A Secure Authentication Mechanism for Federated Metaverses using Non-fungible Tokens (NFTs)

by

Mohammed Shakib Hossain
21301307
Khawaja Ehsun Ul Hawak
21301284
Mutasin Jawad
21301435

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Students Full Name & Signature:

Mohammed Shakib Hossain
21301307

Khawaja Ehsun Ul Hawak
21301284

Mutasin Jawad
21301435

Approval

The thesis titled A Secure Authentication Mechanism for Federated Metaverses using Non-fungible Tokens (NFTs) submitted by

1. Mohammed Shakib Hossain(21301307)
2. Khawaja Ehsun Ul Hawak(21301284)
3. Mutasin Jawad(21301435)

Of Summer, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 20, 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Md Sadek Ferdous
Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

The rapid growth of consumer-grade augmented reality (AR) and virtual reality (VR) devices has driven the interest in Metaverses, a type of fully immersive virtual world where people may communicate and conduct their businesses. However, similar to any restricted online services, the authentication of users is a crucial component to access any Metaverses. The authentication mechanism in the traditional online services has a number of issues such as password vulnerabilities, centralized control, identity fragmentation, session hijacking, scalability challenges, credential theft etc. Adopting such an authentication mechanism would introduce similar issues in the Metaverse domain. Moreover, there are also questions about additional security concerns in Metaverses as it is hard to control an individual's virtual actions. Hence, Blockchain create a decentralize environment that empowers users to take charge of their individual actions and identities. A smart contract that tokenize identity in the Metaverse can be executed securely within the blockchain technology. Therefore, a new authentication mechanism must be explored in order to increase the security of Metaverses. In this research, We propose a system utilizing the concept of Metaverse Federation in order to provide authentication throughout various Metaverse, where multiple Metaverse will form a Circle of Trust (CoT) and the federation facilitates a secured authentication and identity management using a shared registry table for each Metaverse. This system leverages smart contracts and blockchain technology to generate Non-fungible token(NFT's), which allow users to authenticate in various Metaverse. This enables users to navigate multiple Metaverses with ease and a trusted identity, while complying with the specific rules and regulations of each Metaverse. we aim to explore a novel authentication mechanism tailored for Metaverses that will utilize the concept Non-fungible tokens (NFT's) and the Metaverse Federation.

Keywords: Metaverse, BlockChain, Smart Contract, token, Non-fungible token (NFT), federated identity, Authentication.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our Supervisor Md Sadek Ferdous sir for his kind support and advice in our work. He helped us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The following list describes several symbols and abbreviations that will be used within the body of this document:

AES Advanced Encryption Standard

API Application Programming Interface

AR Augmented Reality

DAG Directed Acyclic Graph

DID Decentralized Identity

DDoS Distributed Denial of Service

DoS Denial of Service

FMA Federated Metaverse Authentication

IoT Internet of Things

MR Mixed Reality

NFTs Non-fungible Tokens

PKI Public Key Infrastructure

RSA RivestShamirAdleman

SBT Soulbound Tokens

VR Virtual Reality

zkp Zero-Knowledge Proof

Chapter 1

Introduction

The word 'Metaverse' describes an integrated, shared online environment including persistent virtual spaces, virtually modified representations of the real world, and augmented reality experiences. AR (augmented reality) and VR (virtual reality) technologies play a critical role in the immersive nature of the metaverse, ensuring a seamless and engaging user experience. By creating personal avatars, individuals can gain access to the metaverse and get involved in luxurious, realistic interactions within one another via these avatars [6].

A metaverse is a fully immersive virtual environment where people can communicate, connect with others, and transact commerce. These virtual spaces combine the digital and real worlds in novel and fascinating ways, providing previously unheard-of chances for social engagement and business. Metaverse has a huge market potential and expected to contribute nearly 3 trillion by 2031 to the global GDP [27]. It is a versatile and innovative technology widely spread to various domains and application including gaming, entertainment, real estate, education, retail etc.

Unlike any other online based application metaverse also require to authenticate user's. Traditional authentication mechanism has several issues like phishing attack, password vulnerabilities, Bio-metric data risk etc. Therefore, in order to address the security concern of the Metaverse application, a new authentication technique needs to be explored as it can be challenging to regulate an individual's virtual behavior. Using the immutable and decentralized ledger features of blockchain technology, we can offer a secure authentication mechanism for the metaverse. Blockchain works as an immutable ledger that permits decentralized transaction processing making its difficult to change the data [19]. It can made secure, transparent and immutable digital record. Blockchain technology uses smart contracts, which are self-executing legal agreements with the terms of the agreement explicitly recorded in code. Smart contracts guarantee that the terms of the agreement cannot be changed and offer security and confidence. It create digital token in order to offer automation, security, transparency, and decentralization. One of those digital token is NFT (Non-fungible token) which can be useful for providing authentication in digital environment such as metaverse. NFTs are distinct digital assets stored on a blockchain that aren't trade-able or convertible into other tokens. It can be used to confirm the legitimacy and ownership of users' exclusive possessions, such as virtual real estate, digital artwork, and so on.

Thus, it is imperative that the Metaverse Application offer a novel authentication

mechanism that leverages the concept of Non-fungible tokens (NFTs) to enhance identity management, security, user experience, and transaction across a federated Metaverse environment, where multiple platforms collaborate under a shared framework of trust.

1.1 Problem Statement

In the Metaverse, Users are not merely interacting with one another in real-time but they can also design, as well as personalize their digital identities, which will directly communicate with other users too. As AR and VR technologies continue to gain traction, immersive systems grow to encompass vast territories and this opens up a very broad set of security-related threats. The increasing popularity of cryptocurrencies and NFTs related to the user avatar demonstrates a change in ownership and identity to the digital form. In addition, the idea of the appearance of services, such as virtual healthcare, where biometric information is transferred (shared) and also is connected to the profile of a user, data integrity and security become critically important.

Most of the Metaverse platforms have traditionally followed typical authentication strategies including the combination of usernames and passwords, biometrics, two-factor authentication (2FA), the use of OAuth third party logins, and session tokens. All these processes are associated with weaknesses in an immersive and decentralized environment. Password-based systems are easy targets to phishing, credential stealing, brute-force attacks, and other challenges such as weak passwords or reused passwords and forgotten credentials. Biometric data is more difficult to duplicate but adds the risk of loss of privacy and irreversibility: in the event of a breach, it is not possible to re-set a biometric data, like passwords. The use of OAuth and other third-party authentication solutions brings centralized trust relationships, which is the opposite of Metaverse decentralization and can lead to the existence of single points of failure. Session tokens are efficient and often susceptible to man in the middle attacks and session hijacking. These vulnerabilities are very dangerous in the Metaverse where authentication plays a key role in not only verifying identity but also the capability of owning and transferring digital assets. As an example, users might want to sell or exchange their Metaverse profile, avatar or in game assets, in which case the secure, verifiable, user controlled authentication is critical to enable such exchange to take place without centralized control.

Additionally, the issues of multiple identities and repetitive authentication are becoming an important barrier to adoption as the users engage with multiple Metaverse platforms. A Federated Metaverse model, in which various Metaverses are part of a Circle of Trust (CoT), is capable of providing secure and seamless authentication within a single, trusted user credential. With such a model in place, interoperability between Metaverses would be possible, duplicate identity verification would not need to take place and the general user experience would be improved without sacrificing security. Hence, the need to have a decentralized, NFT-based authentication system is required that will not only provide scaleable security and identity management but also enable identity federation between two or more Metaverses is the future of online interaction and secure transfer of Digital Assets.

1.2 Research Objective

We aim to facilitate a proper authentication mechanism for Metaverse Application by utilizing the concept of Non-fungible token(NFT) in conjunction with Blockchain Technology. Within the help of this idea we aim to overcome the lacking existing authentication mechanism problem for metaverse and provide a robust, user-friendly experience in such an online based platform.

- **Research Objective 1 (RO1): Analysing existing authentication mechanism used for Metaverse.** Our first goal of the research is to collect and analyze different types of authentication systems that are being used for Metaverse. By doing so we can identify the advantages and disadvantages of these authentications. This will help us to understand their strength and weakness, specifically in the context of online service and virtual environment.
- **Research Objective 2 (RO2): Investigating how NFTs based blockchain technology is used in Metaverses for authentication.** Our goal is to investigate how to improve security and identity management in metaverse application utilizing NFT based authentication.
- **Research Objective 3 (RO3): Designing and developing a system based on our proposal.** The goal is to design and create a unique authentication system that make use of blockchain based technology, specifically NFT (Non-fungible token). This also involved a description of its implementation methods, protocols and architecture.
- **Research Objective 4 (RO4): Testing and evaluating our system.** To accomplish this goal, the constructed system must requires comprehensive testing and evaluation in order to make sure its effectiveness, security, privacy, and accessibility. And also Its require to identify the area of improvement.

1.3 Thesis Outline

The rest of the chapters of the paper is described below:

- **Chapter 2:** In this Chapter we provide a summary of the key concepts that are required to fully understand the research, We included the basic idea of metaverse, blockchain, Non-fungible token(NFT) and authentication mechanism and the identity federation.
- **Chapter 3:** In this chapter we review some existing literature paper that related to metaverse, blockchain Based NFT application, authentication mechanism and highlights some of the research gaps and the opportunities for our future work.
- **Chapter 4:** In this chapter we present a proposal of novel NFT based authentication mechanism for metaverse application and discuss about methodology, thread modeling and requirement analysis.
- **Chapter 5:** In this chapter we proposed an Architecture design of our system.

- **Chapter 6:** In this chapter we illustrated certain Use Cases and their protocol flow of users authentication using NFTs in Federated metaverses.
- **Chapter 7:** This chapter contains an in-details discussion on the entire development dependencies and tools that were involved in the implementation of our system.
- **Chapter 8:** This chapter has a detailed discussion which includes requirement analysis, comparative analysis, cost analysis, advantages and challenges, and limitation of the proposed system and the possible future work.
- **Chapter 9:** The chapter is a summary of all the findings of the thesis in which they note and outline the main contributions to the study, the results of implementations, and last remarks on the basis of objectives reached.

Chapter 2

Background

In this chapter, we aim to explore and discuss Blockchain, different types of blockchain, and its key components and applications in (Section 2.1), discuss different aspects of a metaverse in (Section 2.2), discuss regarding authentication and its challenges in (Section 2.3), explore the Non-Fungible Token and how it works for user authentication in (Section 2.4) and cover about Federated Identity Management and its operation in (Section 2.5).

2.1 Blockchain

Blockchain technology was first applied by an unknown person or a group of people known as Satoshi Nakamoto in 2008 [5]. To some extent achieving a decentralized and distributed ledger on a definite platform was the main purpose of creating blockchain. In an effort to ensure that data stored in the blockchain is secure and cannot be altered, the blockchain keeps transaction copies in blocks that are linked to one another through an encrypted hash for each one they form a chain [17]. This makes any data block if provoked to the ledger, very hard to alter or even delete together. In this case, any changes to any data in a block mean that the changes should be made to all the subsequent blocks as well. In a Blockchain, since the nodes are distributed, consensus is necessary, and, in this case, all the machines in the network will carry a copy of the network. Furthermore, since there is an element of decentralization in the management of the network, and with the understanding that there is no possibility that any data within this block can be changed; this approach means that the presence of central authority in the administration of any transactions is not necessary [5].

2.1.1 Type of Blockchain

There are mainly 4 types of Blockchain networks based on different key factors such as accessibility, control, and participation in the network:

1. **Public Blockchain:** A public blockchain is decentralized in the network and it an openly accessible meaning anyone can get involved in the public blockchain network and can contribute to the consensus process as long as they follow certain rules. There is no central authority in the network. So, all the data is accessible and viewable by anyone in the network and they also can

verify all the data in the blockchain network [24]. For example: Ethereum, Bitcoin, Solana, etc.

2. **Private Blockchain:** A private blockchain is a permissioned network that only accesses authorized users. It is usually employed by businesses that seek to maintain more control over the blockchain, providing only trusted parties permission to view the data and transaction process [24]. For example: Hyperledger Fabric, Hyperledger Sawtooth, Corda, etc.

3. **Consortium Blockchain:** A consortium blockchain is a network that is partially decentralized and maintained by several organizations (a set of private entities). It operates to be a common platform in which only specific entities are authorized to take part in the consensus-building process and it would be able to be configured using private blockchain [26]. For example: Ethermint, Energy Web Foundation, etc.

4. **Hybrid Blockchain:** A hybrid blockchain integrates aspects of private and public blockchains. It keeps sensitive data private and only permits authorized parties to access certain areas of the blockchain [24]. For example: LTO Network, Sovrin, etc.

2.1.2 Key Components

The key components of blockchain technology include decentralization, consensus mechanisms, immutability, transparency, and security:

1. **Decentralization:** Blockchain is a distributed network that is open to any member, and every member of the network has copies of the entire network. This mechanism supports the ability to subtract the risk of a single point of failure and also enables the transparency of the data [12]. In addition, decentralized systems can also mean that there is no need for authority to perform any transactions.

2. **Consensus:** In the Blockchain, consensus is one of the most vital aspects. This means that all the machines in the Blockchain network have come to a consensus on the current state of the Blockchain. In making sure that all the machine has an understanding of the state of the Blockchain, the Consensus algorithm is used. Which guarantees data integrity and prevents any possibilities of forking [23].

3. **Immutability:** The fact that once the data is entered in the blockchain it is permanent in the ledger makes it impossible to alter or erase a transaction. This feature is of significant importance in maintaining the necessary accuracy of data history while also making it resistant to alterations.

4. **Transparency:** This aspect makes the transactions transparent and the people involved accountable since every transaction is recorded and can be accessed by all the people in the network. This capability is particularly useful in such fields as supply chain management and finance.

5. **Security:** Among the potential advantages of the Blockchain technique, the safety of data is undoubtedly one of the main advantages. As to the security measures, blockchain uses sophisticated cryptography techniques in dealmaking. Special consensus algorithms like the Proof of Work (PoW) and the Proof of Stake (PoS) help to maintain the integrity of the data and the data cannot easily be changed by the fraudsters [21].

2.1.3 Applications of Blockchain

In the modern world, blockchain has found its application in different spheres of our lives. Some notable examples include:

1. **Cryptocurrencies:** The best-known application of blockchain is in cryptocurrencies, like Ethereum and Bitcoin, due to which they organize transactions safely and without the involvement of any central authorities [16].

2. **Supply Chain Management:** Blockchain can make the supply chain more transparent and easy to track because it enshrines in an irreversible form all the movements of the product.

3. **Healthcare:** Blockchain can safeguard patients records, preserve confidentiality, and enable the interoperability of data in different healthcare organizations, while at the same time providing secure data integrity.

4. **Finance:** Blockchain can transform banking and finance through the enhancement of cross-border payments, clearing, and settlement since they can be carried out in less time and at a cheaper price [15].

2.1.4 Challenges

Despite the possibilities, blockchain technology faces several challenges:

1. **Scalability:** The most significant challenge, as is still the case today, is the throughput or the number of transactions that blockchain networks can process per second. To help overcome this issue, solutions like sharding and off-chain transactions are being researched, for example, [14].

2. **Energy Consumption:** Mainly with PoW consensus algorithms, the utilization of blockchain networks can significantly consume substantial amounts of energy. Alternatives such as PoS, are being made to minimize energy consumption.

3. **Regulation and Legal Issues:** While blockchain is relatively new as a technology, the regulation of this technology is constantly expanding and there are concerns over compliance, privacy, and legal enforceability [13].

2.2 Metaverse

The Metaverse is often described as a collective virtual shared space, created by the convergence of virtually enhanced physical reality and physically persistent virtual

space, including the sum of all virtual worlds, augmented realities, and the internet. Although the term Metaverse was first coined in Neal Stephensons 1992 science fiction novel Snow Crash, the concept of Metaverse in this narrative differs a little bit from its current definition. The Metaverse has been recently receiving considerable attention as applications in VR, AR, and the blending of digital and physical spaces are increasingly becoming pressing [10].

2.2.1 Origins and Development

The concept of Metaverse has its origin in the prior advancement and cultural influences in technological development. It is necessary to mention the apparent antecedents of the Metaverse in other more antecedent virtual environments: MUDs as well as the immersive graphical worlds such as Habitat, developed in the 1980s [9]. They provided the groundwork for the completely populated paradigms of todays games. In addition, the advancement of the internet, the emergence of advanced GPUs strategically enhance VR/AR technology, and the latest generation VR/AR equipment have greatly boosted the prospects of the Metaverse [9].

2.2.2 Technological Components

One of the defining platforms of the Metaverse leverages the use of VR, AR, blockchain, and AI technologies. Both VR and AR are important in giving users experiences that blur the boundaries between the physical and digital worlds. Contrary to this, Blockchain technology is given significance to set the economic frameworks in the context of Metaverse including transactions and digital ownership [5]. AI contributes by enhancing the interactivity and responsiveness of virtual environments, providing realistic simulations and intelligent virtual agents [20].

2.2.3 Social and Economic Implications

The Metaverse means the radical rethinking of social interactions and an economy as a whole. New opportunities are opened for people: socialization, getting to know new people and places, education, entertainment, and work. For example, virtual worlds such as Second Life have shown that a user can develop content, conduct businesses, and organize complex communities within virtual worlds [4]. This leads to such the concept as virtual economics in which the real money value is assigned to digital products and services, and it is also one of the most important Pillars of the Metaverse. Some of these economies employ virtual currencies and blockchain technology to conduct these transactions and create ownership likenesses [2].

2.2.4 Current Trends and Future Directions

From the recent developments, it is clear that the Metaverse is gradually undergoing a transformation to a higher and more advanced level. It is evident that Metaverse is the new frontier of the future world where major technology firms such as Facebook (Meta), Google, and Microsoft are now venturing into establishing their ground. The COVID-19 pandemic has also played a role in making people turn to utilize the Metaverse for distance communication and business [31].

Researchers believe that the Metaverse is still in the process of development and expansion and will sometimes blend with our society and daily routine. Nevertheless, major impediments, including data privacy, security, and ethical dilemmas shall continue to emerge as burning issues that need to be addressed. Hence as the Realms of Metaverse increase, research and collaboration across various fields will be crucial in order to have harmonious progress that will benefit everyone in society as we progress in the future.

2.3 Authentication

In simple terms, authentication is a process of recognizing and confirming the identity of a user requesting to access a particular system or certain particular service. It confirms the identity of an entity, which is especially important for safeguarding digital systems and businesses. The following information looks at concepts related to authentication in detail.

2.3.1 Types of Authentication

Depending on the verification factors, there are three authentication techniques [33]:

1. **Something You Know:** This type includes details that are personal to the user only, for instance, log in details to an account, or identification numbers. It remains vulnerable to phishing attacks, brute force, and social engineering. For instance, a user enters their password or answers a security question to log in to an account.

2. **Something You Have:** This type utilizes objects that are possessed by the user. For example, smart cards, tokens, and mobile devices. It is relatively more secure compared to knowledge-based methods but there is the risk of the object being misplaced or stolen. For example, a user inserts a smart card into a reader or uses a one-time password (OTP) generated by an authentication token or mobile app.

3. **Something You Are:** It employs data such as fingerprints, facial recognition mechanisms, or scanning of the eyes to authenticate a persons identity. This type of identification can be said to be very secure because it involves using biometric features, which are believed not to be comparable to any other thing, although there are costs that come with it such as privacy and quite recently biometric data leakage. For instance, a user unlocks their phone using fingerprint recognition or logs into a system through facial recognition.

2.3.2 Multi-Factor Authentication (MFA)

MFA makes up at least two of the mentioned above types of authentication to increase the security of the system. For instance, a user might be required to use a password-which is knowledge-based, and a code from their mobile phone which is a possession-based authenticator to access their account. MFA then minimizes the risk of unauthorized entry in the system by introducing different levels of authentication involved [39].

2.3.3 Challenges in Authentication

In spite of improvement, there are still a number of issues with the present authentication system [32]:

1. **Usability vs. Security:** In many cases, the main problem of the process of authentication is the balance between Ease of use, which means that users should not face any difficulties in the process, and Security which means that the system should be as protected as possible and intruders should not be able to perform the process of authentication. This is especially true when the format of authentication requires multiple steps which may in the long run discourage the users.

2. **Scalability:** The management of authentication is always difficult as the system gains complexity. One of the most significant problems is how to maintain an adequate level of security when applied to a numerous number of users and devices.

3. **Privacy Concerns:** Storing peoples distinctive characteristics as well as other types of personal information leads to privacy violation threats and data leaks.

2.4 Non-Fungible Tokens (NFTs)

NFT is a cryptographically secure token that proves ownership or guarantees the authenticity of an item, even if it is digital. Unlike Bitcoin, or any other cryptocurrency for that matter, NFTs cannot be exchanged in a 1:1 manner, as they are non-fungible a concept that offers a whole new way of confirming and exchanging ownership rights to tangible and intangible assets [25]. We will be discussing concepts related to using and managing NFTs below.

