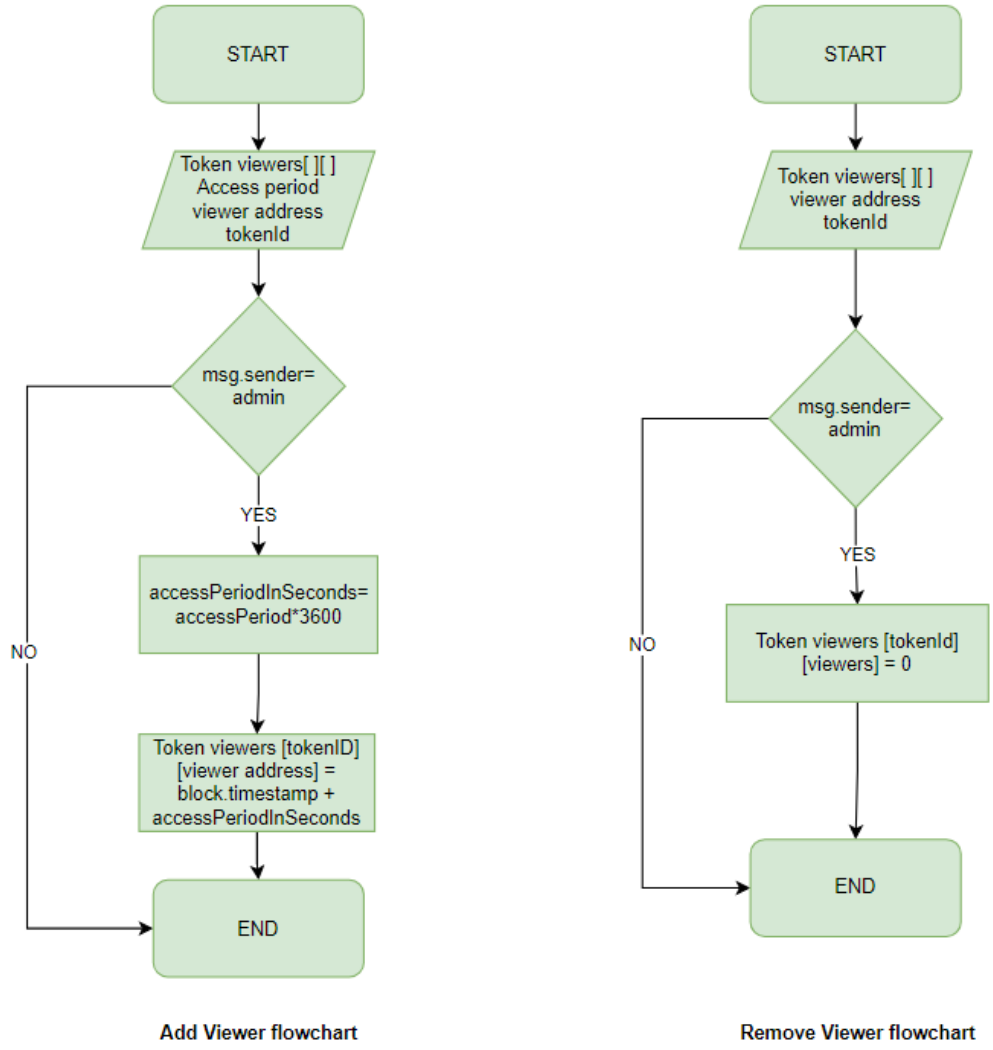

Figure 3.7. Flowchart of Add and Remove viewer functions.

PART 4

IMPLEMENTATION AND PERFORMANCE EVALUATION OF OUR PROPOSED MODEL

In this section, full details about the implementation of our model to manage the educational assets will be introduced. Tools used for development, stages of development, and smart contracts that are used for the minting and access control processes will be explained. A performance evaluation for our model will be done, and the validity of our model will be checked.

4.1. MODEL IMPLEMENTATION

Full details about the implementation of our model, tools used for development, stages of development, and smart contracts that are used for the minting and access control processes will be shown in this subsection.

4.1.1. Development Stages

This section will show the tools used to develop our model and the stages of development that followed.

we provide details about the used tools for development and a brief introduction about each of them. We used six tool to develop our model which are Goerli Ethereum network to deploy smart contracts over it, Solidity which is the programming language used to write SCs for Ethereum, Metamask which is the wallet that will show our balances and assets, IPFS which will store the data of the assets that will be minted, @openzeppelin which is a library that will be used to help building the SC, and Remix IDE which is the an open source web and desktop application that will be used to write

the SC. In the table below we show the tools used for developing provided with a brief explanation about each of them.

as will be shown in table.

Table 4.1. Tools of the model.

NAME	DESCRIPTION
Goerli Ethereum network	Goerli is a test network similar to Ethereum main network used to test smart contracts without the need to spend real ethers for deployment or executing transactions [30].
Solidity	Solidity is an object-oriented curly-bracket language which is influenced by Javascript and C++. It is used to write smart contracts for the Ethereum blockchain [31].
Metamask	MetaMask is one of the most popular Cryptocurrency wallets that aims to allow users to interact with decentralized applications over the Ethereum network [32].
IPFS	InterPlanetary File System (IPFS) is an open-source protocol for transferring and storing data in a decentralized environment. IPFS uses content Identifiers to identify each file stored in it. In our model, we will use IPFS to store the asset of and the JSON file of metadata, which will enable us to reduce the cost of minting the NFTs [33].
OpenZeppelin	Openzeppelin is an open-source framework used to enable developers to build secure smart contracts. It contains a variety of reusable smart contracts like Access control and token standards which we used in our proposed model [34].
Remix IDE	Remix IDE is an online development toolset for Ethereum network. It contains code editor, compiler, debugger, and many environments to develop and test the smart contracts [35].

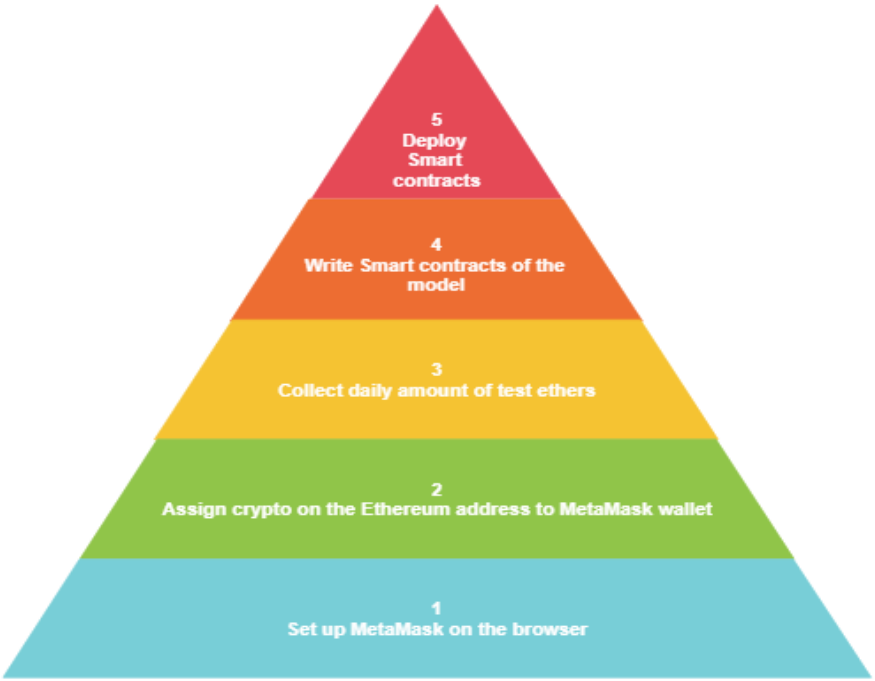

Figure 4.1. Stages of Development.