2.4.1 Origins and Evolution

NFT started out in 2012 with the introduction of the Colored Coins in the Bitcoin platform. These were earlier suggested by Yoni Assia and were planned to represent physical assets like property or shares where certain satoshis would be set as the smallest value of Bitcoin. Nevertheless, Colored Coins encountered certain problems because of the Bitcoin scripting language that contributed to the creation of other sophisticated blockchain technologies [7].

Ethereum, which was developed in 2015, introduced the feature of smart contracts in the blockchain. This set the stage for NFTs on Ethereum, which was the dominant platform for such items [8].

2.4.2 NFTs Work Flow

The process of creating an NFT involves several key steps[25]:

1. **Minting:** The process to create an NFT is called minting. Minting converts digital assets into crypto assets on the blockchain. It is done by using smart contracts.

2. **Smart Contracts:** On Ethereum, ownership and transferability of an NFT is managed using smart contracts. The most commonly used standard for smart contracts is ERC-721.

3. **Metadata:** Metadata contains important data that can describe the attributes of each NFT. It is stored on the blockchain and generally contains data related to the creator's identity, history of the asset, and any additional attributes.

4. **Storage:** Although we store the NFT on the blockchain, the asset is generally stored in somewhere else to save gas fees.

2.4.3 NFT Standards

There are different kinds of NFT standards available to fulfill various functionalities:

1. **ERC-721:** This was the first standard for NFTs on Ethereum. It makes it possible to generate new tokens with an ID number and due to that, one token cannot be exchanged with the other token. This standard is perfect for digital art, non-fungible tokens, collectibles, and items in virtual games.[25]

2. **ERC-1155:** Introducing an enhancement to the ERC721 standard, ERC1155 enables the users to create both fungible and non-fungible tokens in a single contract. This is less costly in terms of gas fees and space and is preferable for gaming where several forms of assets are likely to be used.[25]

2.4.4 NFT-Based User Authentication

NFTs have recently been proposed as a special means of authenticating user identities since they rely on blockchain technology to enable unique digital signatures. Currently, NFTs are nontraditional tangible assets contained on the blockchain and linked to metadata that can embody various forms of identity credentials. As a result of connecting an NFT to the user identity, it allows the system to authenticate users in a distributed and safe manner. Next, we will be discussing the process, advantages, and challenges of NFT-based user authentication[29]:

Process:

1. **Issuance:** A reliable third party, such as an organization, creates an NFT that contains the users details, and this is recorded in a blockchain.

2. **Verification:** If authentication is needed, the system checks whether the NFT is genuine and belongs to the rightful owner through the help of blockchain validation.

3. **Access Granting:** Once the ownership is confirmed, the user is given the green button, and s/he is allowed to use the system or service.

Advantages:

1. **Security:** Blockchain technology guarantees that data related to NFTs cannot be changed or modified in any way, thus strengthening the protection of the identification procedure.

2. **Decentralization:** It also has the advantage of eliminating the necessity of central control of the users information and accounts, thus minimizing potential risks of data leakage from a single source.

3. **Uniqueness:** Collectively, each of the NFTs is distinct and gives the mechanism a solid foundation to guarantee and authenticate unique identities.

Challenges:

1. **Complexity:** It can be challenging to implement due to its drastic changes in the architecture of its design and utilization of NFT in the process of authentication.

2. **Adoption:** It is worth mentioning that some of the sectors have already used some sort of verification to a considerable extent. However, for the use of NFTs for authentication to be made as a standard practice, it may probably take time.

3. **Privacy:** Managing the identity of users together with anonymity and maintaining the network clear and open, is also an issue with otherwise uses of a blockchain.

2.4.5 Future Prospects

However, NFTs are not without issues, such as the problems of energy consumption caused by Ethereums proof-of-work consensus algorithm. This caused proposals for creating blockchain alternatives, such as proof-of-stake, that are more ecologically friendly. In addition, the market has been noted to be very speculative, which has sparked concerns about its viability and even the potential for market bubbles. Moving forward, NFTs will continue to diversify its involvement in numerous sectors including gaming, entertainment, real estate as well as in finance. New enhancements in the ways of connection, management, and interface with NFTs are likely to enhance the practicality, adoption, and usability of these digital assets [28].

2.5 Federated Identity Management (FIM)

Federated Identity Management is an identity management system that allows users to authenticate in different organizations and get services using a single set of credentials. In a federated identity system, entities, such as the service provider and the identity provider, establish trust between each other in order to exchange user authentication data securely. This is typically used when the users require access to services from different organizations without having to provide individual logins.

2.5.1 Key entities in Federated Identity Management

1. **Identity Provider (IdP):** An Identity Provider (IdP) authenticates a user and maintains the user's credentials which include usernames, passwords, or other authentication tokens. The IdP verifies the user's identity and after successful authentication, an authentication token is provided that can be shared with other metaverses.

2. **Service Provider (SP):** A Service Provider (SP) is an entity that provides services or access to resources (such as websites, applications, and platforms) that users desire to access. Instead of managing the user credentials itself, the Service Provider trusts the Identity Provider to verify the user. After the identity provider authenticates the user, the service provider gives the user access to services.

3. **User (Clients):** A user is an entity that obtains services from Service Providers (SPs) upon successful authentication by an Identity Provider (IdP).

2.5.2 Federated Identity Management Operation

- A federated identity management system builds trust relationships amongst different domains (e.g. a company, a platform, or a service). They agree to trust any authentication done by a trusted Identity Provider. By doing so, users can use these services from any one domain within them without having to re-enter their credentials at each one.

- Once authenticated by an Identity Provider, a user receives a token (often using standards like SAML, OAuth, or OpenID Connect). This token includes information about the users identity and is sent to the Service Provider to allow them to access without requiring the user to authenticate again.

2.5.3 SAML in Federated Identity Management

Security Assertion Markup Language (SAML) is an XML-based open standard for exchanging authentication and authorization data between parties, specifically between an identity provider (IdP) and a service provider (SP) based on the Federated User Identity model[3]. SAML enables a secure, federated authentication process by allowing the identity of a user to be verified and communicated across different systems without exposing sensitive credentials. SAML is built around several key XML-based components that facilitate the secure transmission of data between an identity provider (IdP) and a service provider (SP)[3]:

i. **SAML Assertion:** The core data structure in SAML, an assertion is an XML document that contains statements about a users authentication, attributes, and authorization. The IdP issues this to communicate the user's identity and authorization rights to the SP.

ii. **SAML Protocol:** This defines the rules and procedures for communication between the IdP and SP, outlining how requests for authentication and responses (assertions) should be formatted, transmitted, and handled.

iii. **SAML Binding:** SAML bindings specify how SAML protocol messages are transported over underlying communication protocols such as HTTP or SOAP. For instance, HTTP-Redirect and HTTP-POST are common bindings used to exchange SAML messages in web-based scenarios.

iv. **SAML Profile:** Profiles define specific use cases and provide detailed implementation guidelines for various scenarios, such as Single Sign-On (SSO). They ensure that SAML is consistently applied across different systems, making interoperability possible.

In a typical SAML authentication flow, three key entities are involved: the user (client), the identity provider (IdP), and the service provider (SP). The SAML protocol flow is captured in figure 2.1, where, in the Web Browser SSO Profile, both the SP and the IdP belong to a federation, and the binding is the HTTP post binding, the flow is as follows:

i. **User Request to Access SP:** The process begins when a user attempts to access a service provided by the SP. The user may not yet be authenticated at this point. The SP recognizes that it relies on an external IdP for user authentication.

ii. **SP Initiates Authentication Request to IdP:** Since the SP does not handle the user's authentication directly, it sends a SAML authentication request to the IdP. This request typically contains information about the SP, including the URL the user is trying to access. The request is either sent through the users browser (browser-based flow) or directly via back-channel communication, depending on the implementation.

iii. **User Authentication by IdP:** Upon receiving the request, the IdP checks if the user is already authenticated (via a valid session or token). If the user is not authenticated, the IdP prompts the user to log in, often using credentials like a username and password or another authentication method (e.g., multi-factor authentication). The IdP verifies the credentials and creates a session for the user if authentication is successful.

iv. **IdP Issues SAML Assertion:** After successful authentication, the IdP generates a SAML assertion, which is a structured XML document containing authentication information about the user, such as the user's identity, attributes, and any roles or permissions. The assertion can also include an expiration time for the authentication and the assurance level of the identity verification process.

v. **Assertion Sent to SP:** The IdP sends the SAML assertion back to the SP, usually by redirecting the user's browser to a pre-defined URL at the SP (Single Sign-On (SSO) URL). The assertion is often signed and may be encrypted to ensure integrity and confidentiality during transmission.

vi. **SP Validates the Assertion:** The SP receives the SAML assertion and verifies it by checking the digital signature and ensuring it matches the expected

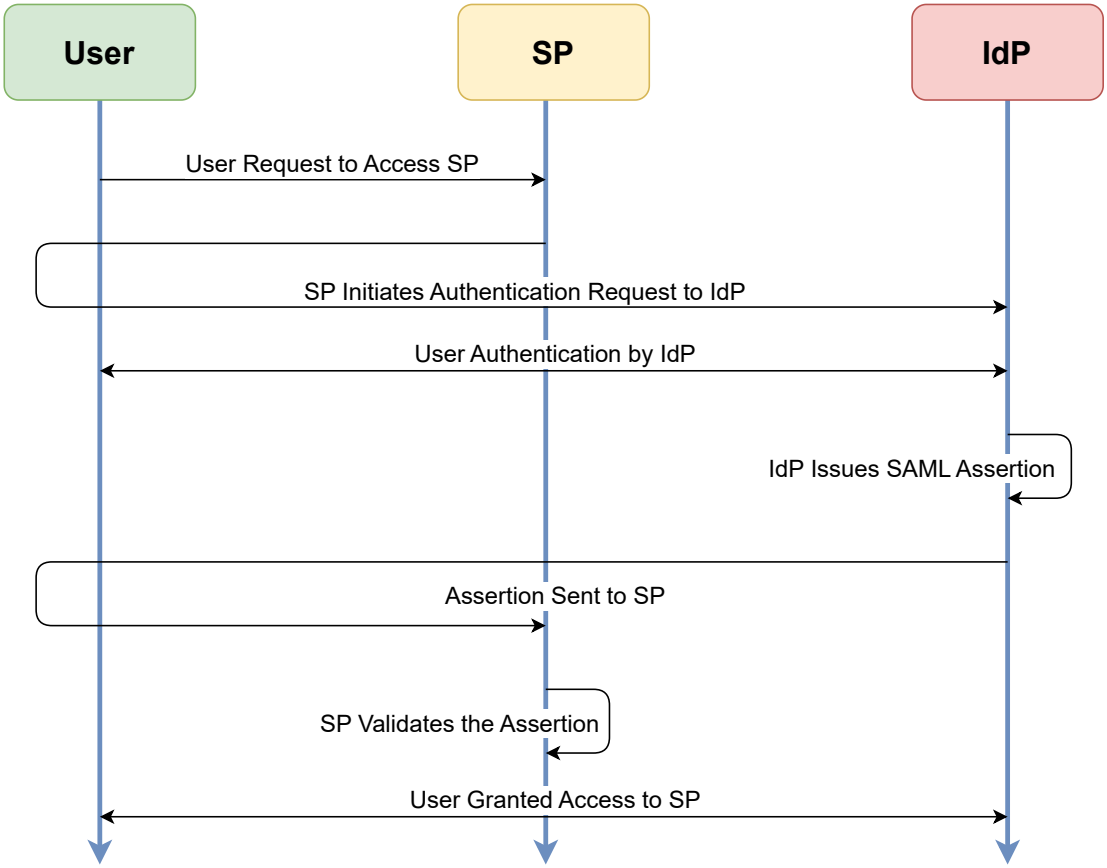

Figure 2.1: SAML protocol flow

information (e.g., the identity of the IdP, timestamps, and target URL). The SP also confirms that the assertion has not expired and that it includes the necessary information for service authorization.

vii. **User Granted Access to SP:** Once the assertion is validated, the SP grants the user access to the requested service based on the information provided in the assertion. The SP may create a session for the user to maintain their authentication state across subsequent interactions within the same session.

2.5.4 Single Sign-On (SSO) in Federated Identity Management

Single Sign-On (SSO) is one of the most important features of federated identity management because it lets each user log in once and then appear to be able to log into multiple related systems without having to reenter separate credentials for each one. SSO simplifies the user experience because it allows users to access different systems without having to remember different credentials, which ensures security in any system. Heres how SSO typically works within a federated identity model:

- The user enters their credentials into a primary Identity Provider (IdP).

- If the login is successful the IdP issues a session token or authentication assertion.
- When the user requests access to other SPs services, this token will be passed to them.
- Each Service Provider checks the token against the IdP's validation and allows the user access without needing additional authentication.

With SSO, the friction is reduced for multiple logins and the user experience is improved; federated identity management also ensures that each system and organization trusts the same set of credentials. By factoring in the authentication and authorization processes, users get access to multiple services easily, while organizations can keep their security posture strong by having the authentication tasks delegated to trusted identity providers.

Chapter 3

Literature Review

In this chapter, we present a summary of relevant research works and provide a comparison between different papers in Table 8.2.

The research paper [40] focuses on the issue of technologies of Metaverse and the applicability of non-fungible tokens (NFT) to safeguard copyrights in this virtual world. In the context of this paper, various threats and opportunities that appear within the evolving digital environment will be discussed in detail, especially regarding the security and authenticity aspects of digital transactions within the Metaverse. The authors highlight the privacy and ownership, authenticity concerns in the Metaverse where standard legal concepts such as copyright and ownership are being tested. This research aims to advance a more secure and protected quantum-based NFT solution that will also safeguard copyrights in Metaverse. Furthermore, the paper discusses the basic technologies underpinning NFTs, including blockchain technology and smart contracts, and their importance in providing copyright protection. It also describes the approaches and the methods that NFT marketplaces have adopted to enhance security, prevent copyright violations, and ensure that the Metaverse fosters respect for intellectual property rights. Furthermore, the paper presents the concept of quantum hypergraph states as well as its possibility to apply them to the construction of secure blockchain systems, offering an alternative approach to traditional ledger and hash functions. In conclusion, the given research paper offers relevant information on the use of Non-Fungible Tokens (NFTs), blockchain technology, and copyright protection in the context of the Metaverse. It offers a comprehensive analysis of the security measures implemented by NFT marketplaces in different countries, highlighting the advancements and challenges in this rapidly evolving landscape. The proposed quantum NFT solution presents a novel approach to enhancing security and ensuring copyright protection in the Metaverse, paving the way for future studies and implementations in the field.

Hong et al. [30], published an eye-blinking EOG user authentication system based on a blockchain, Ethereum, and Non-Fungible Token (NFT). The study suggests that the decentralized system based on blockchain technology can solve the problems associated with the use of traditional methods of user identification. The study proposes to implement blockchain technology to eliminate the challenges linked to ordinary user authentication methods. The article presents a method for the storage distribution of biometric factors of users identification based on NFT and IPFS, implemented through the Ethereum platform. The paper's key contributions include providing a safe mechanism for handling user biometric data, showing the

registration and authentication procedures in the proposed model, and assessing its effectiveness. They efficiently incorporate eye-blinking EOG data as a secure authentication technique while also separating the service and storage environments to improve security. Furthermore, it discusses how to utilize the Ethereum blockchain, NFT, and IPFS for distributed storage; and how biometric information can be fabricated or falsified. In conclusion, the research emphasizes the effectiveness of applying blockchain technology and NFT for ensuring the safety, openness, and credibility of login procedures for customers. It describes how blockchain can address the limitations of the current approaches to authentication and guarantee a reliable outcome for accredited user authentication and data validation.

Fauzi et al. [22], introduced a new approach to the integration of Non-Fungible Tokens (NFTs) in enhancing the security of Single Sign-On (SSO) systems based on OpenID Connect and OAuth 2. 0 protocols. The study assesses todays security concerns with the existing SSO systems, with the primary focus on the IDP. Introducing the idea of applying NFTs as IDC ownership proof allows the authors to aim at enhancing the security levels and usability of the employed IDPs authentication process. The concepts considered in the paper are the SSO application with the necessary options for registration, password recovery, and log-in; the model and architecture for NFTs integration with SSO; and the explanation of the login process and NFT operation on OIDC and OAUTH 2. 0. The authors offer very vivid illustrations of how NFTs help overcome typical weaknesses, including ID spoofing, key confusion, etc., in conventional SSO platforms. This paper also raises a concern about the lack of study on scalability, real-world testing, and a comparison with other SSO solutions. In general, the paper discusses the benefits of implementing blockchain-based NFT to eliminate shortcomings of conventional SSO. It shows how with NFT, the concept of digital identity can be altered significantly and offers a realistic approach to how the level of trust and security in user authentication may be improved. From the study displayed it is observed how the current use of blockchain technology is capable of handling the management of the authentication methods and providing a reliable solution towards the removal of existing methods which hampers the trust and data integrity of the users.

Banaeian et al. [34], explored the use of Non-Fungible Tokens (NFTs) as a tool for managing identities in metaverses. It Solves the challenges of managing distributed identities in metaverses, like users having multiple different identities to manage, providing certificates for authentication on metaverses, and handling security and privacy issues. Namely, it shows that we ultimately need to reconcile appropriate levels of decentralization with the idea that a non-regulated system will never be allowed at an institutional/centralized level. It explores how identity and authentication are critical and discusses the importance of using non-fungible tokens (NFTs) for identity management in digital worlds, especially metaverses. The research delves into the utilization of NFTs for assigning unique digital identities to users in metaverse environments, emphasizing their advantages such as transparency, immutability, and verifiable proof of ownership. It also addresses issues related to keeping an eye on identities in a distributed system and managing them without any vogue. Moreover, the proposal introduces the cryptographic tools as their own solutions to fix these problems. Additionally, it highlights the multitude of possibilities metaverses carry with their wide adoption, such as the potential for

job creation and investments, as well as how digital twins can be used to enrich realistic experiences.

Guidi et al. [35], explored the societal implications of Non-Fungible Tokens (NFTs) in the context of the Metaverse economy. The authors delve into the use of NFTs to represent virtual assets and properties in the Metaverse, focusing on platforms like Decentraland and The Sandbox. increasing prices of Metaverse parcels over time, making acquisition prohibitive and fueling a speculative market similar to physical real estate. It identifies factors driving parcel prices, such as proximity to landmarks, roads, squares, and parcels owned by influencers or enterprises. The analysis uncovers trends in parcel sales, emphasizing the impact of NFTs on the Metaverse estate economy. The paper suggests further research on the social impact of NFTs in the Metaverse economy, particularly in understanding the driving forces behind parcel prices and the implications for wider adoption of Metaverse platforms. The study provides valuable insights into the evolving landscape of the Metaverse and the role of NFTs and blockchain technology in shaping its economy.

Tsegaye et al. [38], presented a comprehensive strategy for combating counterfeit products using blockchain technology and Non-Fungible Tokens (NFTs). They propose a system that leverages decentralized networks and NFTs to securely authenticate product information. The paper provides a clear outline of the implementation steps for the proposed system, explaining how NFTs and QR codes are utilized to authenticate products and prevent the entry of counterfeit goods into the market. Additionally, the paper discusses the technology and tools employed in developing the system, including ReactJS, Bootstrap, Solidity, Hardhat, OpenZeppelin, and a MongoDB database.

Kim et al. [36], described the method of user authentication in the avatar with the help of DID (Decentralized Identifiers) and SBT (Soul Bound Tokens). The system provides an innovative approach to verifications of identity without revealing personal details from the blockchain through Zero-Knowledge Proofs. In addition to this, it also embraces the privacy aspect of the contract because it can be made in such a way that some information can be concealed while others are revealed. However, some of the limitations stated in the paper include the areas of technicality that include the need for a trusted issuer for creating DID credentials and compatibility issues between the wallets address and DID credentials. In addition, the room for improvement in relation to the existing methods of KYC can be established based on the comparison with the proposed system. In conclusion, the paper enhances the progressive development of decentralized identification frameworks integrated into Web3 solutions.

Manzoor et al. [37], provided a comprehensive insight into the development of the framework that overcomes the problem of authenticity, integrity, interoperability, and efficiency of Metaverse applications. In this paper, a novel security model that leverages Non-Fungible Tokens (NFTs) and Proof-of-Stake (PoS) consensus within the blockchain scenario is proposed to improve the security and effectiveness of data transactions. An important finding that distinguishes the paper is the deployment of NFTs for user verification and PoS for data validity in a Metaverse context, thereby resulting in the optimization of the overall value of throughput, scalability, and security of Metaverse applications. All in all, by integrating all these technologies, the proposed work presented here shows the merits and effectiveness of

the framework in defending and optimizing decentralized applications security and performance. In addition, the authors discuss more details about each technology area and its importance in constructing the framework. In this work, by performing an experimental analysis for Ethereum augmented with smart contracts and NFTs, and Proof of Stake consensus algorithm, the practical implications use and quantitative efficiency of the proposed framework have been demonstrated. Evaluations of the throughput and scalability show that the proposed NFT-based blockchain solution is the best fit for the Metaverse given its capability in maintaining data authenticity, data integrity, and data interoperability. All in all, the paper provides a clear and coherent discussion on the integration of NFTs, blockchain technology, and PoS consensus model in enriching Metaverse applications. As a result, the presented framework does not only respond to the current issues but also offers potential for future development in security and extendibility within the Metaverse domain. Thus, the paper is a highly useful contribution to mapping the trends and developments in using blockchain-based security models in the novel generation of decentralized applications. In Table 8.2, a comparison between key aspects has been highlighted for a better understanding of the discussed paper.

Chapter 4

Proposal

In this section, we present our proposal (Section 4.1) and a threat model (Section 4.2) and then analyze many functional and security requirements (Section 4.3.1)for the proposed Secure authentication mechanism for metaverse Using Non-Fungible Token.

4.1 System Proposal

Our goal in this research article is to employ a non-fungible token for revolutionary Metaverse authentication.

The comparison Table (Table 8.2) illustrates various strategies and techniques for authenticating the Metaverse Application. However, It has a few lacking while providing a complete and coherent solution using NFTs to address all the significant issues like security, usability, scalability, etc. The Table (Table 8.2) indicates several research efforts have focused on specific topics such as enhanced security, production verification, and identity management, but none of them offered an overall integrated approach that effectively combines various components.

To bridge this research gap, our proposal aims to create a reliable and trustworthy authentication mechanism utilizing NFTs to secure identities, ensure scalability, and provide transparency in the Metaverse. One of the significant aspects that we present is the concept of Federated Metaverse, which expands on the notion of federated identity systems to enable smooth transitions between multiple Metaverse. In a federated Metaverse, several Metaverse platforms will use a single identity management and authentication system to create a Circle of Trust (CoT). This idea solves a significant research gap: users are frequently limited to specific platforms, necessitating distinct authentication procedures. It facilitates interoperability across multiple Metaverse and allows users to carry their identity and credentials between platforms without re-authenticating repeatedly. This strategy offers a reliable, decentralized way to manage identities across several virtual environments, which not only increases user convenience but also strengthens security.

Our proposal intends to establish such federated architecture by utilizing NFTs in combination with blockchain technology. By doing this, we guarantee that users may engage across multiple Metaverses in a smooth manner while adhering to platform-specific guidelines and standards and preserving a reliable, consistent identity.

To overcome these shortcomings, our proposed approach using NFTs to develop a strong authentication system for metaverse applications that will guarantee:

i. **Complete identity management:** Each user in the metaverse application will be given a unique, unchangeable identity that will be secure and verified using NFTs.

ii. **Improve Security and scalability:** To improve the security and scalability in metaverse applications, we can combine NFTs along with some cutting-edge blockchain protocols like Proof of Stakes (PoS).

iii. **Information and product verification :** making use of NFTs in order to verify the legitimacy of digital assets and to ensure the safety and transparency of all ownerships and trade activities within the Metaverse environment.

iv. **Federated Identity for Metaverse:** By employing a single authentication mechanism, our method will allow users to move between several Metaverse in a federation. We want to establish a Circle of Trust (CoT) that facilitates secure and uniform identity management across Metaverses by utilizing NFTs and a common registry across many platforms. This ensures users have a reliable and easy transition across multiple Metaverse without compromising security or guidelines to platform-specific guidelines.

By combining these components, our solution will provide a scalable, secure, and intuitive authentication system for metaverse applications.

4.1.1 Methodology

There are six necessary steps based on the design science research methodology. The Design Science Research Methodology (DSRM) is an organized process for creating and assessing artifacts to solve recognized issues or difficulties, specifically within the fields of information systems and related fields [1]. The following steps we need to follow to propose research are:

1. **Threat modeling:** In the initial step, we will determine and evaluate possible security risks and vulnerabilities related to NFT-based Metaverse authentication.

2. **Requirement Analysis:** In the requirement analysis step, we identify and analyze the functional and non-functional requirements essential for the secure authentication mechanism in the metaverse using NFTs.

3. **Architecture Design:** In this step, the overall work plan of the assigned literature for developing an authentication system will be designed. Key components and their interactions are defined to ensure a secure and efficient system.

4. **Protocol Flow:** The protocol flow step outlines the detailed steps involved in the NFTs authentication process.

5. **Implementation:** In the implementation phase, the designed architecture and protocols are developed and integrated into the metaverse system.

6. **Testing & Evaluation:** The last step is the testing and validation process where the implemented system is tested to see if it meets all the requirements and functions optimally.

In Figure 4.1, A visual representation is given for the methodology steps for better understanding.

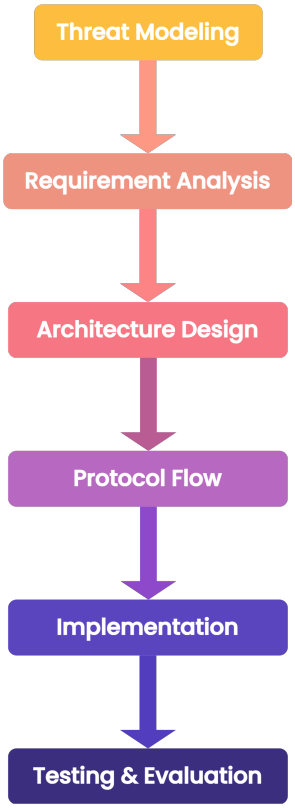

Figure 4.1: Methodology Steps

Here, we first examine the ideas of Blockchain, Non-Fungible Tokens, Metaverse, and authentication mechanisms in order to properly grasp the topic. We have therefore read a number of books and research articles in order to be able to convey the central idea that is necessary for the thesis. Next, we discovered an issue with the current authentication protocols, particularly with online services like Metaverse.

In the next sub-section, our focus is mainly on Thread modeling and Requirement Analysis.

4.2 Threat Modeling

Threat modeling is a way of utilizing a model to evaluate threatening factors that can cause security issues in any system [11]. This way it also enhances the strength of the systems as any faults are pinpointed before the faults are even put in place. Since our work is dedicated to using NFTs for authentication in the metaverse, where identity management plays a vital role, threat modeling is required. In order

to categorize the threats we have identified, we have selected the STRIDE threat model [11], which is quite popular and which preserves the legitimacy, improbability, confidentiality, availability, integrity, and authorization of your system.

T1 **Spoofing:** In metaverse, intruders may steal login information or use some other means to steal the metaverse identity and avatar of a user.

T2 **Tampering:** Intruders may modify tokens or other data that may be linked to an NFT, together with the ownership of digital assets or the details of the tokenization methodology. This might further erode confidence in the use of the NFT for purposes of validation and genuineness.

T3 **Repudiation:** In an open world like metaverse, users may reject operations such as NFT transfer or activity in the virtual society. This can be a challenge when it comes to holding someone responsible and keeping records of historical events on a blockchain since some blockchains do not offer good logging or auditing.

T4 **Information Disclosure:** In metaverse, identities of users, ownership of an NFT, and the authentication details could get leaked accidentally. This can be due to weak encryption, weak access control, insecure APIs, or logging mechanisms that compromise user privacy and hence hinder the adoption of an NFT-based authentication system.

T5 **Denial of Service (DoS):** It can be used to disrupt services and impede the functioning of NFT marketplaces and metaverse platforms, blocking users and their avatars as well as their transactions with each other and between them. This can be done by providing too much traffic to the network so as to cause congestion or by identifying a likely time when the system vulnerabilities are likely to be exploited.

T6 **Elevation of Privilege:** Specific weaknesses may include system vulnerabilities that the hackers can exploit to gain higher privileges or in the process of creating, modifying, or deleting NFTs or even interfering with authentication procedures. This can become a major issue of concern because it may compromise the stability as well as security of the general metaverse authentication system [18].

4.3 Requirement Analysis

Before implementing the model, it is necessary to carry out a requirement analysis. It is also important to note that prior to the implementation phase, it is helpful to discuss and assess functional and non-functional or security requirements to avoid future problems that may hinder us from doing our work effectively. Functional requirements state what is required of the system to be effective and efficient while nonfunctional requirements state security issues and other important aspects of the system.

4.3.1 Functional requirements

This sub-section outlines the capability and essential components of our proposed system is represented by functional requirements.

- **F1:** Implementing an NFT-based authentication to preserve the privacy of metaverse users.
- **F2:** Developing a blockchain middleware to authenticate user interaction with the system as well as mitigate centralization problems.
- **F3:** Ensuring various metaverse platforms can integrate properly with the authentication system.
- **F4:** Enabling users to authenticate using authentication NFTs from their digital wallet (E.g. Metamask).

4.3.2 Security requirements

This sub-section outlines the capability and essential components of our proposed system represented by security requirements.

- **S1:** As we are using NFT-based identity management, access to the services will be strictly given only to those who have verified and genuine NFT credentials, thus mitigating T1 and T6.
- **S2:** Implementing blockchain technology that will track every authentication transaction in the chain should make data tempering difficult and ensure accountability for every transaction. This will mitigate T2 and T3.
- **S3:** Ensure to filter out and mitigate all the requests that do not have a proper and valid NFT credential and thereby mitigate T5.
- **S4:** Use secure encryption methods for the exchange and storage of data which should mitigate T4.

Chapter 5

Architecture

In Figure 5.1, we present an architectural representation of our proposal. The diagram illustrates a system where a user engages with a Metaverse Federation to obtain services. The flow can be demonstrated through various situations, depending on the user's registration status and their current NFTs recorded inside the Blockchain system generated via the Metaverse federation or certain Metaverse. The architecture gives users flexibility by enabling them to authenticate with several NFTs in different Metaverses, all the while preserving the process's security and integrity through an integrated trusted body.

The application makes use of a public blockchain in order to user authentication via providing a transparent and decentralized approach to manage and validate the NFTs in order to authenticate the user produced through the Metaverses within the federation. It will allow the user's credentials to be accessed and validated in all participating Metaverses. Users will get full visibility of the issued NFTs for them by different Metaverse. As long as the NFT was issued by a reliable Metaverse or the service provider Metaverse, the user will be allowed to choose any of them for authentication.

The ERC-721 standard non-fungible tokens (NFTs) are used in each Metaverse in this federation to authenticate and identify users. This guarantees compatibility across Metaverses and streamlines the process of confirming the legitimacy and ownership of NFTs on various platforms. The federation's trust requirements are fulfilled by using ERC-721, which ensures that each user's NFT is distinct, non-fungible, and safely maintained on the public blockchain.

In order to develop trusting connections with other Metaverses, each Metaverse in the federation creates a trustee body. The initial protocol for the Metaverse does this by establishing a registry table shown in Figure 5.1. The registry table contains a list of various Metaverses that can attest to a user's identity by verifying the NFTs generated via those Metaverses. This serves as the basis for a circle of trust (CoT) in which various Metaverses serve as one another's Identity Providers (IdP).

The authentication process begins when a user sends a request to a service provider, in this case of Figure 5.1, Metaverse 1 (M1), to access specific services. To verify the user's authenticity, Metaverse 1 (M1) first sees if they have an NFT. If the user is not verified by Metaverse 1 (M1), It consults its trusted partners, Metaverse 2 (M2) and 3 (M3) to see if the user has been authenticated. If none of the metaverses

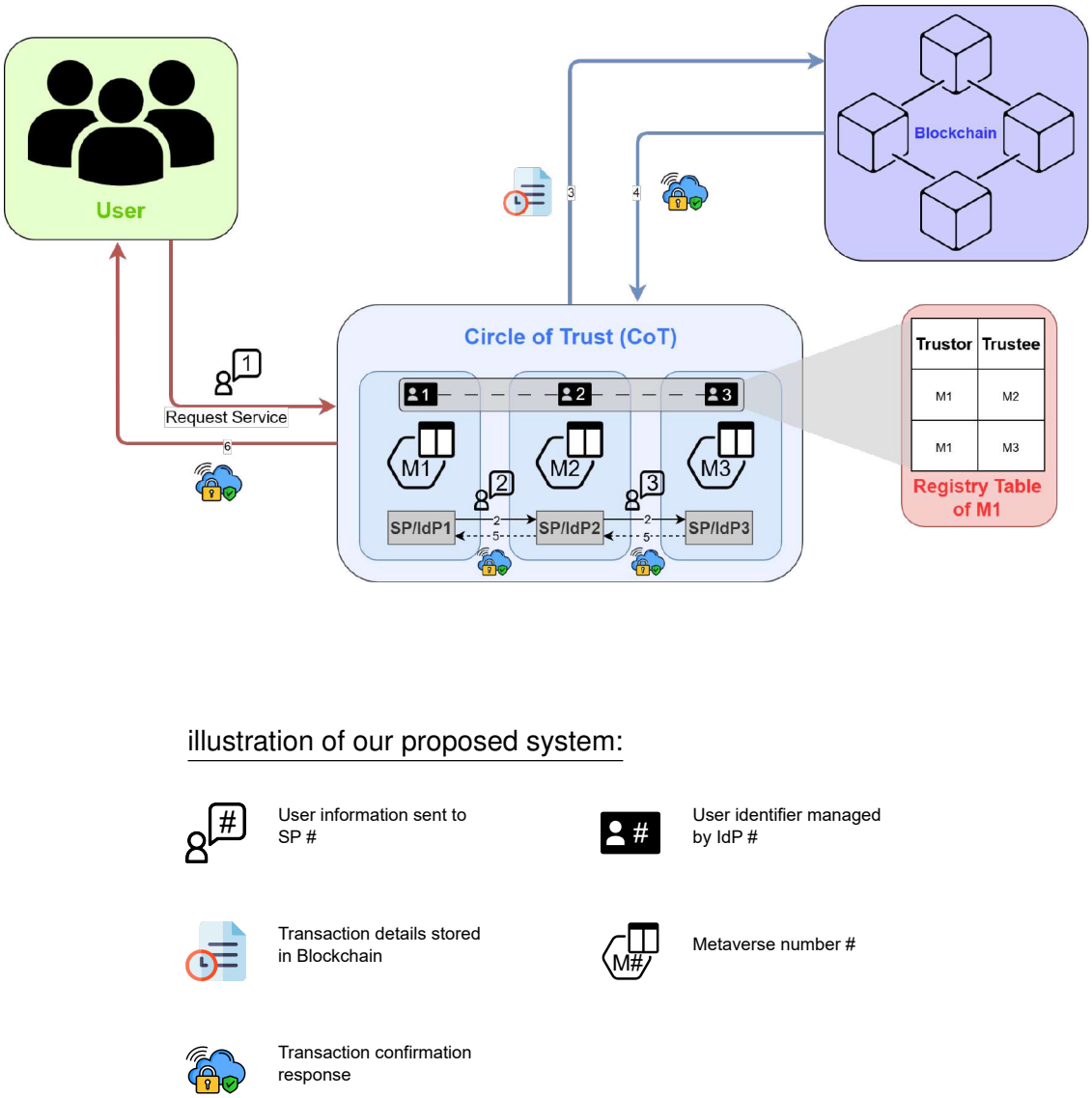

Figure 5.1: Architecture of the proposed system

have verified the user, the user can request to create an NFT and be authenticated by any of the metaverses acting as identity providers. When the NFT is generated, a transaction is recorded on the blockchain proving the ownership between the user and the NFT. This ownership is verified by the blockchain and sends a response back to the metaverse. The verified ownership of the NFT is then transmitted back to Metaverse 1 (M1) and ultimately returned to the user, completing the authentication process and allowing the user access to the requested services.

This architecture provides a secure, adaptable, and transparent user authentication process throughout a Metaverse Federation by utilizing the feature of public blockchain system and NFTs that comply with ERC-721. The relationship of Trust between Metaverse holds in individuals' registry tables provides redundancy and

improves users' experience, by ensuring users can authenticate using pre-existing NFTs. By utilizing the structure, the Metaverse Federation can offer seamless service access while preserving the security, confidentiality, and authenticity of users' identities.

Chapter 6

Use-case & Protocol Flow

We show the use cases with their protocol flows of our proposed architecture in this section. We have discussed and elaborated the data model in (Section 6.1). Then we have shown some specific alforithm in anohter subsection. Moreover, we have discussed different protocol flows in (Section 6.3), (Section 6.4), (Section 6.5), (Section 6.6). We have also mentioned all the cryptographic notation that have been used in each protocol flow in (Table 6.1) and they are self explanatory.

Notations	**Description**
U	User
MV_i	i^{th} Metaverse of federation
SC_i	Smart Contract of Blockchain or Registry Table or Federation
$Signature_i$	cryptographic proof of i^{th} Metaverse Admin or Users
$NFT^{U}_{MV_i}$	NFT of MV_i for U
$NFTs^{U}_{TruesteeMV}$	All the NFTs of the user issued by trusted Metaverse in the federation
$SignedCredentials^{U}_{MV_i}$	Signed credentials of a user by i^{th} Metaverse
$WalletAddress_i$	Wallet Address of i^{th} Metaverse of the federation or the User
URL_i	A URL Address of i^{th} Metaverse of the federation
N_i	A fresh Nonce
$[\dots]_K$	Communication over a channel encrypted with key K
$[\dots]$	Communication over an unencrypted channel
$[[\dots]]$	Optional data

Table 6.1: Cryptographic Notations

6.1 Data Model

In table 6.2, the data model is designed to facilitate a system to handle request (req) and response (res) of different operations such as authentication, the minting of NFTs, the verification of credentials, and Metaverse registration. Each of these requests causes a particular response: registerMetaverseReq, getURLReq($URLmv$), addTrusteeReq, SignatureReq, serviceReq, registryCheckReq, WalletAddressReq, NFTReq, addUserAllowenceReq, isUserAllowenceReq, registerReq, NFTMintReq, NFTMetadataReq.
All these requests cause similar responses: registerMetaverseRes, getURLRes($URLmv$), addTrusteeRes, SignatureRes, NFTRes, serviceRes, registryCheckRes, NFTMintRes,

WalletAddressRes, addUserAllowenceRes, isUserAllowenceRes, NFTMetadataRes, registerRes. Each request triggers a corresponding response, ensuring that tasks are executed securely and appropriately.

The *serviceReq* initiates a request using a URL and an optional *inviteCookie*, while the *serviceRes* provides access to the relevant service. The *registerReq* includes a service provider URL (url_{MV_i}) and a nonce (N_i) to initiate the registration process, with the *registerRes* confirming the registration. The *credentialReq* seeks credentials, and *credentialRes* returns a verifiable credential (VC_{MV_i}) issued by the metaverse.

User credentials (such as email and username) are managed in credentials, which may also include *mvCredentials* that the metaverse uses for identity purposes. The $NFTMintReq$ involves a request for minting an NFT and includes NFT details. Upon minting, the $NFTMintRes$ generates and returns the NFT token ($NFT^U_{MV_i}$).

req $\triangleq$ ($registerMetaverseReq, getURLReq_{(URL_{mv})}, addTrusteeReq, SignatureReq,$ $NFTReq$, $serviceReq$, $registryCheckReq$, $WalletAddressReq$,$NFTMintReq$, $addUserAllowenceReq$,$isUserAllowenceReq$,$NFTMetadataReq, registerReq$,)
res $\triangleq$ ($registerMetaverseRes, getURLRes_{(URL_{mv})}, addTrusteeRes, SignatureRes,$ $NFTRes, serviceRes, registryCheckRes, WalletAddressRes, addUserAllowenceRes,$ $WalletAddressRes, isUserAllowenceRes$,$NFTMetadataRes, registerRes,$ $NFTMintRes$)
$registerMetaverseReq \triangleq$ (mvPublicAddress)
$mvSignatureReq \triangleq$ (mvPublicAddress, mvTrustRegistryAddress)
$mvCredentials \triangleq$ (mvPublicAddress, URL_{mv})
$addTrusteeReq \triangleq$ (mvPublicAddress, mvTrustRegistryAddress)
$addTrusteeRes \triangleq$ (Boolean)
$addUserAllowenceReq \triangleq$ (mvPublicAddress)
$addUserAllowenceRes \triangleq$ (Boolean)
$registerReq \triangleq$ (url_{MV_i})
$registerRes \triangleq$ ($NFT^U_{MV_i}$)
$credentials \triangleq$ (email, username)
$NFTMintReq \triangleq$ (Name, Email, mvPublicAddress, Signature)
$NFTMintRes \triangleq$ (Minted NFT token, $NFT^U_{MV_i}$)
$NFTReq \triangleq$ (Request NFT for verification)
$NFTRes \triangleq$ (Verified NFT response, $NFT^U_{MV_i}$)
$registryCheckReq \triangleq$ (Identity check request, MV_i)
$registryCheckRes \triangleq$ (Registry verification response)

Table 6.2: Data Model

The $NFTReq$ sends a request to verify an NFT, while the $NFTRes$ provides the corresponding verification. The $registryCheckReq$ involves sending identity check

requests to the registry (MV_i), and the corresponding *registryCheckRes* delivers the verification response. Furthermore, a metaverse registration can be initiated through the $registerMetaverseReq$, while the $registerMetaverseRes$ confirms the registration. Then, with $addTrusteeReq$ adding a new trustee request, and the $addTrusteeRes$ confirms the addition of the trustee.