Stages of development were done as follows:

- Set up a metamask wallet on the browser which is one of the most popular digital wallets used for Ethereum network we will use it to interact with the smart contracts that we will use in our model.
- Assigning crypto on the Ethereum main network address to the metamask wallet.
- Collecting the daily amount of Goerli test ethers provided by Alchemy which gives 0.02 test Ethers per day in case of having 0.01 Ethers at least in the main Ethereum network address.
- Writing smart contracts needed to perform our model which are: Role contract, Mint contract, and Access control contract for the proposed model using solidity language and @openzeppelin library.
- Deploy the smart contracts on Remix IDE using Metamask as an injected provider and the Goerli test ethers that we collected from Alchemy provider.

4.1.2. Processes Of Our Proposed Model

Our scheme contains registration process, minting process, and access control process as will be explained in the sub-sections below.

4.1.2.1. Registration Process

This process is responsible for adding users to the system and assigning the appropriate role to them. The registration process is done by role contract, which import AccessControl contract from @openzeppelin library. Role contract contains grantRole() and revokeRole() functions to control adding and deleting users from the model. The deployer of the contract grant MINTER_ROLE, ADMIN_ROLE, and DEFAULT_ADMIN_ROLE directly as shown in the constructor. GrantRole() and RevokeRole() functions can only be executed by DEFAULT_ADMIN_ROLE who is the owner of the contract.

In the figure below, we provide the role contract of our proposed model.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/access/AccessControl.sol";
contract MyRole is AccessControl {
    bytes32 public constant MINTER_ROLE = keccak256("MINTER_ROLE");
    bytes32 public constant ADMIN_ROLE = keccak256("ADMIN_ROLE");
    constructor() {
        _setupRole(MINTER_ROLE, msg.sender);
        _setupRole(ADMIN_ROLE, msg.sender);
        _setupRole(DEFAULT_ADMIN_ROLE, msg.sender);
    }
    function grantRole(bytes32 role, address account) public virtual override onlyRole(DEFAULT_ADMIN_ROLE) {
        super.grantRole(role, account);
    }

    function revokeRole(bytes32 role, address account) public virtual override onlyRole(DEFAULT_ADMIN_ROLE) {
        super.revokeRole(role, account);
    }

}
```

Figure 4.2. Role contract.

4.1.2.2. Minting Process

This process is responsible for minting new assets and adding them to the network. The process can be achieved by users who have MINTER_ROLE. Mint contract import AccessControl and ERC-721 contracts which contains a set of methods that help for minting process from @openzeppelin library. The contract has a constructor that takes two parameters which are name and symbol. These parameters must be filled in by the deployer when deploying the contract. Mint function takes three parameters which are to,tokenId, and metadata. _mint function is an internal function provided by ERC-721 contract which will actually mints the token and assign it to the specified address (to) with the given tokenId.

If the new asset belongs to the EI, it is minted and transferred to the address of the EI. If the asset belongs to the user, it is minted and transferred to his or her address, and then Role contract is used to assign the admin role to the user of the asset. In the figure below, the minting contract for our model is shown.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC721/ERC721.sol";
import "./Role contract implementation.sol";

contract MyNFT is ERC721, AccessControl {
mapping(uint256 => string) public tokenURIs;
bytes32 public constant MINTER_ROLE = keccak256("MINTER_ROLE");
    constructor(string memory name, string memory symbol) ERC721(name, symbol) {

    }
    function mint(address to, uint256 tokenId, string memory metadata) public onlyRole(MINTER_ROLE) {
        _mint(to, tokenId);
        tokenURIs[tokenId] = metadata;
    }



    function supportsInterface(bytes4 interfaceId) public view virtual override(ERC721, AccessControl) returns (bool) {
        return super.supportsInterface(interfaceId);
    }
}
```

Figure 4.3. Mint contract.

4.1.2.3. Access Control Process

The access control process is responsible for adding and removing viewers of a specific asset. Access control contract is a time-based contract that has AddViewer() and RemoveViewer() functions to control who can view the metadata of the asset for a specific time. Access Control contract import ERC-721 and AccessControl contracts from @openzeppelin library. AddViewer(), and RemoveViewer() functions can only be executed by addresses which have ADMIN_ROLE. When adding a viewer to an asset the access period must be given by hours. The given value of the time is multiplied by 3600 to grant the number of seconds the asset can be accessed in. In the figure below, the time-based access control contract is shown.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
import "./Role contract implementation.sol";
import"./mintncontract implementation.sol"
    mapping(uint256 => mapping(address => uint256)) private _tokenViewers;
    mapping(uint256 => uint256) public tokenAccessPeriod;

contract MyNFT is ERC721, AccessControl {
    function supportsInterface(bytes4 interfaceId) public view virtual override(AccessControl) returns (bool) {
        return super.supportsInterface(interfaceId);
    }
    function addViewer(uint256 tokenId, address viewer, uint256 accessPeriodInHours) public onlyRole(DEFAULT_ADMIN_ROLE) {
        uint256 accessPeriodInSeconds = accessPeriodInHours * 1 hours; // convert hours to seconds
        _tokenViewers[tokenId][viewer] = block.timestamp + accessPeriodInSeconds;
    }

    function removeViewer(uint256 tokenId, address viewer) public onlyRole(DEFAULT_ADMIN_ROLE) {
        _tokenViewers[tokenId][viewer] = 0;
    }