This comprehensive data model ensures secure handling of metaverse credentials, NFT minting, and verification processes, while facilitating communication between users, the metaverse, service providers, and the blockchain.

6.2 Algorithm

The section lists the essential algorithms whose functionality supports the functions of the entire system, allowing the safe management of digital assets and the decentralized management of governance. Similarly to how Identity Provider and Service Provider can be able to establish a cryptography trust and metadata sharing to enable secure authentication flows, those are algorithms that facilitate verifiable operations in our architecture. We shall enter into Algorithm 1 outlining the secure minting process of the Non-Fungible Tokens; Algorithm 2, which describes how the metadata of NFTs is retrieved depending on their ownership; and Algorithm 3, which describes the secure and transparent voting mechanism of federation membership. By themselves, these algorithms establish the functional core of exclusive communication and decentralized decision-making within the system.

Algorithm 1 `safeMint` Function

```
1: function SAFEMINT(address to, string memory name, string memory email,
   bytes memory signature)
2:     bytes32 hash = keccak256(abi.encodePacked(to, name, email));
3:     bytes32 message = MessageHashUtils.toEthSignedMessageHash(hash);
4:     address recovered = ECDSA.recover(message, signature);
5:     uint256 tokenId = _nextTokenId++;
6:     TokenMetadata memory metaData;
7:     metaData.name = name;
8:     metaData.email = email;
9:     metaData.id = tokenId;
10:    metaData.creator = recovered;
11:    _tokenMetadatas[tokenId] = metaData;
12:    _setTokenURI(tokenId, "https://drive.google.com/uc?export=download&id=
13:    1B5zU_BwuL3-rH_qm44qLsrfvULkRvI-g");
14:    _safeMint(to, tokenId);
15: end function
```

The Algorithm 1 is responsible for minting a new Non-Fungible Token (NFT) safely on the blockchain. It accepts multiple parameters namely *to* (the wallet address of the user), *name* (the name that is tied to the NFT), *email* (the email that is linked to the NFT), signature (a cryptographic signature of the metaverse that authorizes the minting request).

Then the process is initiated by generating a unique hash of the recipient *address*, *name*, and *email* with the help of $keccak256(abi.encodePacked(to, name, email))$. Through $ECDSA.recover(message, signature)$ one can recover the address of the signer with this message hash and the provided signature. Furthermore , an Ethereum-signed message hash is obtained by $MessageHashUtils.toEthSignedMessageHash(hash)$. This is one of the necessary steps in recovery that confirms the minting request is made by the true creator or a specified party.

An incremental $\_nextTokenId$ is created into a new $tokenId$. After that, a structure named $TokenMetadata$ is created that is filled with the provided and generated information being inserted into its *name*, *email*, and *id* variables. The recovered address is also taken as the creator of this particular $tokenId$ in the $TokenMetadata$. Then, such metadata is stored in the $\_tokenMetadatas$ mapping to ensure the association between $tokenId$ and its information.

Then, The $\_setTokenURI$ method is called and is used to set a Uniform Resource Identifier (URI) to the newly minted token that is to reference a Google Drive link that includes the on-chain metadata or asset of the NFT. Lastly, the main $safeMint(to, tokenId)$ method is called to record the official change of ownership to to address in the blockchain in regards to create new NFT. The whole process of this provides the necessary level of security, since the only entities will be allowed to create new tokens of the NFT, and all the corresponding metadata will be attached and saved appropriately.

Algorithm 2 getNftMetadataByAddress Function

```
1:  function getNftMetadataByAddress(address walletAddress)
2:      uint256 nfKeysCount = balanceOf(walletAddress);
3:      TokenMetadata[] memory nfKeys = new TokenMetadata[](nfKeysCount);
4:      uint256 currentIndex = 0;
5:      for uint256 i = 0; i < _nextTokenId; i++ do
6:          if ownerOf(i) == walletAddress then
7:              nfKeys[currentIndex] = _tokenMetadatas[i];
8:              currentIndex++;
9:          end if
10:     end for
11:     return nfKeys;
12: end function
```

The Algorithm 2 allows getting all the NFT metadata tracked by a specific *walletAddress*. It is a view function, which implies that it does not have any effect on the state of the block chain and can call it without gas cost.

The following function begins by counting the total amount of NFTs possessed by the *walletAddress* via the $balanceOf(walletAddress)$. This number is further employed to create a dynamic array of $TokenMetadata$, named $nfKeys$, which is going to hold metadata information of all NFTs owned by the given address. A $currentIndex$ variable is set to allow the tracing of the current position within the array of the $nfKeys$.

Then the function loops over all active $\_nextTokenId$ (that is, all the issued tokens

on the contract). It makes sure that the $ownerOf(i)$ (the person who holds value of any token i) is the same as the $walletAddress$ being used. In the event that it matches, then this NFT is owned by the $walletAddress$. When that happens then the TokenMetadata of that token at the $\_tokenMetadatas[i]$ mapping is copied to the $nfKeys$ array at the $currentIndex$, and then the $currentIndex$ is incremented. This process goes on untill it checked all the token.

Lastly, the algorithm returns the $nfKeys$ array that is filled with all $TokenMetadata$ of the NFTs in the possession of the $walletAddress$ now. This functionality is then a sort of query to the world to know all the NFTs that belongs to a particular user on the smart contract.

Then, The Algorithm 3 specifies an important procedure to control membership in a decentralized federation or governance system. It is mainly used to give current members a chance to vote on a request and this is usually when a new person is to join. The function takes two important inputs, namely the $candidateAddress$ of the entity or a person applying to join the association together with a $boolean$, approve which indicates whether the present voter approves ($true$) or rejects ($false$) the candidate.

The function carries out some critical checks before the vote can be considered valid, using several crucial checks as the first stage of the invocation. It will ensure that only an authenticated member of the federation ($msg.sender$) may use it, without allowing illegal ways into participation. It next confirms that the targeted vote request of the $candidateAddress$ is not concluded yet and that the request is active. A major protection against multiple votes is provided which is that the function processes through the list of past voters on that request and in case the current caller is already a voter, then the transaction is rolled back.

There is also a time-based mechanism, whereby a request is considered expired in case the current blockchain time ($block.timestamp$) is greater than 7 days after the time when the vote request was issued ($request.timestamp$). In the case of this, the request is made inactive, an event is posted to indicate its expiration, the candidate is removed any active candidate lists, and the data of the request is destroyed thereby cancelling the vote.

When all checks are pass, the address of the voter ($msg.sender$) is added on to the voters list of that request. The vote count is then updated, that is, when the approve parameter is set to $true$, the $yesVotes$ count of the candidate gets increased; otherwise, the $noVotes$ count increases. After the vote, the role immediately determines whether enough votes have been cast to achieve a $quorum$ or not. It sums up the $totalVotes$ (both the yes vote and no vote) and the $quorumVotes$ is calculated with a certain $quorumPercentage$ of the total active members. In case that the $totalVotes$ are equal to or greater than the number of $quorumVotes$ as calculated (meaning that sufficient members have voted to render a decision), the vote request is set as inactive. The running mate is not also left in any list of limited running mates to be voted.

At this point, the final decision is made:

- If the candidate is admitted successfully in case the $yesVotes$ exceed the $noVotes$. They are added to the $members$ list of the federation, their $isMember$

Algorithm 3 voteOnRequest to add into the federation Function

```
1: function VOTEONREQUEST(address candidateAddress, bool approve)
2:    require(isMember[msg.sender], "Only federation members can
   vote");
3:    VoteDetails storage request = voteRequests[candidateAddress];
4:    require(request.active, "Vote request is not active");
5:    for uint i = 0; i < request.voters.length; i++ do
6:       if request.voters[i] == msg.sender then
7:          revert("Already voted");
8:       end if
9:    end for
10:   if block.timestamp > request.timestamp + 7 days then
11:      request.active = false;
12:      emit VoteExpired(candidateAddress);
13:      removeCandidateFromVoteCandidates(candidateAddress);
14:      delete voteRequests[candidateAddress];
15:      revert("Vote request expired");
16:   end if
17:   request.voters.push(msg.sender);
18:   if approve then
19:      request.yesVotes++;
20:   else
21:      request.noVotes++;
22:   end if
23:   uint totalVotes = request.yesVotes + request.noVotes;
24:   uint totalMembers = members.length;
25:   uint quorumVotes = (quorumPercentage * totalMembers + 99) /
   100;
26:   if totalVotes >= quorumVotes then
27:      request.active = false;
28:      removeCandidateFromVoteCandidates(candidateAddress);
29:      if request.yesVotes > request.noVotes then
30:         address candidate = candidateAddress;
31:         members.push(candidate);
32:         isMember[candidate] = true;
33:         metaverses[candidate] = Metaverse(wallet:  candidate,
   websiteUrl:  request.websiteUrl);
34:         emit MemberAdded(candidate);
35:      else
36:         delete voteRequests[candidateAddress];
37:      end if
38:   end if
39: end function
```

property is changed to *true*, and other metadata containing their wallet address and their weblink URL are stored. An event called $MemberAdded$ is subsequently published to signal external systems of the new member.

- In-case $yesVotes$ are not larger than $noVotes$ ($noVotes$ are not smaller than $yesVotes$), a candidate is rejected, and their related vote request information is deleted in the system.

The above detailed process makes all membership changes to be democratic, safe and compliant with set rules and schedules.

6.3 Federated Metaverses Registration Protocol Flow and Use Cases

To be part of a federation, Metaverses must register in the federation registry. Metaverses can register themselves by following the protocol outlined in (Table 6.3):

Steps	Description
M1	$MV_1 \rightarrow SC_{Federation}$: [N1, $registerMetaverseReq$, $mvCredentials$]$_{\text{HTTPS}}$
M2	$SC_{Federation} \rightarrow MV_1$: [N2, $SignatureReq$]$_{\text{HTTPS}}$
M3	$MV_1 \rightarrow SC_{Federation}$: [N2, $SignatureRes$, $Signature_{mv}$]$_{\text{HTTPS}}$
M4	$SC_{Federation} \rightarrow MV_1$: [N1, $registerMetaverseRes$] $_{\text{HTTPS}}$

Table 6.3: Protocol Flow for registration of a Metaverse to Federation Registry

- **Step 1:** A metaverse (MV_1) becomes the first initiator of the registration procedure by issuing a message $M1$ containing a nonce (N_1), the registration request ($registerMetaverseReq$), and the credentials ($mvCredentials$) of MV_1 that are securely passed using HTTPS to the federation registry($SC_{Federation}$).
- **Step 2:** In response to message $M1$, the federation registry ($SC_{Federation}$) will reply with $M2$, second nonce ($N2$) and signature request ($SignatureReq$) of the metaverse encrypted with HTTPS.
- **Step 3:** The metaverse (MV_1) responds with a message $M3$, which has the earlier-received nonce ($N2$), the signature response ($SignatureRes$) and the digital signature ($Signature_{mv}$) and this is sent back to the federation registry ($SC_{Federation}$) via HTTPS.
- **Step 4:** At last, the federation registry ($SC_{Federation}$) returns message $M4$ to the Metaverse (MV_1) consisting of the initial nonce ($N1$) and the success data of the registration ($registerMetaverseRes$) which completes the secure registration of the Metaverse to the federation registry.

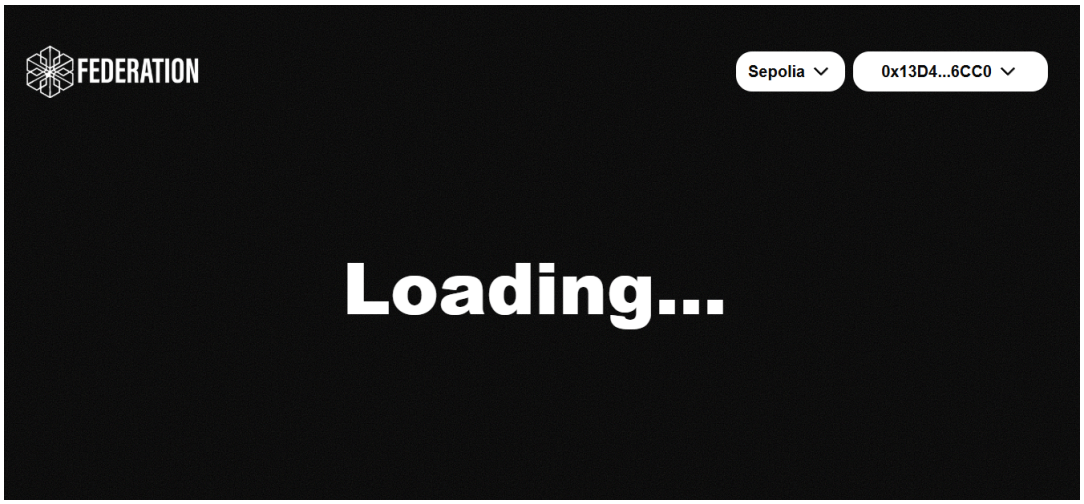

Figure 6.1: Logging into Federation Panel

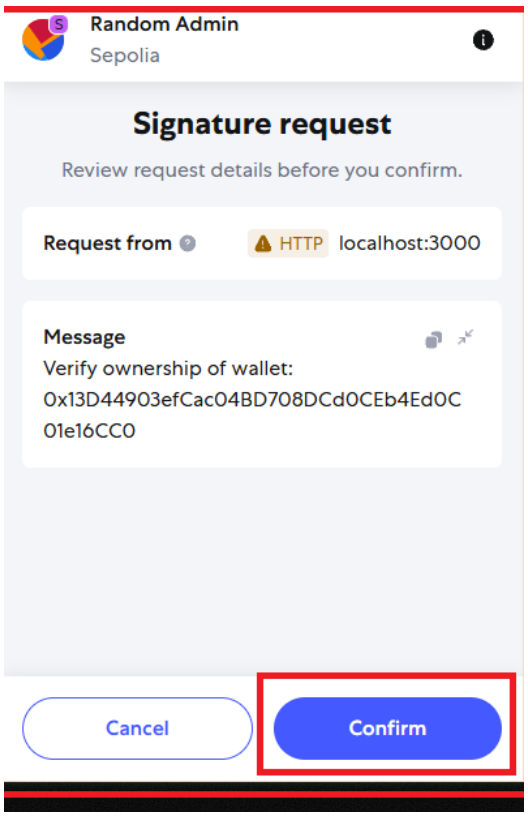

Figure 6.2: Providing Authentication Signature through Metamask

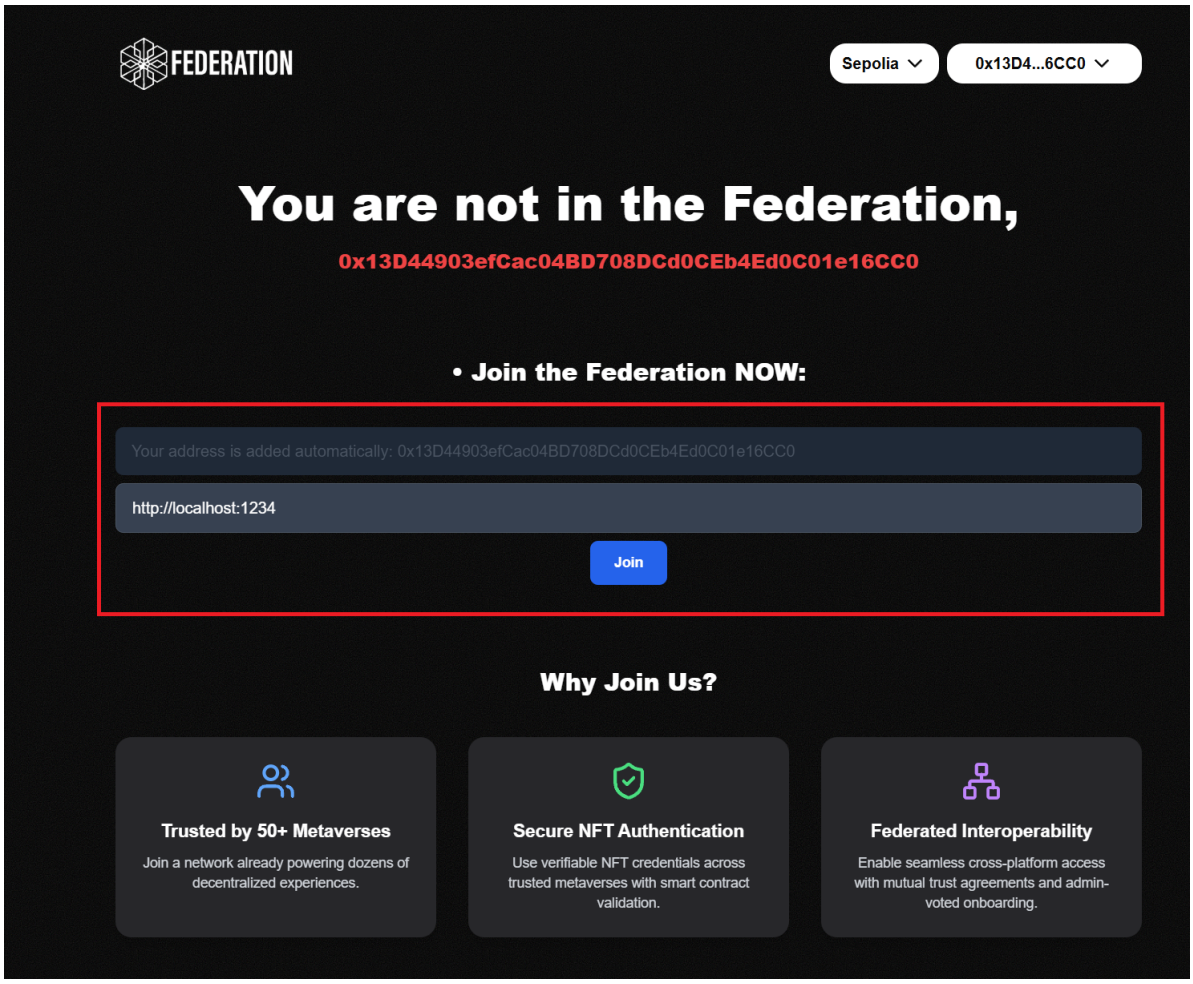

Figure 6.3: Federation Joining Request Page

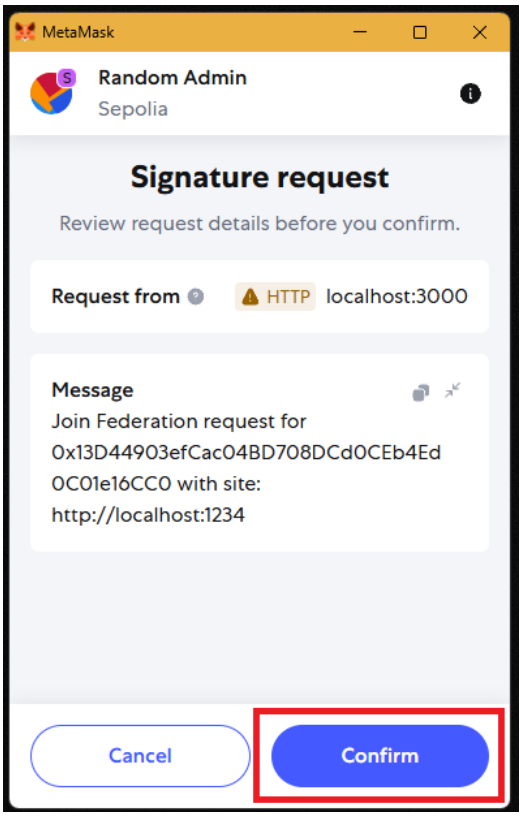

Figure 6.4: Providing Signature for Joining Request

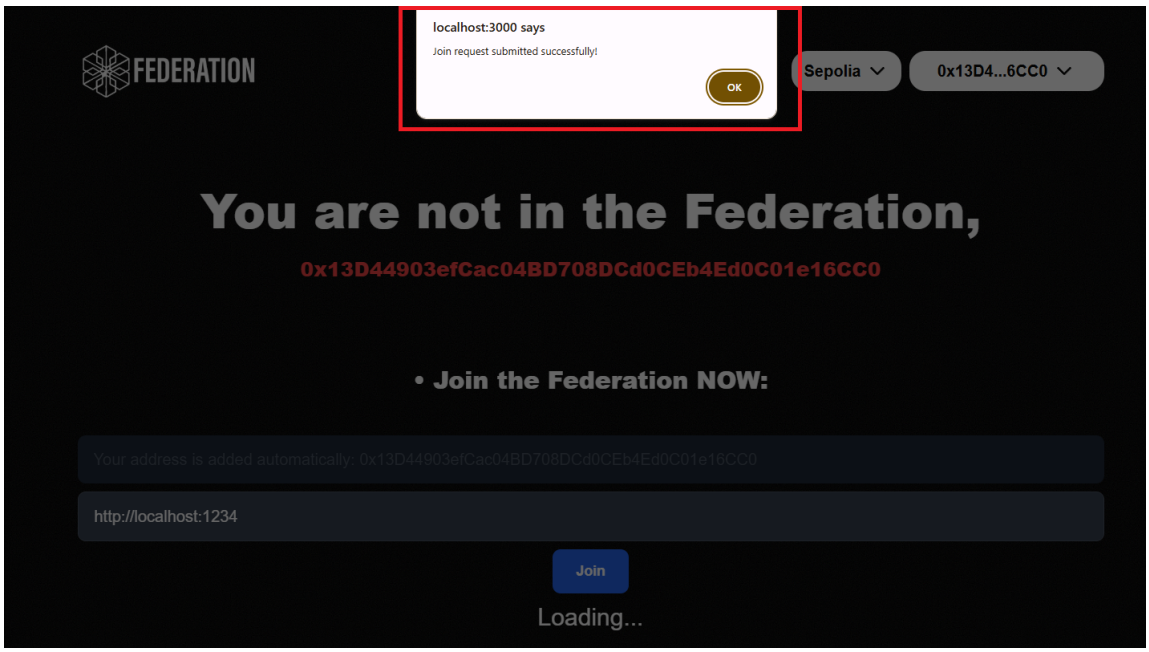

Figure 6.5: Request Submission Success Message

Here, a Metaverse Admin will enter the federation shows in (figure 6.1). the, It will conncet the wallet address to the federation registry Application shows in (figure

6.2). Then, It will make a join request in the federation in (figure 6.3). After providing the signature for the joining request like (in figure 6.4), It will be successfully sent to the federated metaverses like (figure 6.5).