}
```

Figure 4.4. Access control contract.

4.1.3. Full Model Experiment

We will cover a full experiment for our model, starting with contract creation, passing to assigning and revoking roles for users, adding the content to IPFS, minting new educational content, and ending with adding and removing viewers of a specific token. We started the deployment of our contract over Remix IDE [35]. After the compilation of the contract is done successfully, it asks for the name and symbol strings of the contract required for compilation. These parameters can be considered as a category for the assets that will be minted using this contract.

In the figure below, we show the deployment fields of the smart contract that will be deployed.

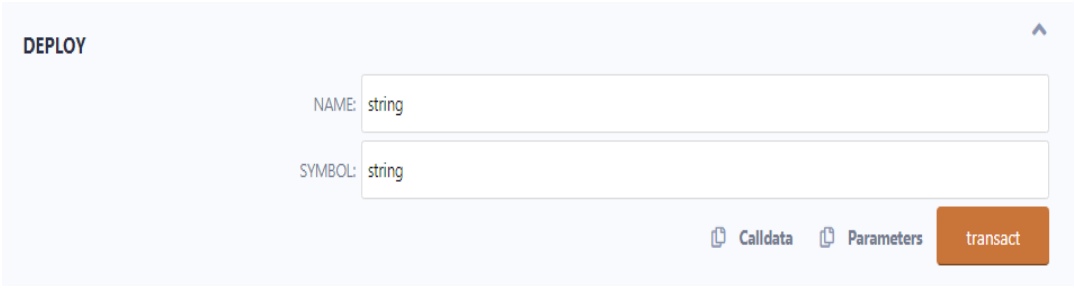

Figure 4.5. Fields required to deploy the smart contract.

In our demo, we filled the NAME as MY NFT and the SYMBOL as MNFT. When we press transact, we will get a notification from our Metamask that provide details about the estimated time and gas fee for the deployment process of the contract, as in the following figure.

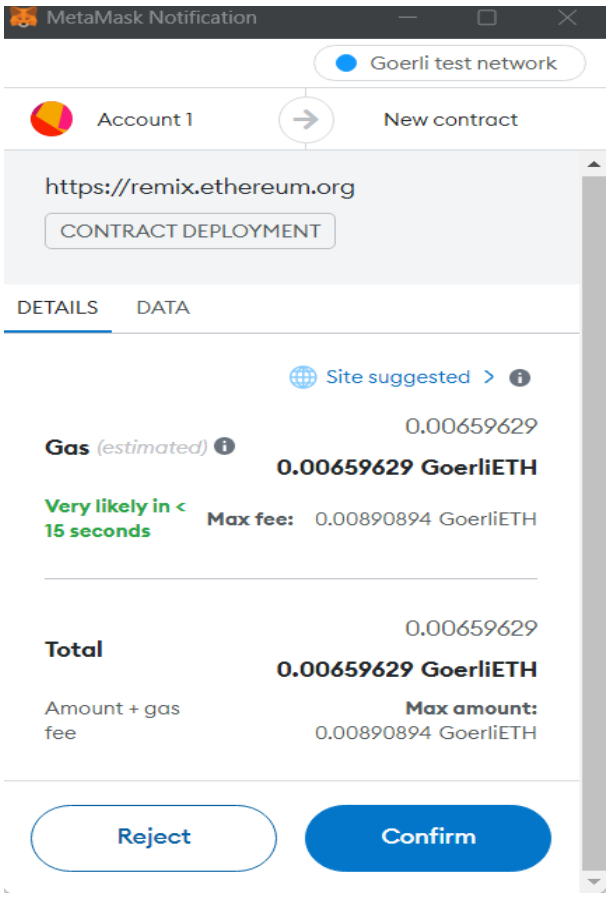

Figure 4.6. Metamask confirming notification.

After confirming the transaction, the deployment process will start, the contract creation will be done, and a transaction hash will be given to us. The transaction hash

helps us control the status of the contract over Etherscan, which is a block explorer that allows us to access the details of any pending or confirmed transaction over the Ethereum blockchain. Etherscan enables searching using address, transaction hash, and block ID. While we are using the Goerli test network, we will use Goerli Etherscan to control our transaction status.

In the figure below, we show the details of our contract deployment transaction over the Goerli test network using Goerli Etherscan.

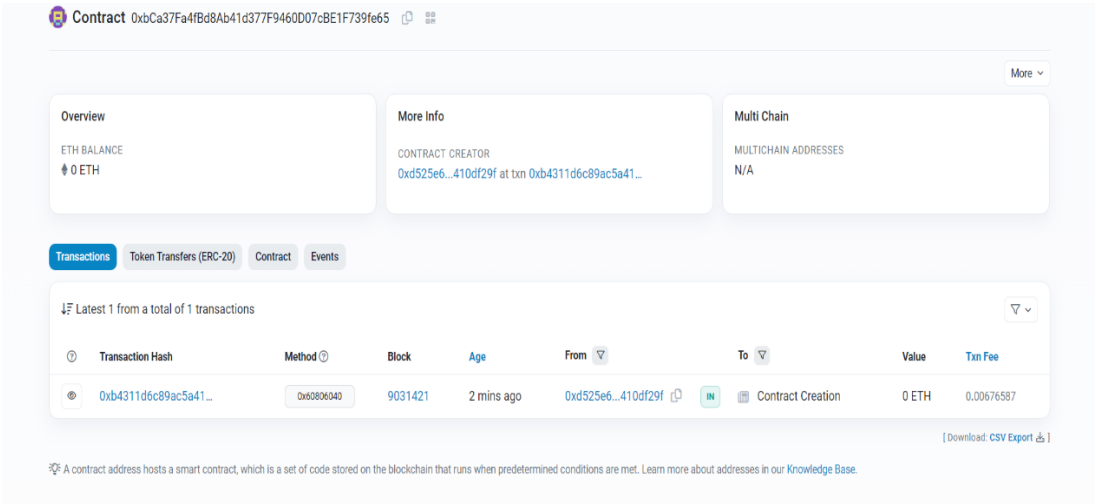

Figure 4.7. Contract creation on Goerli Etherscan.

Figure 4.8. The metadata JSON file stored in the IPFS server.

As shown above, the metadata of the asset that will be minted is represented as a JSON file that contains full details about the asset, like the institute name, authors, title, type, and date of issue is stored in IPFS. Each file stored in IPFS has a unique content identifier (CID), which ensures that the content in the minted NFT is the real content. In our model the real content is stored over IPFS and the CID of it will be attached to

the JSON file which is also stored in IPFS as we explained before. The CID of the JSON file will be stored in Blockchain as a metadata which will reduce the cost of storing the asset.

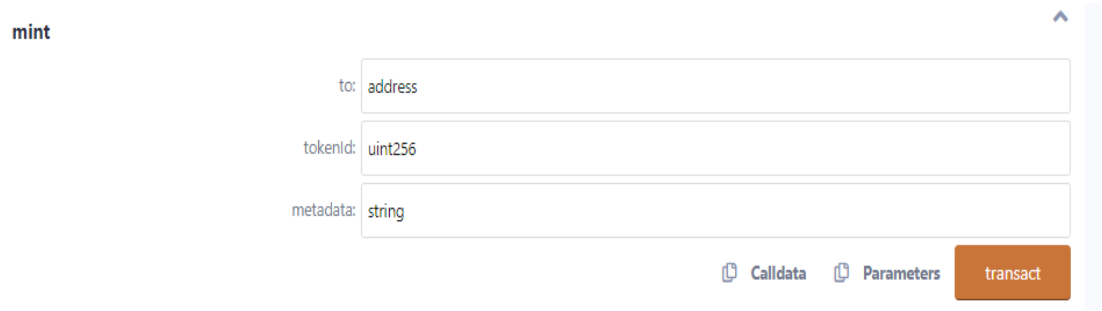

Figure 4.9. Minting an asset to a specific address.

As shown above, the mint function is to create a new NFT asset on the blockchain. The function enables the minting of a token with a unique ID and specific metadata, which is the IPFS link of the JSON file to an address. When the function is run, a notification from the Metamak wallet will appear to confirm or deny the transaction, and then it will be minted to the network and can be checked on Etherscan.

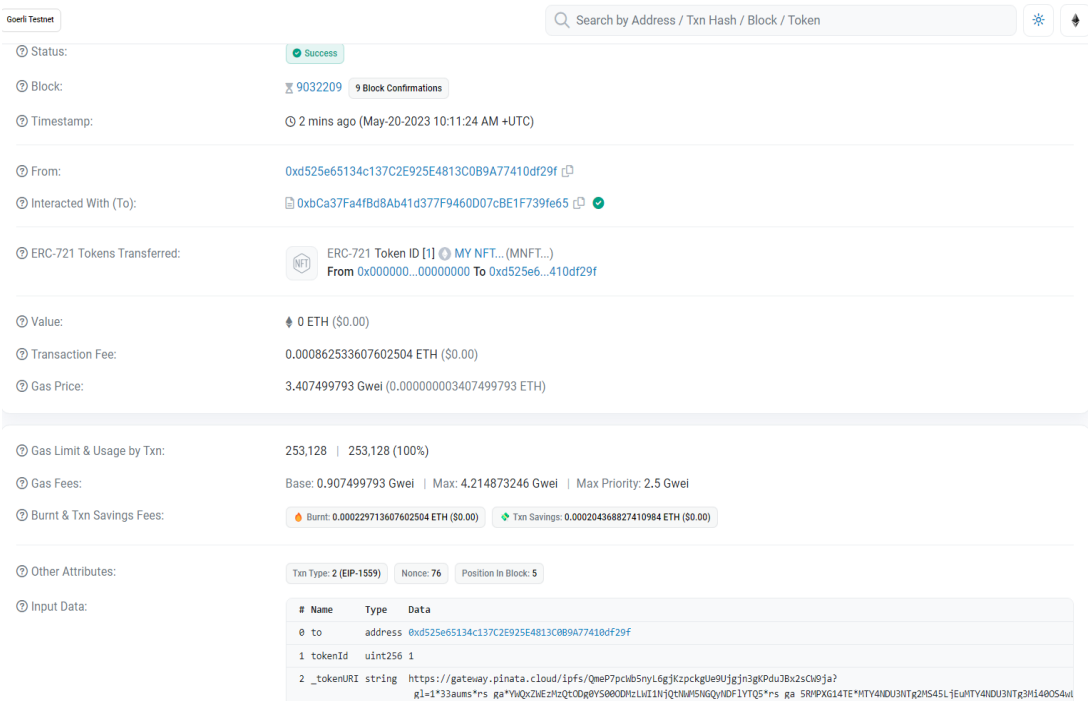

Figure 4.10. Etherscan page for the minted asset.

As shown above, the transaction is confirmed, and the NFT is minted to the desired address. We can also notice the stored data of the minted asset, which is the data we provided while minting the token.

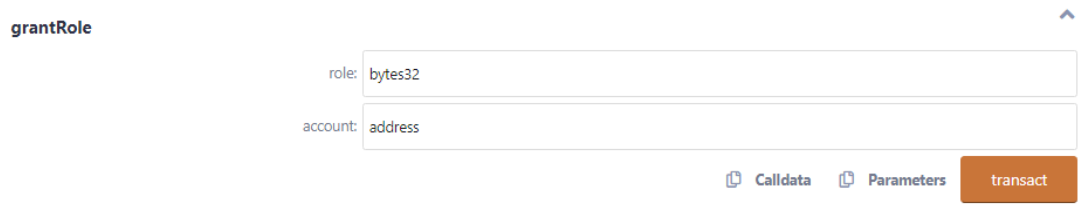

Figure 4.11. Assigning role to a specific address.

In figure 4. 9. grantRole function enables the owner of the contract to assign a role to a specific address. Three types of roles can be added, and the duty of the address is defined by the role given to it.

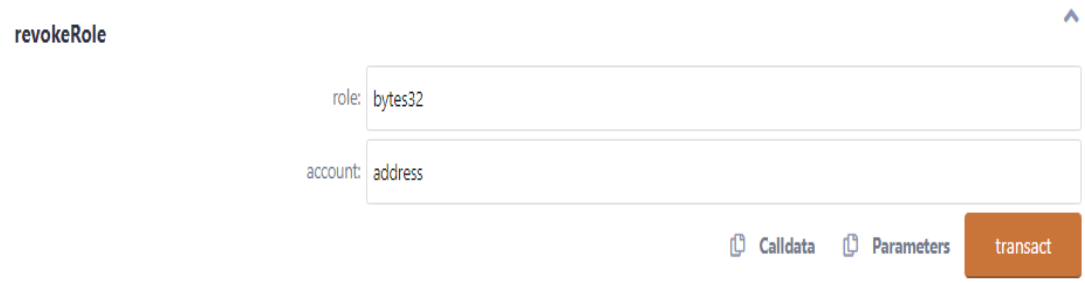

Figure 4.12. Revoking role from a specific address.

In the figure above, the RevokeRole function enables the owner of the contract to revoke a role. Three types of roles can be removed from an account, depending on the role value provided to the function.

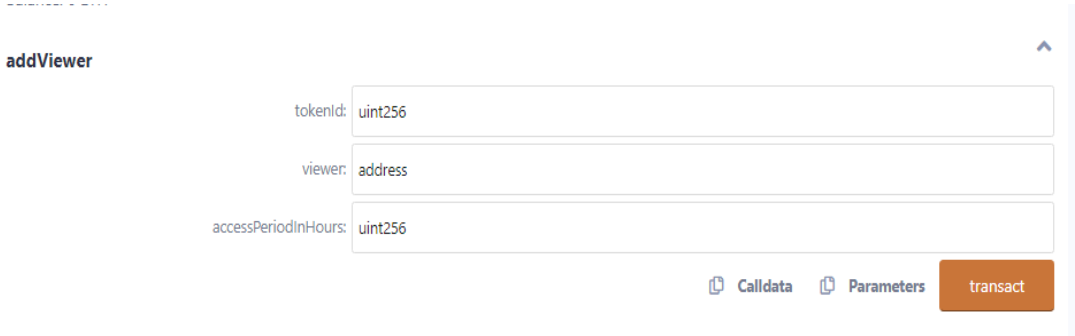

Figure 4.13. Adding viewer for a specific token.

In the figure above, the AddViewer function enables admins to add a viewer to a viewers list for a specific token for a limited time, measured in hours. The addresses in the viewer list have the ability to access the metadata of the token with the given token ID.

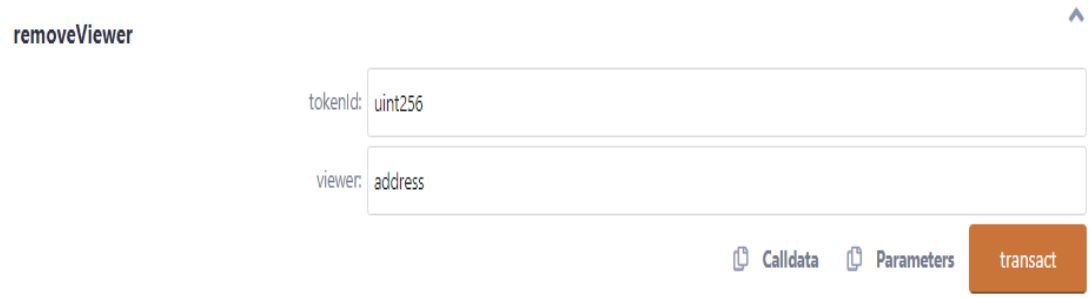

Figure 4.14. Removing viewer for a specific token.

In the figure above, the RemoveViewer function enables admins to remove a viewer from the viewer list of a specific token. The addresses removed from the viewer list will be denied access to the metadata of the token ID given in the function.

4.2. INFORMAL SECURITY ANALYSIS

In this section, we performed a security analysis for our model using the Scyther tool. Scyther is an automated security protocol verification tool that has the ability to characterize protocols and verify protocols with an unbounded number of sessions [36, 37]. Scyther will be used to analyze the security properties of our model by implementing an automated verification and identifying potential vulnerabilities in our system. If any vulnerability or attack is detected, the attack situation will be represented by a graph. Scyther is an effective tool for identifying many types of attacks, which we will mention in the next sections.

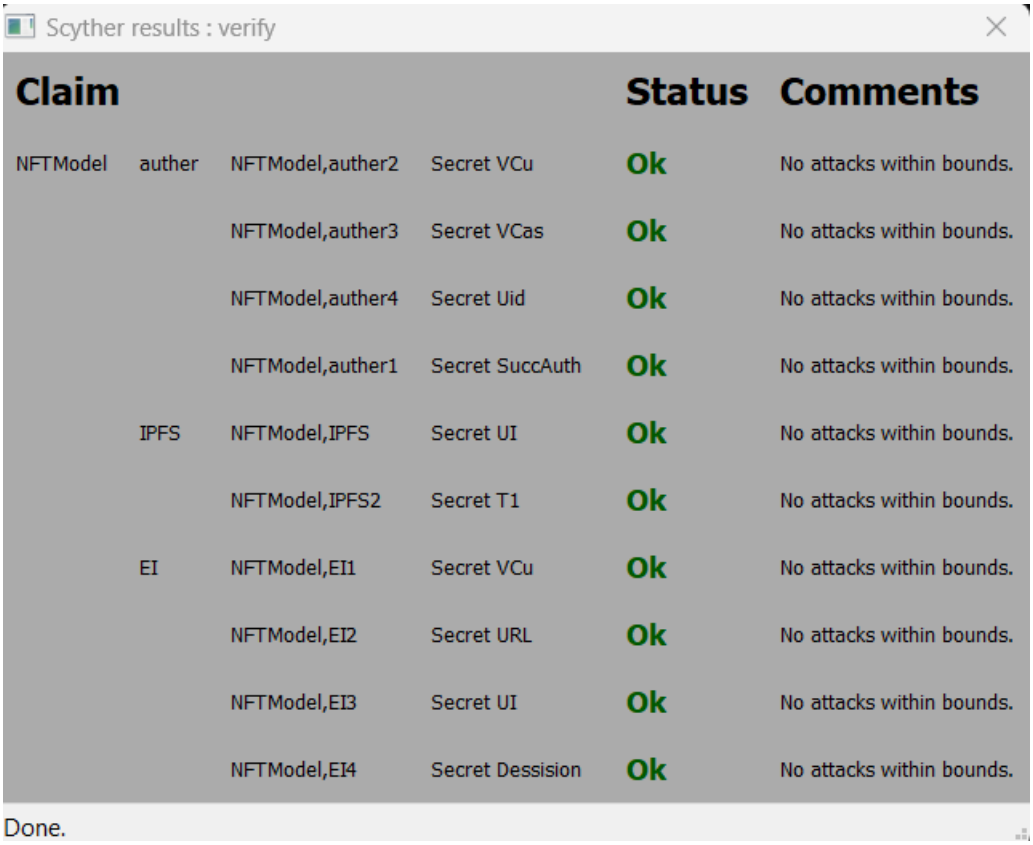

Figure 4.15. Verification claim of the model using Scyther.

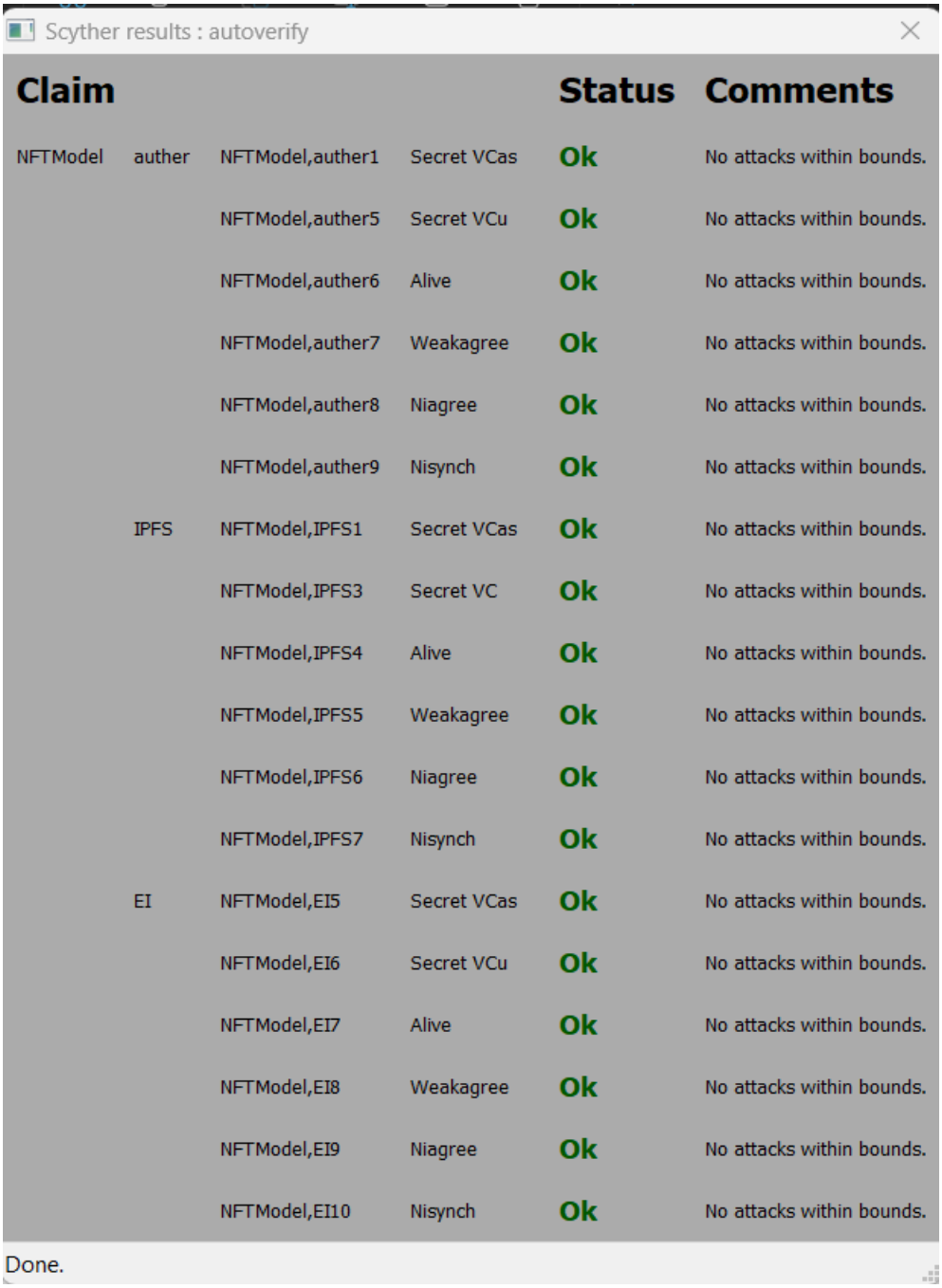

Figure 4.16. Auto verification claim of the model using Scyther.

As we can see in Figures 4.16 and 4.17 above, our model passed Scyther tests successfully. Due to Scyther our model has demonstrated a level of security against various security threats and vulnerabilities. While our proposed model passed the Scyther test, it means that our model was designed with security considerations that identify the potential vulnerabilities. Passing the test enables users and stakeholders to

boosts the confidence in the proposed model and that shows it undergone formal analysis and meets the security standards.

The model that will be tested using Scyther is written using the Solution Process Definition Language [38], so we represented our model using SPDL, as shown in the figure below.

The code below represents a protocol that shows the interactions between actors of our model. The protocol takes three parameters which are EI, IPFS, and author. We didn't use the blockchain network as an actor while we are using a public network which is already secure. The author role has the ability read data from the EI and send data to the EI. IPFS role which represents the InterPlanetary File System has the ability to send and receive data from EI. EI role which represents the Educational Institute has the ability to send data to author and other EIs, read data from IPFS and other EI, and send data to EI and IPFS. We recorded all scenarios of interactions between roles and represented it using SPDL language to test it by Scyther tool. Our model will be explained using the Security Protocol Description Language (SPDL). After running the code on Scyther, it provides us with two figures, as we will show below.

Figure 4.17. Scyther code of our proposed model.

4.2.1. Unauthorızed Access Attack

Where the AccessControl contract doesn't run properly and allows unauthorized users or contracts to access functions or data of the model [39], in our model, an unauthorized access attack can happen if the attacker can use the functions AddViewer(), RemoveViewer(), GrantRole(), and RevokeRole() while he doesn't have ADMIN_ROLE or can execute the Mint() function without the need for MINTER_ROLE. Scyther tool perform a simulation of the unauthorized access attack to check if the model is valid against it via simulating multiple scenarios for the attack and if it discover a vulnerability Scyther will show it as a graph. our model is safe from vulnerabilities that lead to unauthorized access attacks due to Scyther tool.

4.2.2. Denial-Of-Service (Dos) Attack

In simple terms, a DOS attack is to make the network inaccessible to users. A DOS attack occurs by flooding the network with traffic or sending information to the network that causes a crash for it. Flood attacks occur when the attackers send too much traffic for the server to buffer, which leads to a stop in service [40]. The second type of DOS attack is crash attack which happens when the attackers send an input to the system that targets a vulnerability in the system, which leads to a crash or destabilizes the system, making it inaccessible.

4.2.3. External Contract Dependencies Attack

Is the vulnerability that happens when the smart contract depends on or interacts with an external smart contract [41]. If an inherit contract has bugs or vulnerabilities, it can impact the security of the inherited contract. The vulnerabilities of the external contract can also be used by the attackers to affect the behavior of the dependent contract, which will lead to a probability of data leakage and unauthorized access to the dependent contract. In our model, we used Openzeppelin contracts as an external contract, which is an open-source framework to build smart contracts. Openzeppelin provides secure and trusted smart contracts that were already tested and checked for vulnerabilities and