6.4 NFT-Metaverse Federation Membership Protocol Flow and Use Cases

To act as a member of the federation, metaverses must have to build a trust among them. Therefore, we will to add these records in a registry by following the protocol presented in (Table 6.4). The flow is outlined below:

Steps	Description
M1	$MV_1 \rightarrow SC_{Federation}$: [N1, $getURLReq(URL_{mv_2})$, $WalletAdress_{MV2}$]$_{\text{HTTPS}}$
M2	$SC_{Federation} \rightarrow MV_1$: [N1, $getURLRes(URL_{MV_2})$, URL_{MV_2}]$_{\text{HTTPS}}$
M3	$MV_1 \rightarrow MV_2$: [N2, $addTrusteeReq$, $WalletAddress_{mv_1}$]$_{\text{HTTPS}}$
M4	$MV_2 \rightarrow MV_1$: [N3, $SignatureReq$, $Challange$]$_{\text{HTTPS}}$
M5	$MV_1 \rightarrow MV_2$: [N3, $SignatureRes$, $Signature_{mv_1}$,URL_{MV_1}]$_{\text{HTTPS}}$
M6	$MV_2 \rightarrow MV_1$: [N2, $addTrusteeRes$]$_{\text{HTTPS}}$

Table 6.4: Protocol Flow for Adding a Metaverse Membership to Federation Registry

- **Step 1:** A Metaverse ($MV1$) asks the federation registry ($SC_{Federation}$) by $M1$ message with nonce ($N1$), a request of the URL of another Metaverse ($getURLReq(URL_{mv_2})$), and the Wallet address of another Metaverse MV_2 ($WaAddress_{MV_2}$) through HTTPS.
- **Step 2:** The federation registry ($SC_{Federation}$) responds with message $M2$ to the Metaverse 1 (MV_1) with same nonce ($N1$), response of the URL request ($getURLReq(URL_{mv_2})$) and the actual URL of Metaverse 2 (URL_{MV_2}) via HTTPS.
- **Step 3:** By using the information received, Metaverse 1 (MV_1) sends $M3$ directly to Metaverse 2 (MV_2), comprising a new nonce ($N2$), a membership trust adding request ($addTrusteeReq$) and the wallet address of Metaverse 1 ($WalletAddress_{MV_1}$) through HTTPS.
- **Step 4:** Then, Metaverse 2 (MV_2) uses that message by responding with $M4$ containing a new nonce ($N3$), signature request ($SignatureReq$), and a challenge value ($Challenge$) that will be sent securely over HTTPS.
- **Step 5:** Now, Metaverse 1 (MV_1) will respond with $M5$ containing received nonce ($N3$), signature response ($SignatureRes$), digital signature ($Signature$) and Metaverse 1 URL (URL_{MV_1}) via HTTPs.
- **Step 6:** At the last, Metaverse 2 (MV_2) finishes trust registration and sends $M6$ to Metaverse 1 (MV_1), nonce ($N2$) and the response of the membership adding process of trustees ($addTrusteeRes$) in HTTPS protocol, realizing trust establishment.

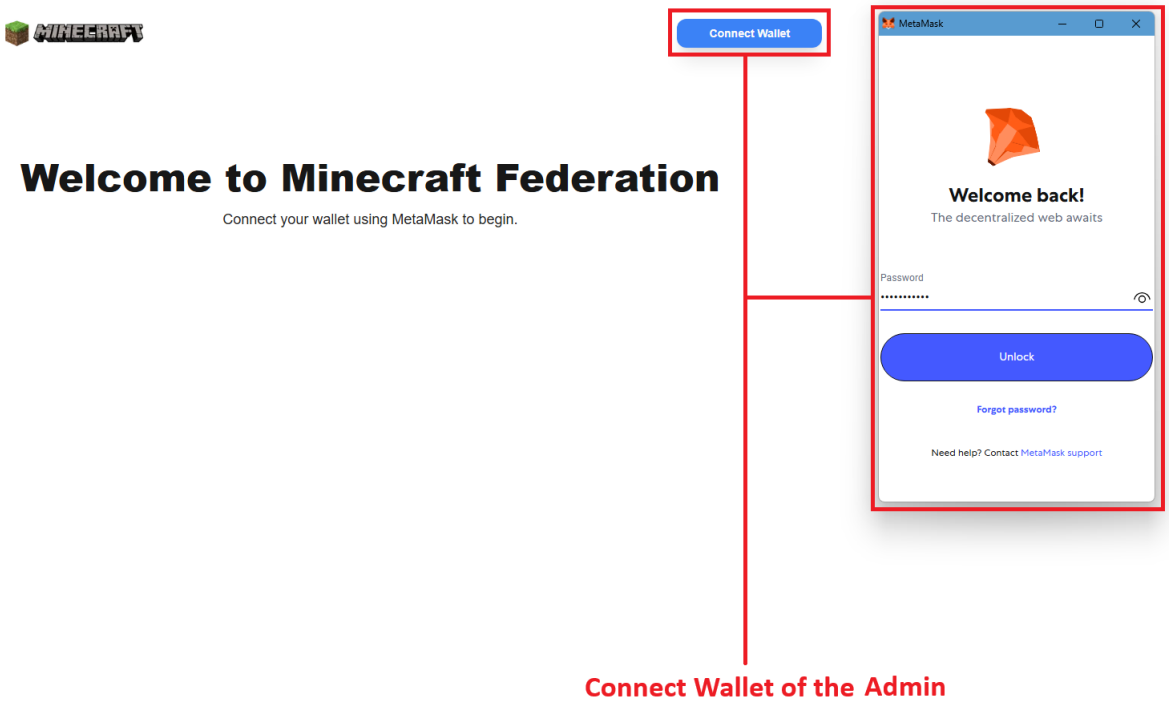

Figure 6.6: Connecting Metamask Wallet

Figure 6.7: Verifying as Admin

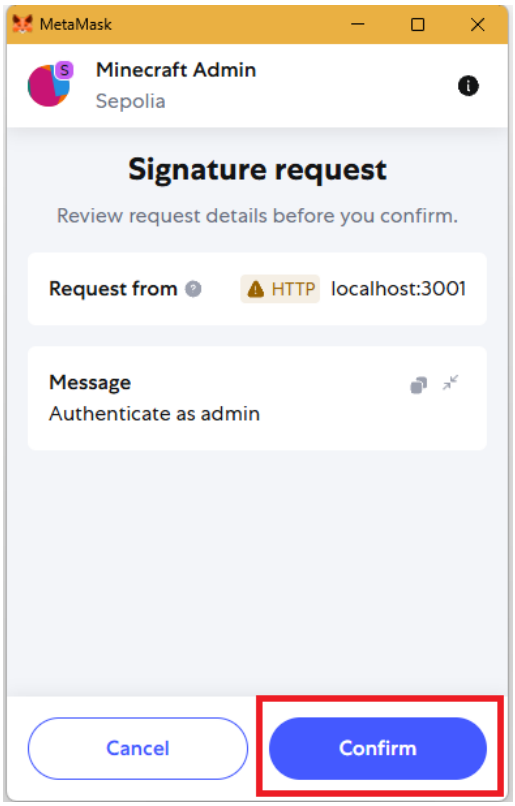

Figure 6.8: Providing Authentication Signature through Metamask

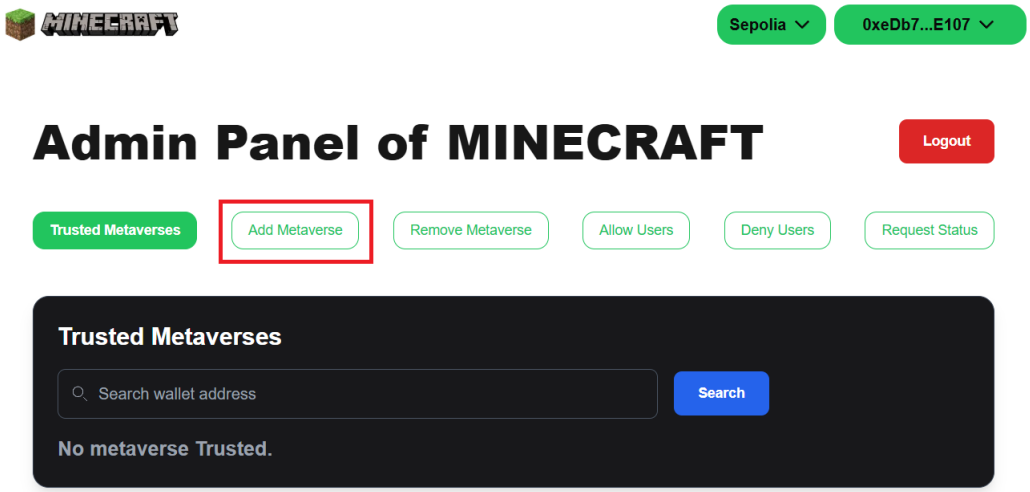

Figure 6.9: Initial Trust Registry Screen

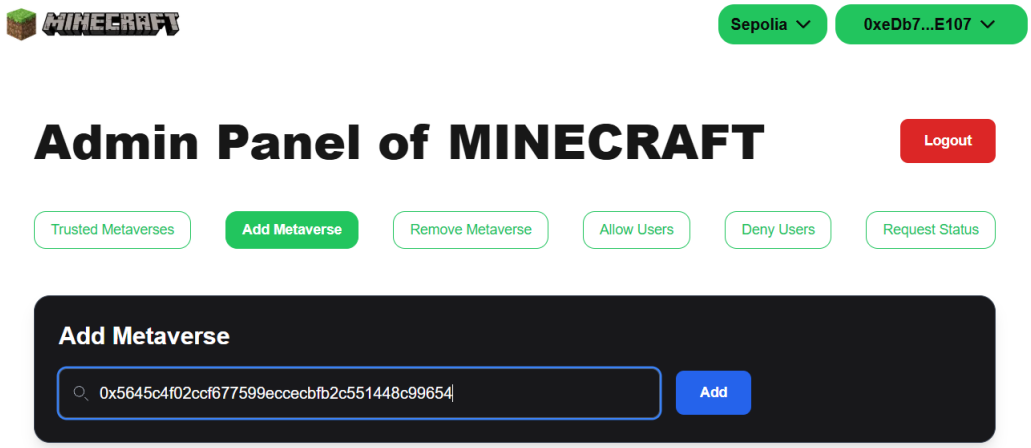

Figure 6.10: Requesting to Add a Metaverse to Trust Registry

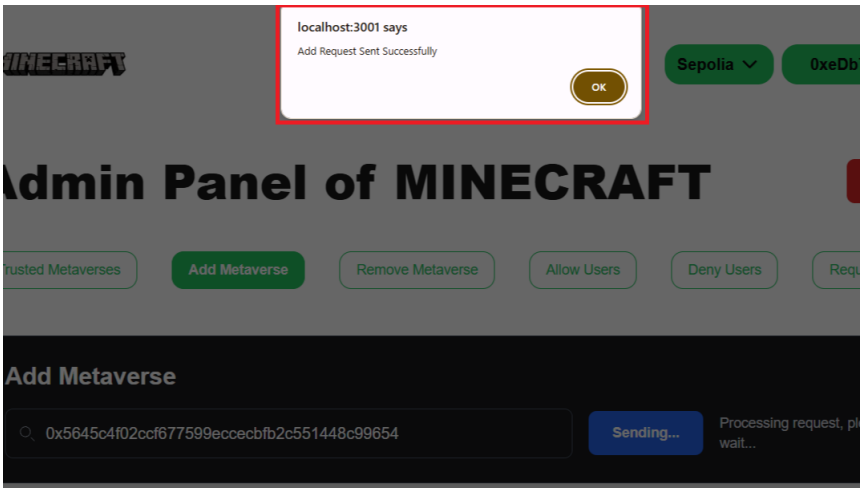

Figure 6.11: Confirmation of Request

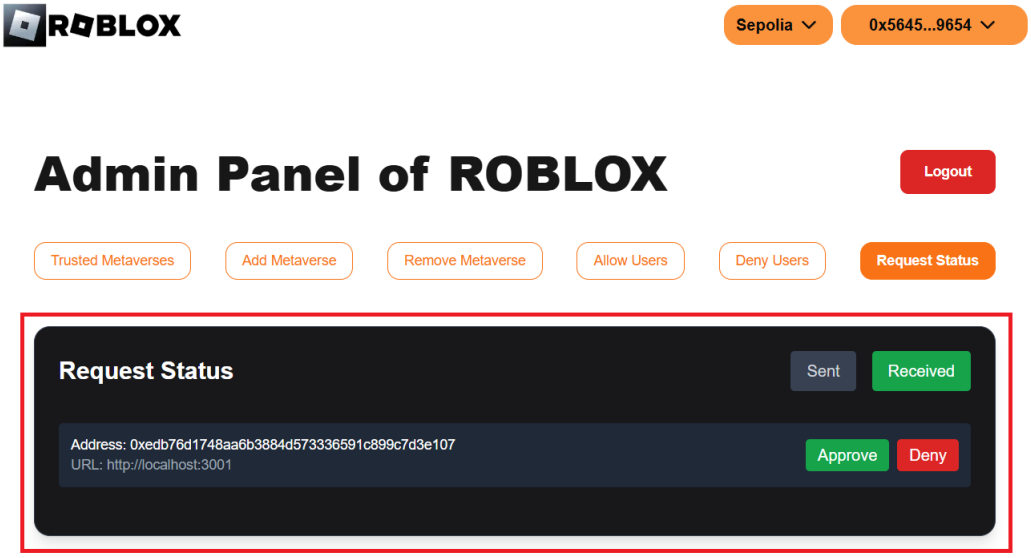

Figure 6.12: Approving Request from Trusted Metaverse

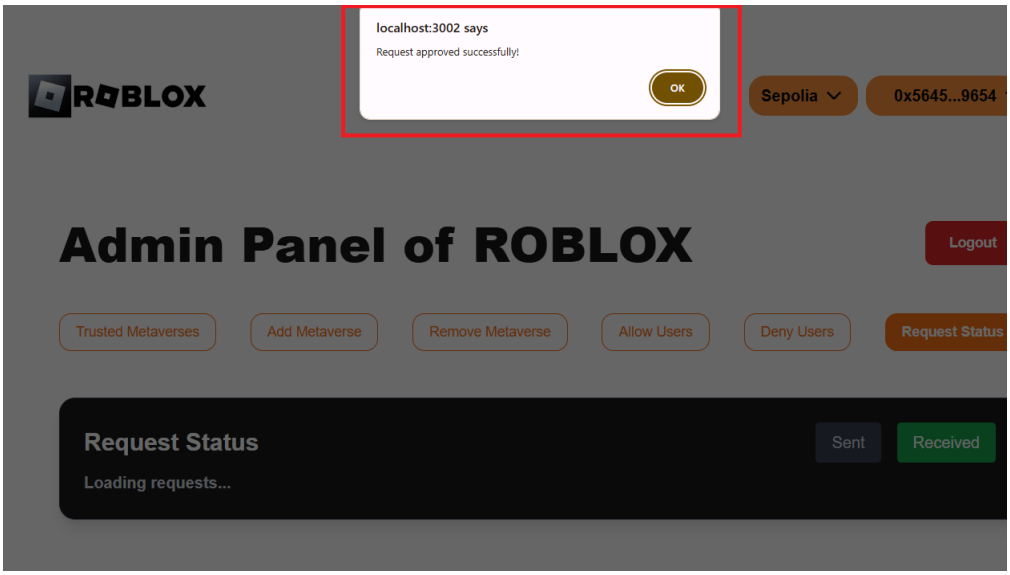

Figure 6.13: Confirmation of Approval

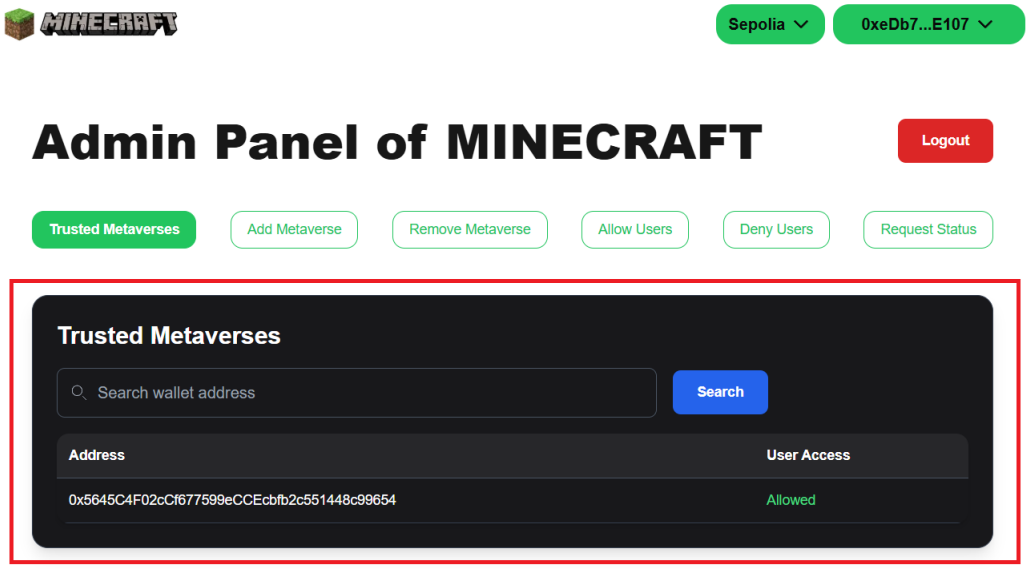

Figure 6.14: Updated Trust Registry Screen

Here,The metaverse enter to its admin page shows in (figure 6.6). Then, The Metamask wallet will be connected to verify the Metaverse admin in (Figure 6.7). After that it will provide an authentication signature (figure 6.8). The successfull attempt will be redirect to the Metaverse admin ($Minecraft$) application in (Figure 6.9). In the Add Metaverse tab the admin will provide the wallet address of the another Metaverse ($Roblox$) admin to request for adding the metaverse into the trust registry like (figure 6.10) and It will show the confirmation of the request shows in (figure 6.11). Then, another Metaverse ($Roblox$) admin will get the request in his admin panel shown in (Figure 6.12). After Approving the request by $Roblox$ metaverse admin shows in (Figure 6.13), its walllet address will be shown in the Truested Metaverse tab of $Minecraft$ Metaverse admin like (figure 6.14).
Similarly, $Minecraft$ Metaverse Admin's Wallet Address will also be shown in another metaverse ($Roblox$) admin's application.