bugs, which makes our contracts safe from this kind of attack. Which is also shown by the results of Scyther testing.

4.2.4. Re-Entrancy Attack

An attack over smart contracts, which allow the attacker to call a function repeatedly and re-enter it before the completion of the previous execution [42]. While smart contracts have an asynchronous nature, which means the state of assets and funds doesn't change until the execution is completed, the contract state can be changed after the function is called to an external contract but before the control back to the initial contract. Due to this attack, unauthorized cryptocurrency or asset transfers can occur. While our contracts enable only authorized roles to call the functions to mint, add viewers, and transfer assets, our proposed model has resistance from re-entrancy attacks, as shown by Scyther tool.

4.2.5. Distributed Denial of Service (Ddos) Attack

is the attempt to disrupt the functioning of the network by flooding it with requests which causes it to exhaust the resources of the network and make it inaccessible to users. DDOS attacks are generally done using a network of computers controlled by the attacker, which is called Botnets [43]. Botnets send a huge number of requests at the same time to the network, which causes a delay in service and in some cases, stops the network [44].

4.3. PERFORMANCE ANALYSIS

In this section, we made a performance evaluation for our model and compared the evaluation results with similar previous works.

We used the Metamask wallet to deploy our smart contract over the Ethereum Goerli test network to test our work. First, the contracts were deployed, and the average cost for deployment was recorded. Then, we ran all methods in the proposed contracts, which are as follows: GrantRole(), RemoveRole(), AddViewer(), RemoveViewer(),

and Mint() 20 times, then we recorded the results and calculated the average gas cost for each method, and lastly, we compared the smart contracts deployment cost between our proposed contracts and the other contracts in similar works.

4.3.1. Gas cost of smart contracts functions

In this section we introduced a Gas cost-based evaluation for our model. Gas is a unit of measurement to the amount of computational efforts required to achieve an operation or transaction over Ethereum blockchain. The gas cost of a transaction is specified by multiplying the gas price and the amount of gas consumed by the transaction. The unit used to identify the gas used is Wei which is equal to 10^{-18} ETH . We ran our methods 20 times and recorded the average evaluation Gas cost of each method, and then we provided a comparison between our results and the results of [29]. The results represented in the figure below are as follows:

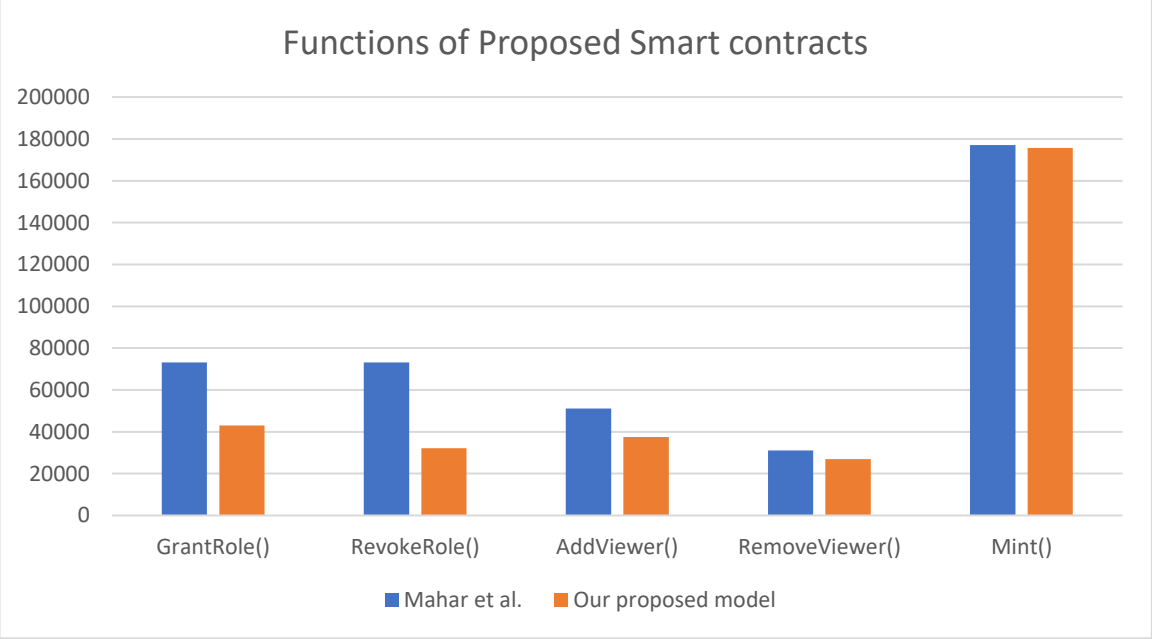

Figure 4.18. Gas cost.

The most expensive method was mint() due to the parameters used in the method and the metadata field, which is the IPFS link of the token. The average cost of methods was as follows: mint() was 175,713 wei, GrantRole() was 43,061 wei, RevokeRole() was 32,183 wei, AddViewer() was 37,556 wei, and RemoveViewer() was 26,907 wei.

On the other hand, average gas cost for the previous work was as follow: mint() was 177,080 , GrantRole() was 73,168 wei, RevokeRole() was 73,168 wei, AddViewer() was 51,122 wei, and RemoveViewer() was 31,079 wei .Due to the results our model has 41.15% less gas usage for GrantRole(), 56.01% less gas usage for RevokeRole(), 26.54% less gas usage for AddViewer(), 13.42% less gas usage for RemoveViewer(), and 0.77% less gas usage for minting contract.

4.3.2. Comparatıve Analysıs of Smart Contracts Gas Cost

In this section, we compared the average cost of deploying our proposed smart contract with the cost of deploying smart contracts of Mahar et al [29] and Sherazi et al [45].

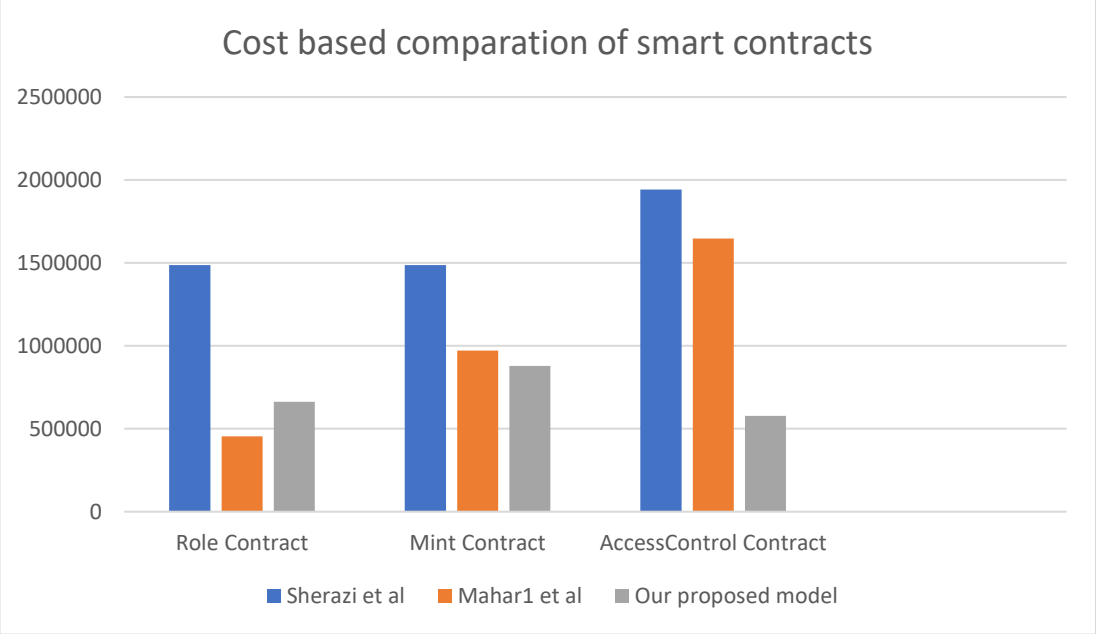

Figure 4.19. Gas cost smart contracts.

As we see in the chart above, our model is the best in terms of cost for AccessControl and Mint Contracts and is approximately the same as Sherazi et al for Role Contracts. The cost of deploying the Mint contract was 970,862 wei for Mahar et al., 1,487,644 wei for Sherazi et al., and 879,388 wei for our proposed contract. The cost of deploying the Mint contract was 1,646,132 wei for Mahar et al., 1,942,996 wei for Sherazi et al., and 578,520 wei for our proposed contract.