6.5 NFT-Metaverse Federation Membership Acceptance Protocol Flow and Use Cases

To Allow the User of Trustee Metaverse of the federation to get access, the trustor metaverse need to Add the User Allowence (Table 6.5). The flow is outlined below:

Steps	Description
M1	$MV_1 \rightarrow SC_{Registry}$: [N1, $addUserAllowenceReq$, $WalletAddress_{MV_2}$]$_{\text{HTTPS}}$
M2	$SC_{Registry} \rightarrow MV_1$: [N1, $addUserAllowenceRes$]$_{\text{HTTPS}}$

Table 6.5: Protocol Flow for Allowing Adding a Metaverse Membership to Federation Registry

- **Step 1:** The trustor metaverse (MV_1) initiates the first message $M1$, with nonce ($N1$), the allowence request ($addUserAllowenceReq$), and a reference to the trustee ($WalletAddress_{MV_2}$) encrypted with HTTPS sent to the registry table($SC_{Registry}$).
- **Step 2:** Then, The registry table ($SC_{Registry}$) sends the last message $M2$ to the trustor (MV_1), which includes the initial nonce ($N1$) and the User Allowence response ($addUserAllowenceRes$). This ensures that the User of trustee Metaverse (MV_2) can get access to the Trustor Metaverse (MV_1).

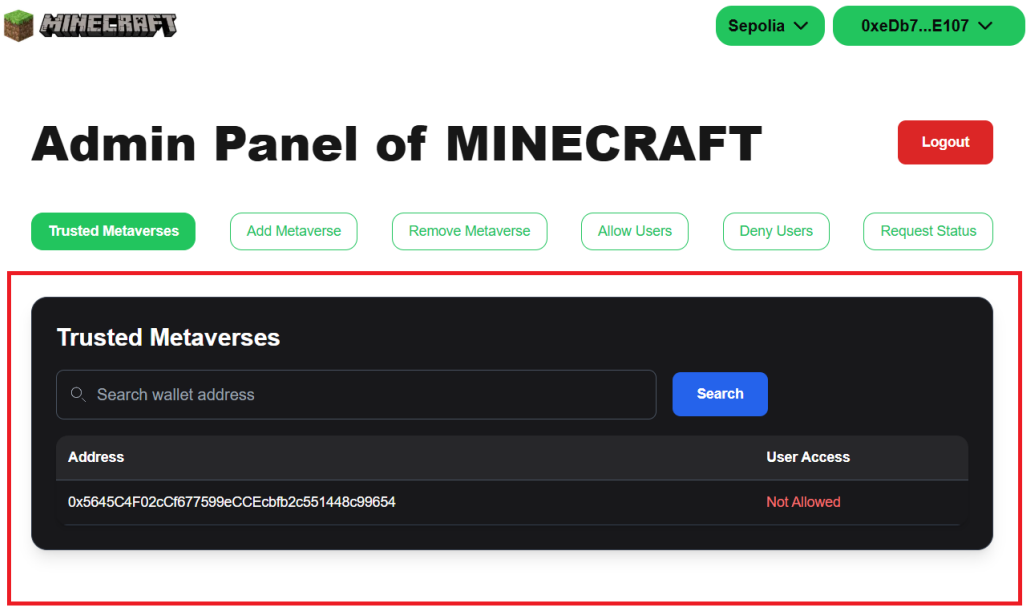

Figure 6.15: Initial Trust Registry Screen

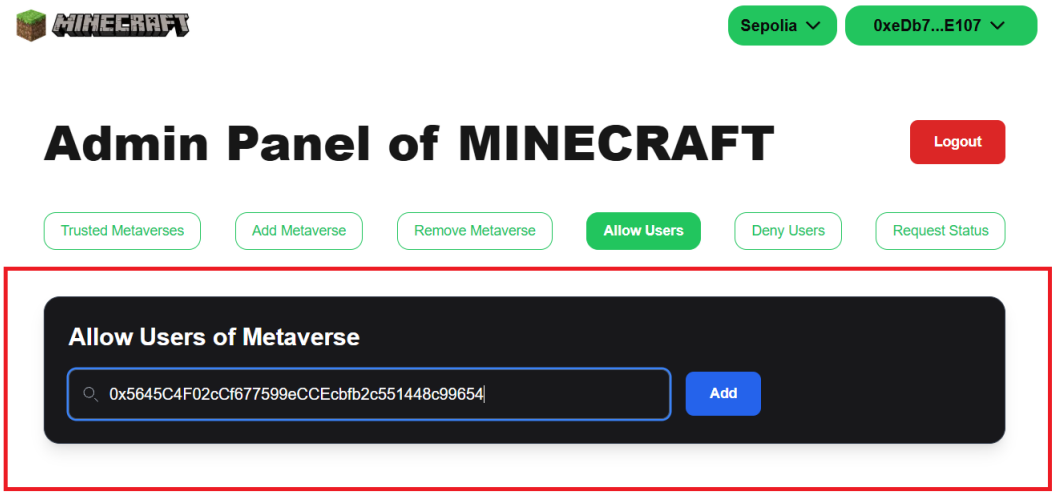

Figure 6.16: Allowing Users from Trusted Metaverse

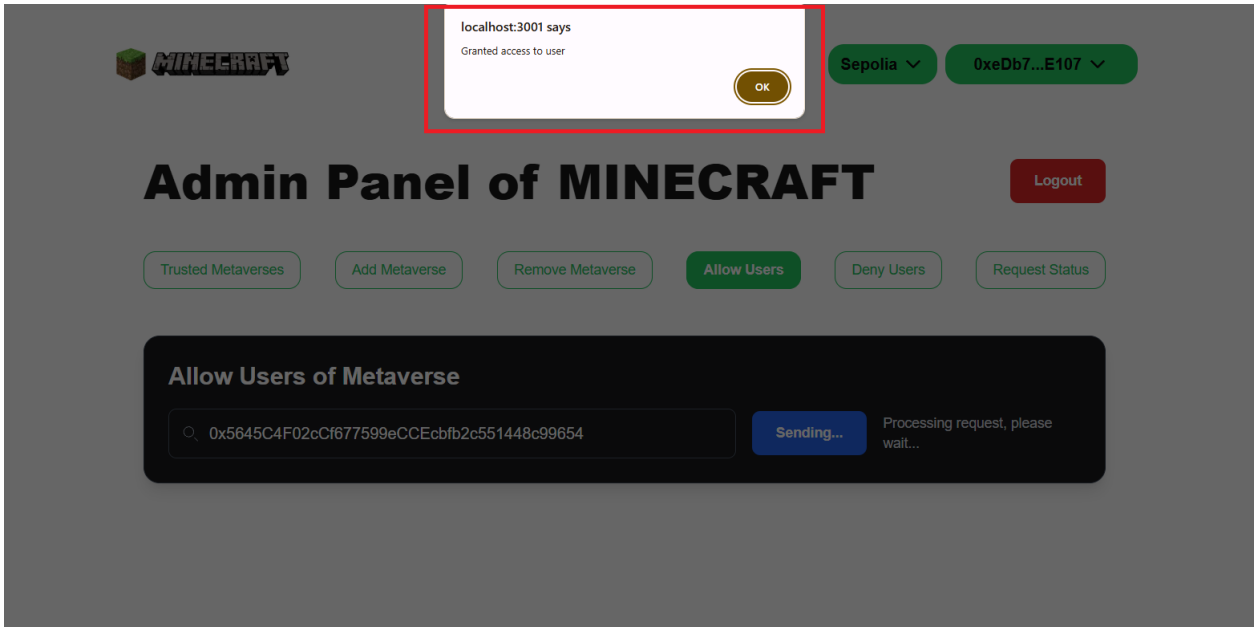

Figure 6.17: Confirmation Message of Allowance

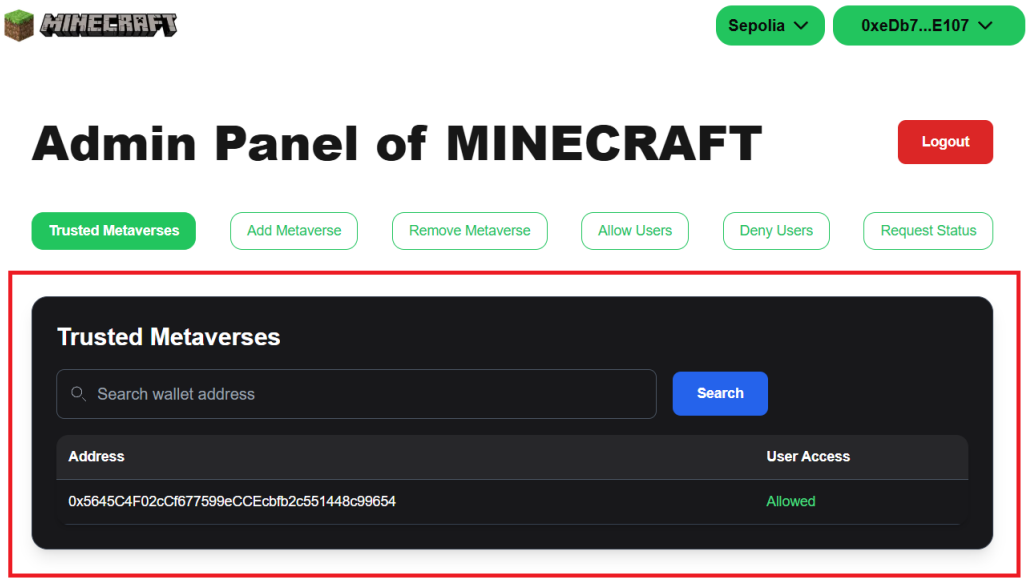

Figure 6.18: Updated Trust Registry Screen

A metaverse Admin can allow Users access to the trusted Metaverse user's after denying it for several reason's. If the User allowence is not given by the $MInecraft$ Metaverse to another Metaverse which is shown in (figure 6.15), The Metaverse admin can allow the Users of that trusted metaverse again after providing the wallet address in the Allow Users tab in (Figure 6.16). Finally, After getting the comfirmation message shows in (figure 6.17), The Metaverse $Minecraft$ will get allow all the users of its trusted Metaverse in (figure 6.18).

6.6 NFT-Metaverse Federated Identity Registration Protocol Flow and Use cases

To access metaverses, a user must register and receive an NFT following the protocol presented in (Table 6.6). The flow is outlined below:

Steps	Description
M1	$U \rightarrow MV_1$: [N1, $registerReq$, $credentials$]$_{\text{HTTPS}}$
M2	$MV_1 \rightarrow U$: [N1, $registerRes$, $SignedCredentials^{U}_{MV_1}$]$_{\text{HTTPS}}$
M3	$U \rightarrow SC_{Token}$: [N2, $NFTMintReq$, $credentials$, $SignedCredentials^{U}_{MV_1}$]$_{\text{HTTPS}}$
M4	$SC_{Token} \rightarrow U$: [N2, $NFTMintRes$, $NFT^{U}_{MV_1}$]$_{\text{HTTPS}}$

Table 6.6: NFT-Metaverse Federated Identity Registration Protocol

- **Step 1:** The user (U) starts the registration process with the first message $M1$, nonce ($N1$), the ($registerReq$) and the user information ($credentials$) encrypted with HTTPS sent to the metaverse (MV_1).

- **Step 2:** Upon receiving message $M1$, the metaverse (MV_1) sends back a response of registration ($registerRes$) and signed credentials of the user ($SignedCredentials^{U}_{MV_1}$) with nonce ($N1$) as per message $M2$.
- **Step 3:** Then, The user (U) responds by sending $M3$ message, which includes nonce ($N2$), NFT minting request ($NFTMintReq$) with the signed credentials of the user ($SignedCredentials^{U}_{MV_1}$) and the user's credentials ($credentials$) to the Blockchain (SC_{Token}). These credentials are sent over HTTPS to ensure confidentiality.
- **Step 4:** Finally, the Blockchain (SC_{Token}) sends the last message $M4$ to the user (U), which includes the initial nonce ($N1$), NFT Minting response ($NFTMintRes$), and the minted token ($NFT^{U}_{MV_1}$). This ensures that the user receives confirmation of their registration and an NFT credential to interact with the system.

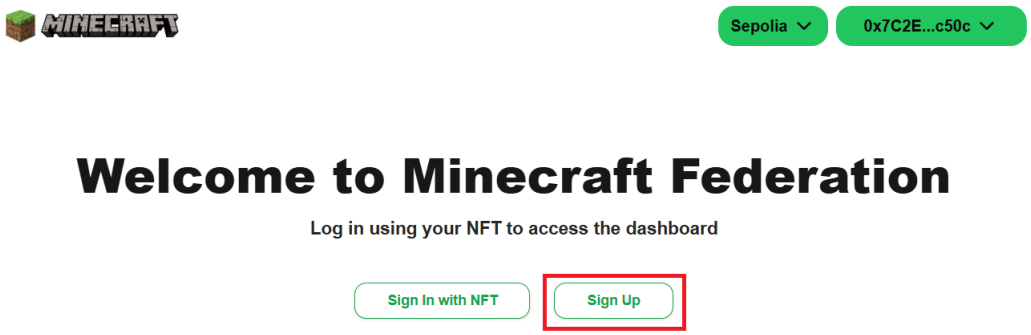

Figure 6.19: Authentication Page

At first The Authentication Page will be shown to the User by the Minecraft Metaverse in (Figure 6.19).

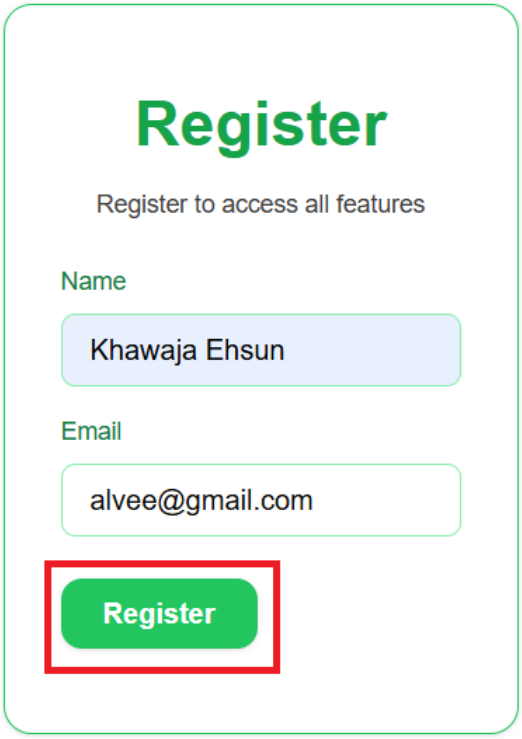

Figure 6.20: Register Form

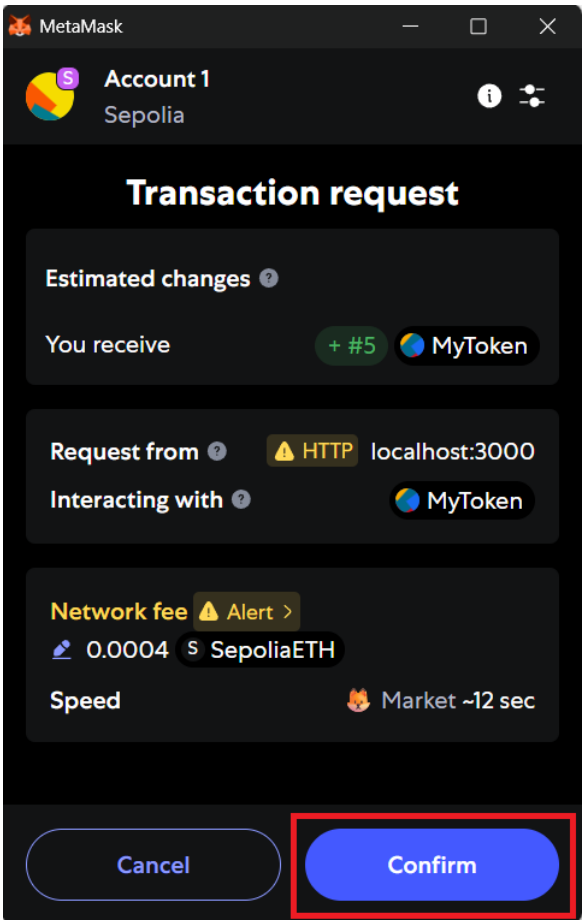

Figure 6.21: Confirming Smart Contract Transaction on Metamask

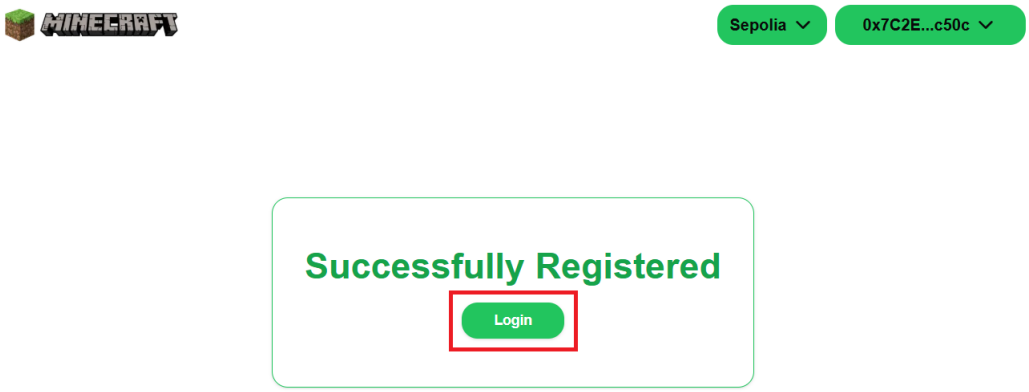

Figure 6.22: Register Success Screen

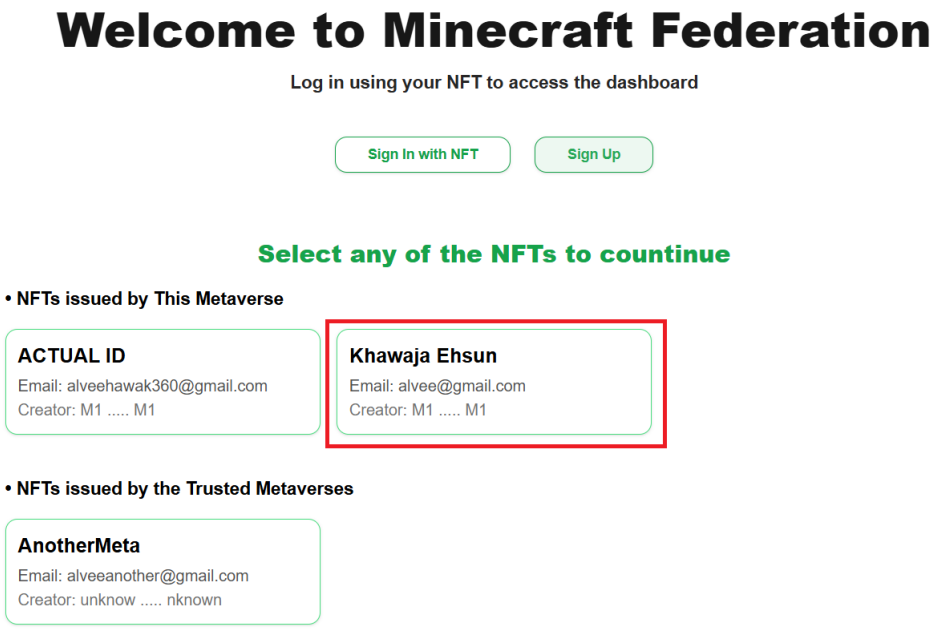

Figure 6.23: Freshly Minted NFT Showing up for Login

when the User clicked the Sign Up button, it will redirect to the Registration form in (figure 6.20). Then, after confirming the transaction via metamask (show in figure 6.21), it will show a successfulling registerd page (in figure 6.22). A new minted NFT will be shown to the user wallet in the Sign in with NFT page in (figure 6.23).