The cost of the Deploying Role contract was 454,357 wei for Mahar et al, 1,487,644 wei for Sherazi et al, and 663,062 wei for our proposed contract. Our Role contract perform better than Sherazi's by 55.43%.While our Role contract has GrantRole(), and RevokeRole(), the Role contract in Mahar1 et al has only GrantRole(), which makes the deployment cost of our Role contract a bit higher than Role contract in Mahar et al Mahars role contract has lower cost than our's by 40.52%. Our mint contract cost is lower than Sherazi et al, by 40.89% and lower than mahar et al by 9.42%. Our Access Control contract has lower gas cost from Sherazi et al by 70.23% and better gas consumption from Mahar et al by 64.86%.

PART 5

CHALLENGES AND FUTURE WORKS

5.1. CONTRIBUTIONS

This research reviewed the principles of Metaverse, Blockchain, and NFT technologies, then discussed the models and standards related to our work and identified the limitations of it. We introduced our proposed model to cover gaps and limitations in the previous work. The contribution of our proposed model will be as follow:

- Design a decentralized model for managing educational assets using blockchain and NFT technologies, which provide new solutions to fraud operations, intellectual property protection, and access control issues.
- Design time-based Access Control scheme to enable users and EI to control the access requests to their assets and the duration of accessing the resources.
- Protect the intellectual property of authors by using NFTs to represent the desired assets. This allows users and EIs to protect their work, and control it easily.

5.2. IMPLEMENTATION CHALLENGES

Using blockchain technology helped us solve many issues, but at the same time, it comes with many problems, especially during evaluation and after creating the whole system. Presenting our model to be available for all needs a lot of support from the educational institutes and the Ministry of Education because of the high cost of testing the model in the Ethereum mainnet.

One of the most important issues we have faced is how to trust the educational institutes that offer minting for users and authors. We solved this concern by minting an NFT for the authorized EI, which will be minted by the Ministry of Education and

contains the public addresses of the authorized EI and can be checked by users and authors of the model.

Making performance evaluations for the system was also hard because of the cost of performing many transactions to test the average cost of each smart contract and function. To solve this issue, we used the Ethereum Goerli test network, which enables us to use the Ethereum network for testing our smart contracts and functions without the need to use real ether.

Another important challenge we faced during the implementation was compatibility with the educational platforms and systems. While our model is the first to present all kinds of educational assets as NFTs using public blockchain, we needed to change the architecture of the model many times to make it compatible with the current educational platforms and systems.

5.3. FUTURE WORKS

After the promising results we got from our model, we aim to improve it in terms of scalability and optimize its performance. While we aim to attract a large user base, we need to improve our smart contracts and optimize the cost of deploying the smart contracts and running the functions that will enable the model to be braced by educational institutes. Also, we aim to create a standard for creating the metadata of the assets, which will enable users and other educational institutes to understand the contents of the metadata of the minted token.

5.4. CONCLUSION

In this thesis, a background of the educational asset management and its development over the years are introduced. An introduction to blockchain technology and its applications was given, as was an introduction to NFT technology and its applications were given. Also the usage of NFT technology to manage the educational assets and its importance in preserving intellectual property.

The second chapter provides more details about blockchain and NFT technologies and addresses their importance in managing the educational assets in the Metaverse. A literature review for the related works in managing educational assets using blockchain was introduced.

The third chapter in this work started with referring to the benefits of storing the assets as NFTs, followed by an introduction to the architecture of our proposed model. Then tools used to develop the model were introduced, and development stages were shown. The smart contracts used to build the model were given, and the functions written in every contract were explained. Finally, use cases for our proposed model were given.
The fourth chapter contains the performance evaluation and a comparison between our proposed model and other works due to the gas cost of deploying each contract and function that was deployed.

The last chapter addressed the challenges that we faced during the implementation period, followed by future work that explaining the needs of the area in terms of research.

REFERENCES

1. Shore, C. (2018). How corrupt are universities? audit culture, fraud prevention, and the Big Four accountancy firms. *Current Anthropology*, *59*(S18), S92-S104.

2. penal code of turkey. (2016). european commission for democracy through law. retrieved may 20, 2023, from https://www.venice.coe.int/webforms/documents/default.aspx?pdffile=cdl-ref(2016)011-e

3. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*, 21260.

4. Karim, S. M., Habbal, A., Chaudhry, S. A., & Irshad, A. (2023). BSDCE-IoV: Blockchain-Based Secure Data Collection and Exchange Scheme for IoV in 5G Environment. *IEEE Access*.

5. Hocaoğlu, M., & HABBAL, A. (2022). NFT based model to manage educational assets in Metaverse. *Avrupa Bilim ve Teknoloji Dergisi*, (42), 20-25.

6. Singh, J., Malhotra, M., & Sharma, N. (2022). Metaverse in education: An overview. *Applying Metalytics to Measure Customer Experience in the Metaverse*, 135-142.

7. Yahya, A., & Habbal, A. (2021, October). Music Royalty Payment Scheme Using Blockchain Technology. In *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* (pp. 539-545). IEEE.

8. Sheth, H., & Dattani, J. (2019). Overview of blockchain technology. *Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146*.

9. Al-Amri, R., Zakaria, N. H., Habbal, A., & Hassan, S. (2019). Cryptocurrency adoption: current stage, opportunities, and open challenges. *International journal of advanced computer research*, *9*(44), 293-307.

10. Niranjanamurthy, M., Nithya, B. N., & Jagannatha, S. J. C. C. (2019). Analysis of Blockchain technology: pros, cons and SWOT. *Cluster Computing*, *22*, 14743-14757.

11. Puthal, D., Malik, N., Mohanty, S. P., Kougianos, E., & Das, G. (2018). Everything you wanted to know about the blockchain: Its promise, components, processes, and problems. *IEEE Consumer Electronics Magazine*, *7*(4), 6-14.

12. Cui, P., Guin, U., Skjellum, A., & Umphress, D. (2019). Blockchain in IoT: current trends, challenges, and future roadmap. *Journal of Hardware and Systems Security*, *3*, 338-364.

13. Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., & Qijun, C. (2017, October). A review on consensus algorithm of blockchain. In *2017 IEEE international conference on systems, man, and cybernetics (SMC)* (pp. 2567-2572). IEEE.

14. Efanov, D., & Roschin, P. (2018). The all-pervasiveness of the blockchain technology. *Procedia computer science*, *123*, 116-121.

15. Mukherjee, P., & Pradhan, C. (2021). Blockchain 1.0 to blockchain 4.0—The evolutionary transformation of blockchain technology. In *Blockchain technology: applications and challenges* (pp. 29-49). Cham: Springer International Publishing.

16. William Entriken (@fulldecent), D. S. (2018a, January 24). *ERC-721: Non-Fungible token standard*. Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-721.

17. Wang, Q., Li, R., Wang, Q., & Chen, S. (2021). Non-fungible token (NFT): Overview, evaluation, opportunities and challenges. *arXiv preprint arXiv:2105.07447*.

18. Bao, H., & Roubaud, D. (2022). Non-fungible token: A systematic review and research agenda. *Journal of Risk and Financial Management*, *15*(5), 215.

19. Chevet, S. (2018). Blockchain technology and non-fungible tokens: Reshaping value chains in creative industries. *Available at SSRN 3212662*.

20. Musamih, A., Dirir, A., Yaqoob, I., Salah, K., Jayaraman, R., & Puthal, D. (2022). NFTs in Smart Cities: Vision, Applications, and Challenges. *IEEE Consumer Electronics Magazine*.

21. Fabian Vogelsteller, V. B. (2015, November 19). *ERC-20: Token standard*. Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-20.

22. Witek Radomski, A. C. (2018, June 17). *ERC-1155: Multi token standard*. Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-1155.

23. Hwang, G. J., & Chien, S. Y. (2022). Definition, roles, and potential research issues of the metaverse in education: An artificial intelligence perspective. *Computers and Education: Artificial Intelligence*, *3*, 100082.