6.7 NFT-Metaverse Federated Identity Authentication Protocol Flow and Use Cases

In this protocol (Table 6.7), we authenticate a user to access a Metaverse service leveraging an NFT in a federated authentication manner. The flow is presented below:

Steps	Description
M1	$U \rightarrow MV_1$: [N1, $serviceReq$]$_{\text{HTTPS}}$
M2	$MV_1 \rightarrow U$: [N2, $WalletAddressReq$]$_{\text{HTTPS}}$
M3	$U \rightarrow MV_1$: [N2, $WalletAdressRes$, $WalletAddress_U$]$_{\text{HTTPS}}$
M4	$MV_1 \rightarrow SC_{Token}$: [N3,$NFTReq$, $WalletAddress_U$]$_{\text{HTTPS}}$
M5	$SC_{Token} \rightarrow MV_1$: [N3, $NFTRes$, $NFTs^U$]$_{\text{HTTPS}}$
M6	$MV_1 \rightarrow SC_{Registry}$: [N4, $registryCheckReq$, $NFTIssuerPublicAddresses$]$_{\text{HTTPS}}$
M7	$SC_{Registry} \rightarrow MV_1$: [N4, $registryCheckRes$]$_{\text{HTTPS}}$
M8	$MV_1 \rightarrow U$: [N5, $NFTs^U_{TrusteeMV}$]$_{\text{HTTPS}}$
M9	$U \rightarrow MV_1$: [N5, NFT^U_{mv}]$_{\text{HTTPS}}$
M10	$MV_1 \rightarrow U$: [N6, $SignatureReq$]$_{\text{HTTPS}}$
M11	$U \rightarrow MV_1$: [N6, $SignatureRes$, $Signature_U$]$_{\text{HTTPS}}$
M12	$MV_1 \rightarrow SC_{Token}$: [N7, $NFTMetadataReq$]$_{\text{HTTPS}}$
M13	$SC_{Token} \rightarrow MV_1$: [N7, $NFTMetadataRes$]$_{\text{HTTPS}}$
M14	$MV_1 \rightarrow SC_{Registry}$: [N8, $isUserAllowedReq$]$_{\text{HTTPS}}$
M15	$SC_{Registry} \rightarrow MV_1$: [N8, $isUserAllowedRes$]$_{\text{HTTPS}}$
M16	$MV_1 \rightarrow U$: [N1, $serviceRes$]$_{\text{HTTPS}}$

Table 6.7: NFT-Metaverse Federated Identity Authentication Protocol

- **Step 1:** In the first stage, Authentication process is initiated by a user (U) sends a service request ($serviceReq$) and a nonce ($N1$) to a metaverse (MV_1) through HTTPS.
- **Step 2:** As a response, a message $M2$ is sent by the metaverse (MV_2) which includes a nonce ($N2$) and a Wallet Address Request ($WalletAddressReq$).
- **Step 3:** The user (U) responds with its own wallet address ($WalletAddress_U$) together with the response ($WalletAddressRes$) and with the same nonce (N2) to the Metaverse (MV_1) over HTTPS.
- **Step 4:** Then, the Metaverse (MV_1) sends a request to get NFTs ($NFTReq$) of that specific Wallet by passing its address($WalletAddress$) to the blockchain smart contract (SC_{Token}) in a secure channel through message $M4$.
- **Step 5:** After that, the blockchain Smart contract (SC_{Token}) verifies it, processes the verification, and responds with message $M5$ containing the NFT response ($NFTRes$) with a list of NFTs of the User($NFTs^U$) back to the Metaverse (MV_1).
- **Step 6:** In message $M6$, the Metaverse (MV_1) sends a registry check request ($registryCheckReq$) to the registry table ($SC_{Registry}$) via HTTPS, which carries a new nonce ($N4$) and a list of NFT isssuers public addresses ($NFTIssuerPublicAddresses$) to check whether NFT issuers are trusted or not.
- **Step 7:** After that, the request is processed by the registry ($SC_{Registry}$) and it sends back a registry check response ($registryCheckRes$) with an array of *booleans* that show which NFTs have a trusted source (i.e. were not issued by rogue federated members) in message $M7$ to the Metaverse (MV_1).

- **Step 8:** Upon verifying the NFT information, the Metaverse (MV_1) shows the trusted NFT data ($NFTs^{U}_{trusteeMV}$) and nonce ($N5$) to the user (U) using HTTPS.
- **Step 9:** The user(U) will respond with the chosen NFT (NFT^{U}_{mv}) that they would like to use during the authentication process, referring to a nonce ($N5$) and sending it via HTTPS.
- **Step 10:** Now, the Metaverse (MV_1) sends a challenge- response step by delivering Signature Request ($SignatureReq$) and a new nonce ($N6$) to the user (U) by HTTPS.
- **Step 11:** The user (U) responds with Signature Response ($SignatureRes$) that includes the user signature ($Signature_U$) and the nonce ($N6$) over Https.
- **Step 12:** The Metaverse (MV_1) transmits an NFT metadata query ($NFTMetadataReq$) to the blockchain smart contract (SC_{Token}) using a new nonce ($N7$) to verify the content of NFT.
- **Step 13:** The blockchain smart contract (SC_{Token}) sends the metadata ($NFTMetadataRes$) of the NFT back to the Metaverse (MV_1) with the nonce ($N7$) over HTTPS.
- **Step 14:** The Metaverse (MV_1) requests the registry table ($SC_{Registry}$) via HTTPS by sending a nonce ($N8$) and ($isUserAllowedReq$) to query whether the user is permitted to use the service given.
- **Step 15:** The registry table ($SC_{Registry}$) returns the outcome of the authorization check ($isUserAllowedRes$) together with a nonce ($N8$) to the Metaverse (MV_1) via HTTPS.
- **Step 16:** Then, the Metaverse (MV_1) sends message $M16$ to the user (U). The service response ($serviceRes$) along with the initial nonce ($N1$) are both contained in the message.

First of all, User will connect his Metamask wallet as shown in (figure 6.24). After confirming the in the Metamask wallet (figure 6.25), the welcome page will be shown to the user. Then, when user to the $ViewMyNFTs$ tab, It will show all the NFTs to the user like (figure 6.26). After clicking the specific NFTs and completing signature trhought metamask (figure 6.27), the User will be Successfully Authenticate to the Minecraft game in the Metaverse shown in (figure 6.28).

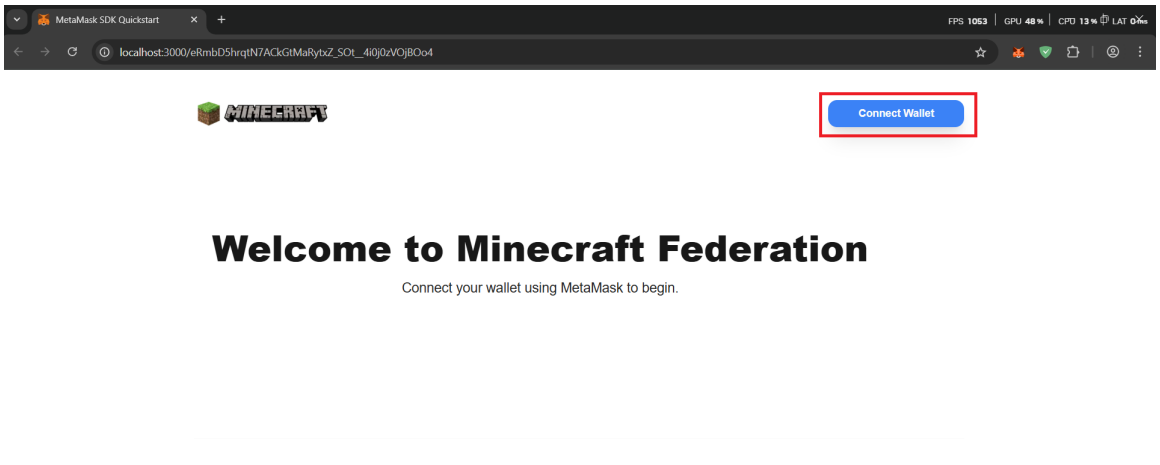

Figure 6.24: Connecting Metamask Wallet

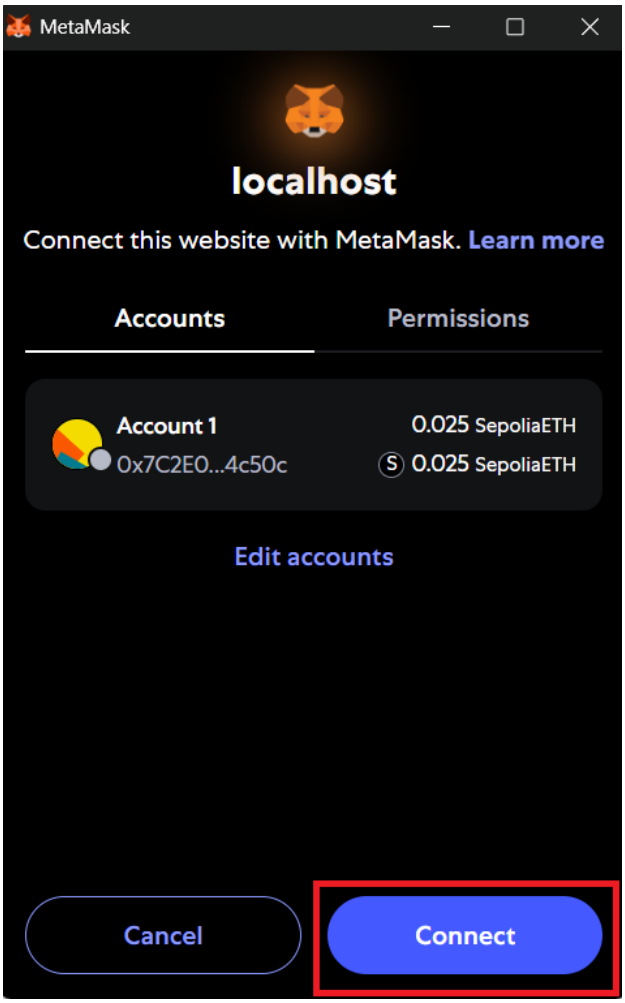

Figure 6.25: Wallet Connection Confirmation from Metamask

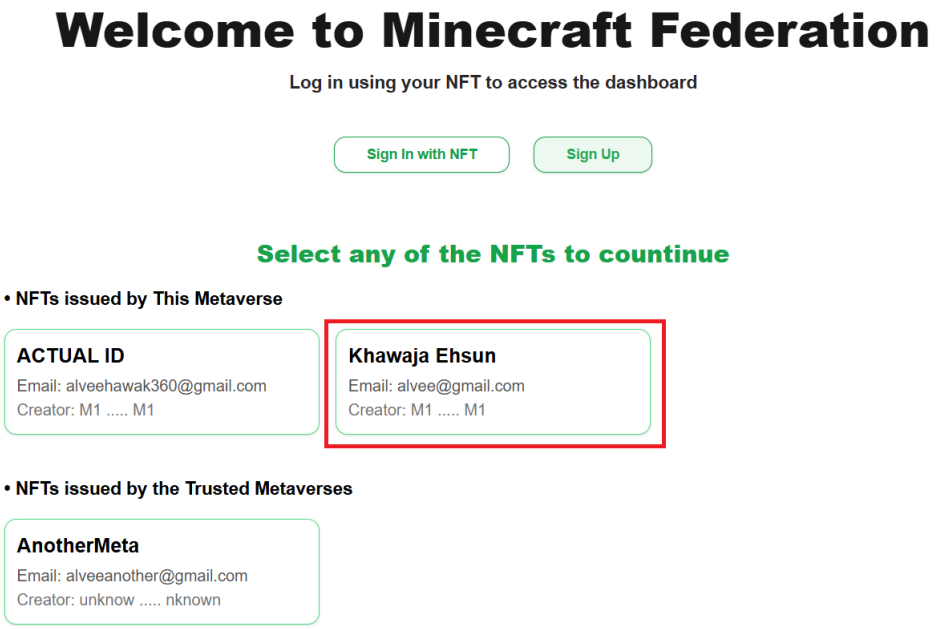

Figure 6.26: Selecting NFT for Authentication

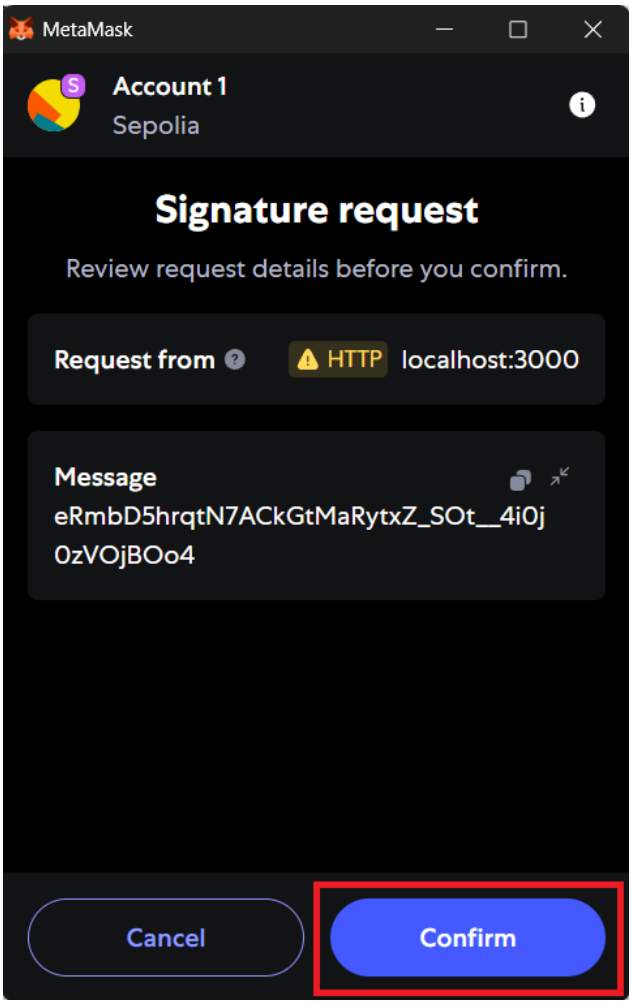

Figure 6.27: Providing Signature through Metamask

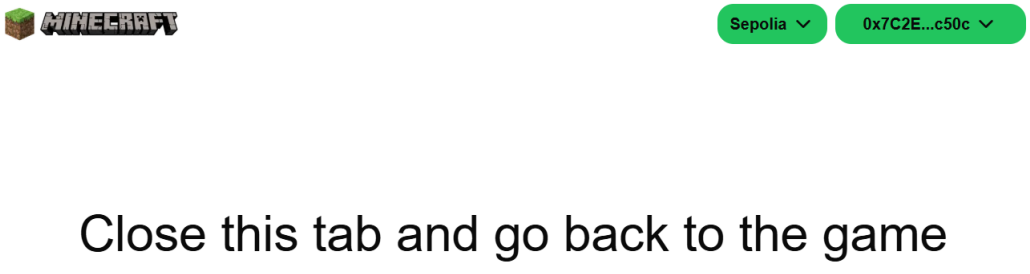

Figure 6.28: Login Confirmation Screen

Chapter 7

Development

This proposed system is established based on the design of secure, decentralized identity management and cross-metaverse platform authentication using NFTs. It has been decided to use a modular, scalable structure together with blockchain infrastructure, web technologies and integrated gaming environments into a seamless and federated whole..

The backbone of the platform is **Next.js v15.1.0**, which is the primary full-stack platform to develop both the interface with the customer and the backend. It also supports server-side rendering (SSR), and static site generators (SSG), providing better performance and scalability of high-traffic decentralized apps. NFT minting, identity verification, signature security, metadata lookup and the most important services are provided through Back end APIs which are constructed using **Node.js v22.14.0**.

To enable Web3 communication and blockchain connections, the system leverages **wagmi v2.14.8** and **web3.js v4.11.0** which enable smooth smart contract call, wallet connections, and broadcasting of transactions on the frontend. The smart contracts are created in **Solidity v0.8** with the reference to **ERC-721**, that have an ability to issue Non-Fungible Tokens to be used as identity tokens and can be validated and cannot be tampered with.

Every NFT identity token is cryptographically linked to a verified user metadata and published with the **Elliptic Curve Digital Signature Algorithm (ECDSA)**. In order to allow strong verification of signatures within federated settings, the system uses the **SignatureChecker** utility, a utility that validates ECDSA signatures in the blockchain such that only authorized parties can issue or verify identity credentials.

To achieve realistic communication with digital environments, the site utilizes **Minecraft Paper v1.21**, a modified game server build, which has the comparison of plugins, and supports federated gameplay factors. In this context, user identity verification is performed by their identity NFT enabling the smooth involvement in federated metaverses. The hooking mechanism of Smart contract and external adapters will allow secure communication between NFT verification infrastructure and Minecraft instances.

The communication between the processes and orchestration of transactions are handled with **SperkJava v2.9.3**, a proprietary glue layer between Java-based game environments and Web3 components. This guarantees low latency interactions and

safe facilitating of blockchain identity to in-game events or permissions.

Throughout the system, **ERC-721 token is not only a representation of identity but also a credential to access different zones and resources in the metaverse**. Metadata Digital signatures of a recognized entity in the federated trust network are included in a unique payload which is verified by a registration authority when creating each identity NFT.

This end-to-end deployment means that an identity infrastructure that is interoperable, verifiable, and tamper-resistant will exist in the metaverse. It enables safe boarding of users and metaverse instances, enacting digital trust with ECDSA signatures, and enables a decentralized authentication across a variety of virtual worlds.

Chapter 8

Discussion

8.1 Functional Requirement Analysis

The suggested system effectively covers all functional needs (F1-F4) and proves practical feasibility to use it to introduce NFT-based authentication in a federated metaverse infrastructure.

To fulfill F1, the system employs NFTs to the ERC-721 standard to provide user authentication. Such NFTs are serve as unique, verifiable credentials that are associated with the identities of individual users. These NFTs are authenticated by ownership verification through blockchain wallet (e.g., MetaMask) and one is able to perform a privacy-preserving authentication without referring to centralized identity stores or credentials.

For F2, the system is implemented in a public blockchain, which works as a middleware level to settle authentication transactions and facilitate identity verification in a decentralized manner. This eliminates the risks on centralized control and helps to support trustless interoperability. Every authentication is recorded in a distributed log that cannot be tampered with in the blockchain, guaranteeing the data integrity as well as transparency.

To meet F3, the system includes a federated architecture and has an admin panel which allows it to manage several multiple instances of the metaverse. The panel will offer the functionalities to add new metaverse networks to the federation, define access requirements and track the confirmation action. The system interoperability was also proved by testing with several simulated metaverses that showed that the NFT-based authentication mechanism can be applied across systems successfully. Furthermore, a federation type of voting system is applied in which all existing metaverses are required to reach a 60% Quorum Voting system to include a new metaverse platform into the federation, ensuring the decentralization as well as the trust within the federation.

For F4, the authentication system has been incorporated into standard blockchain wallets so that people can log in by using their NFTs that are stored in the wallet, like the MetaMask. The ownership is checked with cryptographic signatures, which allows getting the access securely and conveniently without using usernames or passwords. The utilization of the ERC-721 convention guarantees compatibility with extensively carried wallet infrastructures.

8.2 Security Requirement Analysis

Current system partially fulfills the security requirements (S1-S4). Although in S1 and S2 all the common features are implemented, in S3 and S4, some aspects still are not covered completely and propose ways of improvement in the future.

For S1, through the deployment of ERC-721 NFTs as authentication tokens, it is possible to use this blockchain technology to verify users before letting them access metaverse services. The signature is used to authenticate it with a cryptographic wallet, which can not be spoofed without the corresponding cryptographic key. This goes a long way in countering T1 (Spoofing) and T6 (Elevation of Privilege) as the strict access control is ensured and the danger of unwanted privilege elevation is eliminated.

For S2, The implementation public blockchain gives immutability and auditability of authentication transactions. Every authentication activity is recorded transparently, and allows complete traceability under which T2 (Tampering) and T3 (Repudiation) are alleviated. System accountability is enhanced by using blockchain consensus mechanisms, as it means that no entity can control transaction records.

S3 is partially addressed. Only valid NFT users get to authenticate but there is currently no definitive process to avoid or throttle badly-formed or unauthenticated requests early in the handling pipeline. Consequently, the system will remain susceptible against T5 (Denial of Service) when the attacker tries to overwhelm the authentication service with bad or excessive traffic. Such attacks should be countered in future by using request filtering, rate-limiting and validation on the level of smart contracts to improve their prevention.

S4 is also partially addressed. Although blockchain guarantees the authentication and ownership of the NFT credentials, it also makes all metadata of tokens to be publicly exposed because of the transparency of the work of public blockchains. This brings possible T4 (Information Disclosure) issues to the fore, particularly when the information of a sensitive nature metadata is encoded into the NFT. The system does not yet support such practices as encryption and storing sensitive data off-chain. In the future additional features could be implemented like zero-knowledge proofs(ZKPs), privacy-preserving metadata, or even encrypted IPFS links to completely satisfy this requirement.

8.3 Cost Analysis

When it comes to smart contracts, gas cost is the massive deal since every operation should pay the amount of the computational power that they utilise. Consider all the interactions that people have with smart contracts, minting an authentication NFT, voting to add a new metaverse to the federation, add a metaverse to the trust registry, etc. All of those operations lead to a gas charge, and the overall amount of those charges will define the scalability and the extent to which a dapp would be adopted.