24. Zhao, X., & Si, Y. W. (2021, December). NFTCert: NFT-based certificates with online payment gateway. In *2021 IEEE International Conference on Blockchain (Blockchain)* (pp. 538-543). IEEE.

25. Kolvenbach, S., Ruland, R., Gräther, W., & Prinz, W. (2018). Blockchain 4 education. In *Proceedings of 16th European Conference on Computer-Supported Cooperative Work-Panels, Posters and Demos*. European Society for Socially Embedded Technologies (EUSSET).

26. Kumar, N. N., Kumar, R. S., Basale, R. R., & Saffath, M. (2022, January). Decentralized Storage of Educational Assets Using NFTs And Blockchain Technology. In *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 260-266). IEEE.

27. Blockcerts. (n.d.-a). *Blockchain credentials*. Blockcerts. https://www.blockcerts.org/ .

28. Ocheja, P., Flanagan, B., Ueda, H., & Ogata, H. (2019). Managing lifelong learning records through blockchain. *Research and Practice in Technology Enhanced Learning*, *14*(1), 1-19.

29. Mahar, U., Aleem, M., & Zahoor, E. (2023). TTECCDU: a blockchain-based approach for expressive authorization management. *PeerJ Computer Science*, *9*, e1212.

30. Li, K., Tang, Y., Chen, J., Wang, Y., & Liu, X. (2021, November). TopoShot: uncovering Ethereum's network topology leveraging replacement transactions. In *Proceedings of the 21st ACM Internet Measurement Conference* (pp. 302-319).

31. Wohrer, M., & Zdun, U. (2018, March). Smart contracts: security patterns in the ethereum ecosystem and solidity. In *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)* (pp. 2-8). IEEE.

32. Lee, W. M., & Lee, W. M. (2019). Using the metamask chrome extension. Beginning Ethereum Smart Contracts Programming: With Examples in Python, Solidity, and JavaScript, 93-126.

33. Benet, J. (2014). Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*.

34. Amri, S. A., Aniello, L., & Sassone, V. (2023). A Review of Upgradeable Smart Contract Patterns based on OpenZeppelin Technique. *The Journal of The British Blockchain Association*.

35. Amir Latif, R. M., Hussain, K., Jhanjhi, N. Z., Nayyar, A., & Rizwan, O. (2020). A remix IDE: smart contract-based framework for the healthcare sector by using Blockchain technology. *Multimedia tools and applications*, 1-24.

36. Cremers, C. J. (2008). The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols: Tool Paper. In *Computer Aided Verification: 20th International Conference, CAV 2008 Princeton, NJ, USA, July 7-14, 2008 Proceedings 20* (pp. 414-418). Springer Berlin Heidelberg.

37. Cremers, C. (2006). Scyther. Semantics and Verification of Security Protocols, Thesis, University Press Eindhoven.

38. Moscato, F., Vittorini, V., Amato, F., Mazzeo, A., & Mazzocca, N. (2011). Solution workflows for model-based analysis of complex systems. *IEEE Transactions on Automation Science and Engineering*, *9*(1), 83-95.

39. Ding, H., Han, J., Zhang, Y., Xiao, F., Xi, W., Wang, G., & Jiang, Z. (2018, April). Preventing unauthorized access on passive tags. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications* (pp. 1115-1123). IEEE.

40. Hasbullah, H., & Soomro, I. A. (2010). Denial of service (DOS) attack and its possible solutions in VANET. *International Journal of Electronics and Communication Engineering*, *4*(5), 813-817.

41. Praitheeshan, P., Pan, L., Yu, J., Liu, J., & Doss, R. (2019). Security analysis methods on ethereum smart contract vulnerabilities: a survey. *arXiv preprint arXiv:1908.08605*.

42. Samreen, N. F., & Alalfi, M. H. (2020, February). Reentrancy vulnerability identification in ethereum smart contracts. In *2020 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE)* (pp. 22-29). IEEE.

43. Alomari, E., Manickam, S., Gupta, B. B., Karuppayah, S., & Alfaris, R. (2012). Botnet-based distributed denial of service (DDoS) attacks on web servers: classification and art. *arXiv preprint arXiv:1208.0403*.

44. Issa, A. S. A., & Albayrak, Z. (2023). Ddos attack intrusion detection system based on hybridization of cnn and lstm. *Acta Polytechnica Hungarica*, *20*(2), 1-19.

45. Sherazi, S. N. A., Zahoor, E., Akhtar, S., & Perrin, O. (2022). A blockchain based approach for the authorization policies delegation in emergency situations. *Transactions on Emerging Telecommunications Technologies*, *33*(5), e4461.

RESUME

Muhammed HOCAOĞLU then completed the high school education in Alsaade Private School in the same city. He moved to the Republic of Turkiye in 2014, and started undergraduate program in Karabuk University department of Computer Engineering between 2015-2020. Then at 2020 he started to M. Sc. Education at Karabuk University.