We examined the number of units of gas consumed when you mint an NFT. We did every transaction through the Sepolia testnet in a secure environment, ensuring

that each mint had a random username and email.

If we take a look at Table 8.1 below which shows the gas cost incurred during the minting process.

Trial No	Gas Used (units)	Gas Price (GWEI)	Total Cost (ETH)
1	251385	1.633477894	0.00041063
2	268545	1.983769691	0.00053273
3	251409	1.811745275	0.00045549

Table 8.1: Gas Cost Comparison of NFT Minting Method

Moreover, to better illustrate the cost differences, Figure 8.1 presents a bar chart comparing the gas usage across the three methods.

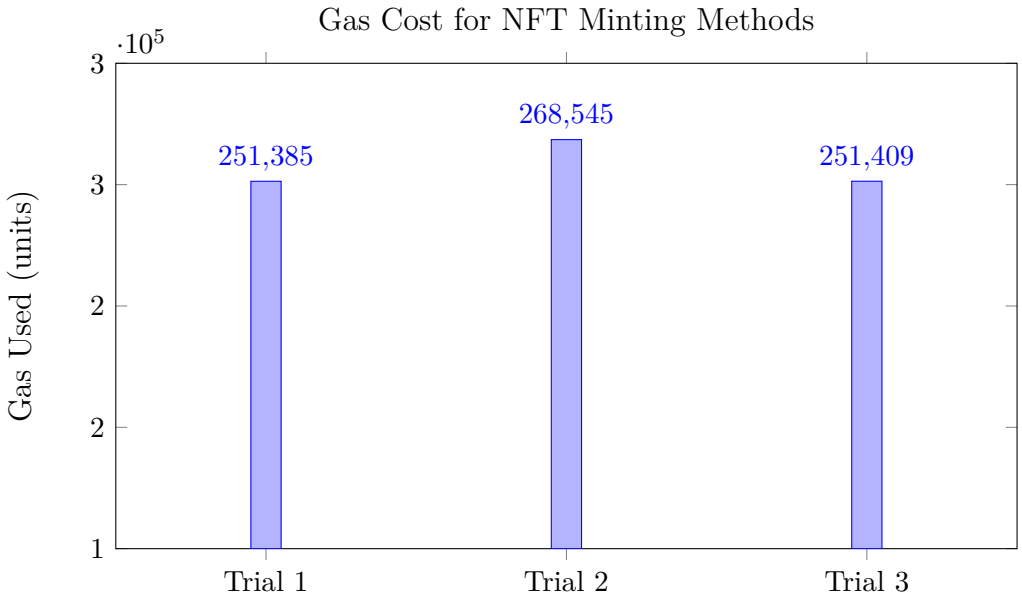

Figure 8.1: Bar chart comparing gas cost for different NFT minting methods

In conclusion, the above information and the visualization demonstrates that each of the mint transactions consumes approximately the same amount of gas, and the differences between them are practically zero. The cost of gas does not creep upwards throughout the trials, and therefore we can assume that the presented authentication mechanism is expected to perform well on large federative systems.

8.4 Comparative Analysis

This comparison table (see Table 8.2) gives an overview of recent studies relevant to the convergence of **NFTs**, **authentication** and the **metaverse**, and their functionality, security, implementation, and threat modeling analysis.

- **Identity & Authentication:** Most of the previous studies use NFTs, biometric scenarios, or Decentralized Identification (DIDs) to verify identity as used by Banaeian et al. [34], Kim et al. [36], and Hong et al. [30]. By comparison, Our work is a unique application of NFTs, which can provide secure, verifiable user authentication across an ecosystem of federated systems that are cryptographically bound to verified metadata and digital signatures.

- **Requirement & Security Analysis:** While some articles present security analysis or requirements analysis separately (e.g., Manzoor et al. [37], Kim et al. [36]) our framework combines the two views. It deals with requirement analysis including identity uniqueness and federation eligibility coupled with strong secret paired with digital signature verification as well as tamper-resistance measures.

- **Metaverse Federation Focus:** Past work has mentioned economic [35] or protocol-level [36], [37] aspects of the metaverse, but no existing protocol has suggested the presence of a specific protocol to achieve federated onboarding. The proposed *Federated Metaverse Registration Protocol* addresses this shortcoming, and allows easy and secure registration of new metaverse instances into a federation.

- **Implementation & Threat Modeling:** While other works such as that by Azzaoui et al. [40] are more theoretical in nature, our work has practical implementation (i.e., secure NFT minting algorithm with digital signature validation). Further, threat modeling is carried out explicitly, and this provides a larger and deployable solution.

In summary, the comparative analysis (Table 8.2) shows that our proposed framework covers a significant gap, as it is based on NFT identity with the possibilities of verifiability, federation readiness, and implementation feasibility, which are little discussed in current research.

Paper's Author	Key Points	Metaverse	NFT	Authentication	Requirement Analysis	Security Analysis	Implementation	Threat Modeling
Banaeian et al. [34]	Identity management, NFT challenges	Yes	ERC-721, ERC-1155	Identity Verification	Uniqueness, Privacy, Authentication, Decentralized management	Key management, Malicious insider Privacy, Decentralization	No	Yes
Manzoor et al. [37]	PoS consensus, NFTs for user authentication	Yes	ERC-721	Enhanced security and scalability	Integrity, scalability, Authentication and Resource Management	Security feature of PoS and NFT for data integrity and authentication	Yes	Yes
Hong et al. [30]	EOG data for user authentication	No	Ethereum standard	Biometric authentication	Secure and distributed Storage of biometric data and user authentication	Integrity and durablity test for NFT forgery detection and recovery	Yes	Yes
Tsegaye et al. [38]	Product authenticity, decentralized network	No	ERC-721	Product verification	combat counterfeiting in various industries (eg. luxury goods and electronics)	tamper-resistance, immutability, data manipulation and counterfeiting	Yes	No
Kim et al. [36]	DID for identity verification, SBT for unique assets	No	DID and SBT	Avatar authentication	Security identity, prevent identity theft and enhance privacy	Ensures privacy with Zero-Knowledge Proof (ZKP) and non-transferable SBTs to verify identity	Yes	No
Fauzi et al. [22]	SSO integration using NFTs	No	Not Mentioned	Web SSO authentication	System Functionality, Security Enhancement interoperability of OpenID Connect and OAuth 2.0	Improvements in security especially IDP-based threats	Yes	Yes
Guidi et al. [35]	Economic impact, NFT-driven markets	Yes	ERC-721	No	Economic and social impact of NFTs in the metaverse specifically in virtual land ownership and speculative market effects	centralization of platforms like The Sandbox leading to potential risks, impacting parcel value.	Yes	No
Azzaoui et al. [40]	Post-quantum security, NFT-based authentication	Yes	ERC-721, Post-Quantum NFTs	User authentication in Metaverse	securing NFTs in the Metaverse, copyright protection and secure transactions	quantum-resistant mechanisms and copyright protection measures for NFTs	No	No
Our Work	NFT-based identity, Federated metaverse registration protocol	Yes	ERC-721	NFT-based user authentication	Identity uniqueness, federation participation, user verification	Digital signature verification, NFT binding to identity, threat resilience	Yes	Yes

Table 8.2: Comparative Analysis table with Existing Literature

8.5 Advantages

The newly proposed NFT-based authentication system has a number of strong benefits that can help improve security, privacy of the user, flexibility and interoperability in a federated metaverse environment. First of all, the system utilizes the ERC-721 standard, which ensures that every NFT used as a tool of authentication is unique, verifiable, and tamper-proof. Users can own and control their digital identity in a safe environment using popular crypto wallets (e.g., MetaMask). Thus, the users are not being dependent on centralized identity service providers.

Among the most important advantages is the possibility of users to prove their authenticity on different systems of the metaverse in the federation when only one NFT credential is used. This lessens the need of numerous registrations and supports an easy access across the platform user experience. The federation model with the support of the admin panel allows having multiple metaverse instances involved in a common authentication environment and retains the decentralized control. Such an organization increases flexibility and scalability and does not entail centralization vices.

Additionally, the authentication processes carried out on-chain are completely transparent and auditable. Since all the authentication transactions are logged on the public blockchain, there is no possibility of altering access records and creating fake access records. This assists in creating trust amongst metaverses that interact and discourage malicious actors.

The other benefit is the autonomy and privacy of the user. Data theft and identity theft is minimized as a user will be in control of his or her NFT and no sensitive data will be stored on a centralized server. The system also provides the uniqueness of session with the help of cryptographic confirmation based on the wallet, which is really effective to avoid replaying attacks and illegal requests of access. Overall, the proposed system increases security, decentralization, interoperability, and trust of users which are fundamental attributes of the secure and scalable network of the federated metaverse.

8.6 Challenges and Limitations

In spite of its advanced architecture, the system has a set of technical and practical challenges. One of them is the public exposure of metadata related to NFTs. Despite NFT-generated invulnerable and traceable immutability and traceability, they have their own operation basis, which occurs on a public blockchain, thus potentially revealing non-sensitive but linkable metadata. The challenge to have privacy and make the best use of blockchain technology transparency is something that continues to face certain challenges.

Another restriction is in how unverified or malicious requests are handled. The implementation that is currently present optimises the use of NFTs by denying access to non-valid NFTs, but does not largely optimise the fore-time rejection of invalid or malformed authentication attempts in the early phase. It may result in loss of efficiency or even points of weakness when the amount of requests are large.

Additionally, cross-platform interoperability which could be proved in internal sim-

ulations, in real life implementation, could be restricted by differences in API structures, rendering engines, and platform policies. It would need more effort to open up standard yet universally applicable integration standard that would need to be worked on to set some common protocols and compatibility rules.

The other restriction is caused by the fact that the existing incompatibility between VR headsets, such as Meta Quest, and authentication systems via a wallet, such as MetaMask. As MetaMask usually works as browser extensions or mobile apps, users need to change their VR setup to a desktop where the authentication can be performed. it not only spoil the immersive experience, but it also creates friction in usability, as those users who do not have the easy access to a physical keyboard or an external device will have problems using them in VR. This limitations restricts ease of access, and user convenience within full immersive environments.

The usability also brings in the aspect of relying on user wallet interactions. Not every user might know what wallet-based authentication is and users who are not as technically savvy might run obstacles on adopting such a solution. When it comes to the optimal level of security and ease of use, it has to be balanced, and it requires constant improvements.

8.7 Future Work

To reinforce the capacities and enlarge the usability of the suggested system multiple future work directions are imagined. Among these priorities is the need to come up with uniform APIs and SDKs that can be used to achieve easy integration of the authentication mechanism on more metaverse platforms. This will facilitate ease of onboarding and improve standardization of communication regulations between systems and speed up adoption and interoperability.

A promising additional improvement will be the incorporation of decentralized file storage, e.g. InterPlanetary File System (IPFS) to store assets (asset-related metadata, or NFT-related metadata) or governance metadata off-chain. By combining IPFS with blockchain, the system will be able to allow enhanced scalability and decreased storage overhead on chain, whilst still keeping the disruptive aspect of immutability and verifiability of essential data through content addressing. It can be particularly handy in decreasing transactional costs and improving privacy because sensitive or bulky data can simply be stored off-chain, yet just cryptography hashes are kept on-chain.

It is also the work of the future to consider privacy protection mechanisms, e.g., zero-knowledge proofs (ZKPs), encrypted metadata or selective disclosure approaches to in order to guarantee anonymity of the user as well as data security in the case of public blockchains. It will be useful in counterbalancing the fundamental intransigence of blockchain and the confidentiality concerns of digital identity.

Further on, introducing automation of the request filtering and intelligent traffic management, which will increase the efficiency and resilience of the authentication process under the changed condition of the load. This involve initial filtering of the invalid, unauthorized request at once to save the unnecessary load to the system and minimize the possible attack plane.

Finally, by expending the administrative capabilities of the system which include dynamic trust scoring, role-based access-control, and adaptive policy-enforcement-the ability to handle smaller grained federation management, and governance would be facilitated. This will enable metaverse networks to specify and develop access models on a per-time basis depending upon the level of trust, or behavior or requirement of compliance to it. With these improvements, the system would be transformed into a privacy-sensitive, highly scalable and versatile solution of secure identity management and federated administration in decentralized virtual world.

Chapter 9

Conclusion

The Metaverse introduces a new age of digital interaction via immersive augmented and virtual realities, but identity and access across overlapping networks becomes a potentially life-altering issue. Conventional, password-driven and centralized authentication systems lack the ability to mitigate the privacy, security and interoperability requirements of decentralized Metaverse systems.

The current thesis suggests a secure and NFT-built authentication system in a federated Metaverse framework, in which a system of various platforms comprises a Circle of Trust (CoT) and executes decentralized and trustworthy identity checks. The system takes advantage of the immutable and transparent nature of blockchain and utilizes smart contracts and ERC-721 NFTs by making them as tamper-proof digital credentials, cryptographically signed and verifiable in federated spaces.

The protocols and architecture have been developed and implemented partially with Solidity, Web3.js, node.js and ECDSA signatures which show the practicality of decentralized identity management. In comparative analysis, our method has significantly improved decentralization, control of users, and ability to resist spoofing of identities compared to approaches that are traditional.

Future research must look into zero-knowledge proofs in the area of privacy, embrace decentralized identity principles such as DIDs and VCs, the shortcomings of small-scale adoption of the current system, create governance frameworks of federated trust and integrate adaptive security and multifactor authentication.

In Summary, this thesis forms the foundation of the decentralized authentication infrastructure based on NFTs, specific to the demands of federated Metaverses. It deals with the shortcomings of the traditional systems, and creates possibilities of safe, scalable and user driven identity management within the immersive digital world.

Bibliography

[1] A. Hevner, A. R, S. March, S. T, Park, J. Park, Ram, and Sudha, "Design science in information systems research," *Management Information Systems Quarterly*, vol. 28, pp. 75–, Mar. 2004.

[2] E. Castronova, *Synthetic Worlds: The Business and Culture of Online Games*. University of Chicago Press, 2005.

[3] OASIS, *Assertions and protocols for the oasis security assertion markup language (saml) v2.0*, http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf, Mar. 2005.

[4] T. Boellstorff, *Coming of Age in Second Life: An Anthropologist Explores the Virtually Human*. Princeton University Press, 2008.

[5] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[6] M. Wasko, R. Teigland, D. Leidner, and S. Jarvenpaa, "Stepping into the internet: New ventures in virtual worlds," *MIS Quarterly*, vol. 35, pp. 645–652, Sep. 2011. DOI: 10.2307/23042801.

[7] Y. Assia, *Colored coins bitcoinx*, https://bitcoinx.org/, 2012.

[8] V. Buterin, *Ethereum white paper*, Ethereum, 2013. [Online]. Available: https://ethereum.org/en/whitepaper/.

[9] J. D. N. Dionisio, W. G. Burns, and R. Gilbert, "3d virtual worlds and the metaverse: Current status and future possibilities," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, pp. 1–38, 2013.

[10] J. D. N. Dionisio, W. G. Burns III, and R. Gilbert, "3d virtual worlds and the metaverse: Current status and future possibilities," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, pp. 1–38, 2013.

[11] A. Shostack, *Threat Modeling: Designing for Security*. Wiley, 2014.

[12] M. Swan, *Blockchain: Blueprint for a New Economy*. O'Reilly Media, 2015.

[13] A. Zohar, "Bitcoin: Under the hood," *Communications of the ACM*, vol. 58, no. 9, pp. 104–113, 2015.

[14] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, and R. Wattenhofer, "On scaling decentralized blockchains," in *International Conference on Financial Cryptography and Data Security*, 2016, pp. 106–125.

[15] Y. Guo and C. Liang, "Blockchain application and outlook in the banking industry," *Financial Innovation*, vol. 2, no. 1, p. 24, 2016.

[16] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies*. Princeton University Press, 2016.

[17] J. Witte, "The blockchain: A gentle four page introduction," *ArXiv*, vol. abs/1612.06244, 2016.

[18] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (sok)," in *Proceedings of the 6th International Conference on Principles of Security and Trust*, 2017, pp. 164–186.

[19] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," Jun. 2017. DOI: 10.1109/BigDataCongress.2017.85.

[20] J. N. Bailenson, *Experience on Demand: What Virtual Reality Is, How It Works, and What It Can Do*. W. W. Norton Company, 2018.

[21] B. Cao, Z. Zhang, D. Feng, S. Zhang, L. Zhang, M. Peng, and Y. Li, "Performance analysis and comparison of pow, pos and dag based blockchains," *Digital Communications and Networks*, vol. 6, no. 4, pp. 480–485, 2020, ISSN: 2352-8648. DOI: https://doi.org/10.1016/j.dcan.2019.12.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864819301476.

[22] E. Fauzi, S. Yuliani, Y. Syukriyah, and A. Zakiah, "Model of nft implementation on web sso over openid connect and oauth 2.0 protocols," *International Journal of Engineering, Science and Technology*, vol. 6, no. 2, 2021. DOI: 10.31539/intecoms.v6i2.6972.

[23] J. Ma, "Blockchain consensus mechanism based on improved distributed consistency and hash entropy," *Journal of Electrical and Computer Engineering*, vol. 2021, Article ID 2030810, 2021. DOI: 10.1155/2021/2030810.

[24] P. Paul, P. S. Aithal, R. Saavedra, and S. Ghosh, "Blockchain technology and its typesa short review," *International Journal of Applied Science and Engineering (IJASE)*, vol. 9, no. 2, pp. 189–200, 2021, Available at SSRN: https://ssrn.com/abstract=4050933, ISSN: 2321-0745.

[25] Q. Wang, R. Li, Q. Wang, and S. Chen, "Non-fungible token (nft): Overview, evaluation, opportunities and challenges," *Journal of Network and Computer Applications*, vol. 174, p. 102 884, 2021.

[26] W. Yao, J. Ye, R. Murimi, and G. Wang, "A survey on consortium blockchain consensus mechanisms," *arXiv preprint arXiv:2102.12058*, 2021.

[27] L. R. Christensen and A. Robinson, "The potential global economic impact of the metaverse," Analysis Group, Tech. Rep., 2022. [Online]. Available: https://www.analysisgroup.com/globalassets/insights/publishing/2022-the-potential-global-economic-impact-of-the-metaverse.pdf.

[28] M. Dowling, "Is non-fungible token pricing driven by cryptocurrencies?" *Financial Economics Letters*, vol. 44, p. 101 283, 2022.

[29] M. R. Gupta, "Nfts in authentication: Use cases and implementation," *Journal of Digital Asset Management*, vol. 6, no. 3, pp. 98–112, 2022.

[30] K. Hong and B. M. Lee, "Ethereum-based user authentication model using eog data for a service provider," *Journal of Multimedia Information System*, vol. 9, pp. 275–286, Dec. 2022. DOI: 10.33851/JMIS.2022.9.4.275.

[31] S. Mystakidis, "Metaverse," *Encyclopedia*, vol. 2, no. 1, pp. 486–497, 2022.

[32] P. Wang, "Challenges in modern authentication systems," *Journal of Network and Computer Applications*, vol. 124, pp. 34–45, 2022.

[33] L. Xie and M. Zhu, "The evolution of authentication systems: A comparative study," *IEEE Transactions on Information Forensics and Security*, vol. 18, no. 4, pp. 234–256, 2022.

[34] S. Banaeian Far and S. M. Hosseini Bamakan, "Nft-based identity management in metaverses: Challenges and opportunities," vol. 5, p. 260, Sep. 2023. DOI: 10.1007/s42452-023-05487-5.

[35] B. Guidi and A. Michienzi, "The social impact of nfts in the metaverse economy," Sep. 2023, pp. 428–436. DOI: 10.1145/3582515.3609564.

[36] G. Kim and J. Ryou, "Digital authentication system in avatar using did and sbt," *Mathematics*, vol. 11, p. 4387, Oct. 2023. DOI: 10.3390/math11204387.

[37] K. Manzoor, U. Noor, and Z. Rashid, *Nft-based blockchain-oriented security framework for metaverse applications*, Jul. 2023.

[38] B. Z. Tsegaye, D. K. Mekonnen, M. H. Sahle, N. G. Fisseha, and K. S. Arvind, "Tracing authenticity: A blockchain and nft-based approach to product verification," 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259934554.

[39] D. Zhang, "Multi-factor authentication in cloud computing environments," *International Journal of Cloud Computing*, vol. 10, no. 3, pp. 221–235, 2023.

[40] A. El Azzaoui and J. Kim, "Qnft: A post-quantum non-fungible tokens for secure metaverse environment," *Journal of Information Processing Systems*, vol. 20, no. 2, pp. 273–283, 2024. DOI: 10.3745/JIPS.03.0196.