2007
KARABÜK
ÜNİVERSİTESİ

A BLOCKCHAIN-BASED IDENTITY MANAGEMENT AND ACCESS CONTROL FRAMEWORK FOR THE METAVERSE

2023
MASTER THESIS
COMPUTER ENGINEERING

Huda Suhail SEYAM

Thesis Advisor
Assoc. Prof. Dr. Adib HABBAL

A BLOCKCHAIN-BASED IDENTITY MANAGEMENT AND ACCESS CONTROL FRAMEWORK FOR THE METAVERSE

Huda Suhail SEYAM

Thesis Advisor
Assoc. Prof. Dr. Adib HABBAL

T.C.
Karabuk University
Institute of Graduate Programs
Department of Computer Engineering
Prepared as
Master Thesis

KARABUK
June 2023

THESIS APPROVAL PAGE

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

Assoc. Prof. Dr. Adib HABBAL ....

Thesis Advisor, Department of Computer Engineering

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

Examining Committee Members (Institutions)			Signature
Chairman	: Assist. Prof. Dr. Nehad T.A. RAMAHA	(KBU)	....
Member	: Assoc. Prof. Dr. Adib HABBAL	(KBU)	....
Member	: Assist. Prof. Dr. Yusuf Yargı BAYDİLLİ	(HU)	....

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

Prof. Dr. Müslüm KUZU ....

Director of the Institute of Graduate Programs

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

Huda Suhail SEYAM

ABSTRACT

M. Sc. Thesis

A BLOCKCHAIN-BASED IDENTITY MANAGEMENT AND ACCESS CONTROL FRAMEWORK FOR THE METAVERSE

Huda Suhail SEYAM

Karabük University
Institute of Graduate Programs
The Department of Computer Engineering

Thesis Advisor:
Assoc. Prof. Dr. Adib HABBAL

June 2023, 104 pages

Metaverse, the next paradigm of the Internet, has attracted the attention of everyone, including the general public, academicians, as well as the industry. The concept of Metaverse aims to offer a shared virtual space to interact with other people or platforms through the integration of several innovative technologies, in particular, virtual and augmented reality, Artificial Intelligence, Blockchain technology, and 5G networks. In the Metaverse platform, not only user data can be collected, stored, and processed, but also every action, interaction, response, etc. will be recorded by the Metaverse platform. In other words, the user's identity and his data are totally managed with less or even minimum control of data owners. Hence, this has become a major challenge leading to serious data security and privacy issues. This research proposes an Identity Management (IdM) and Access Control (AC) framework that adopts the identity management approach for future Metaverse systems by shifting the control of digital identities from the Metaverse platforms to users to have full control over their own identity. Consequently, empowering users to control who can access what in this Metaverse environment. The framework has been implemented on Ethereum consortium Blockchain and it was evaluated based on laws of identity. It turns out that the proposed framework overcomes identity challenges and meets user privacy requirements. Moreover, The framework has been evaluated based on performance metrics, including transaction gas cost, gas limit, block period, and throughput. The experimental results show high performance compared to other IdM solutions.

Key Words: Blockchain, Identity Management, Metaverse, Access Control, Ethereum.

Science Code : 92430

ÖZET

Yüksek Lisans Tezi

METAVERSE İÇİN BLOCKCHAIN TABANLI KİMLİK YÖNETİMİ VE ERİŞİM KONTROLÜ ÇERÇEVESİ

Huda Suhail SEYAM

Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Doç. Dr. Adib HABBAL

Haziran 2023, 104 sayfa

İnternetin bir sonraki paradigması olan Metaverse, endüstrinin yanı sıra kamuoyunun, akademisyenlerin de dahil olduğu herkesin ilgisini çekmiştir. Metaverse kavramı, özellikle sanal ve artırılmış gerçeklik, Yapay Zeka, Blockchain teknolojisi ve 5G ağları gibi çeşitli yenilikçi teknolojilerin entegrasyonu yoluyla diğer insanlarla veya platformlarla etkileşime geçmek için paylaşılan bir sanal alan sunmayı amaçlamaktadır. Metaverse platformunda sadece kullanıcı verileri toplanamayacak, saklanamayacak ve işlenemeyecek, aynı zamanda her eylem, etkileşim, yanıt vb. Metaverse platformu tarafından kaydedilecektir. Diğer bir deyişle, kullanıcının kimliği ve verileri, veri sahiplerinin daha az hatta minimum kontrolü ile tamamen yönetilmektedir. Dolayısıyla bu, ciddi veri güvenliği ve mahremiyet sorunlarına yol açan büyük bir zorluk haline geldi. Bu araştırma, dijital kimliklerin kontrolünü Metaverse platformlarından kullanıcıların kendi kimlikleri üzerinde tam kontrole sahip olmalarına kaydırarak gelecekteki Metaverse sistemleri için kimlik yönetimi yaklaşımını benimseyen bir Kimlik Yönetimi (IdM) ve Erişim Kontrolü (AC) çerçevesi önermektedir. Sonuç olarak, kullanıcıları bu Metaverse ortamında kimin neye erişebileceğini kontrol etme yetkisi vermek. Çerçeve, Ethereum konsorsiyumu

Blockchain üzerinde uygulandı ve kimlik yasalarına göre değerlendirildi. Önerilen çerçevenin kimlik zorluklarını aştığı ve kullanıcı gizlilik gereksinimlerini karşıladığı ortaya çıktı. Ayrıca çerçeve, işlem gaz maliyeti, gaz limiti, bloke süresi ve verim gibi performans ölçütlerine dayalı olarak değerlendirilmiştir. Deneysel sonuçlar, diğer IdM çözümlerine kıyasla yüksek performans göstermektedir.

Anahtar Kelimeler : Blockchain, Kimlik Yönetimi, Metaverse, Erişim Kontrolü, Ethereum.

Bilim Kodu : 92430

ACKNOWLEDGMENT

After thanking and praising Allah, I extend my thanks, gratitude, and appreciation to all those who contributed to bring this thesis to light. I own my deepest gratitude to my supervisor, Assoc. Prof. Dr. Adib Habbal, for his great interest and assistance in preparation of this thesis. He always takes me out of my comfort zone to empower me as a researcher. I really appreciate his effort to provide advice, encouragement, and guidance throughout his supervision. I also do not miss to extend my thanks and appreciation to the teaching staff at Karabuk University, who provided us with science and knowledge. I also admire the help and support of my parents. They raised me to be strong and to work towards my dreams. I am indebted to my family forever. I am thankful to my husband for his patience during my studies. He was always present when it was needed. Finally, I am grateful to my homeland Palestine to make me who I am now.

CONTENTS

Page

CHAPTER 3

CHAPTER 4

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABBREVIATIONS INDEX

ABBREVIATIONS

ABI	: Application Binary Interface
API	: Application Programming Interface
CA	: Certificate Authority
DApp	: Decentralized Applications
DLT	: Distributed Ledger Technology
DSR	: Design Science Research
EVM	: Ethereum Virtual Machine
HTTP	: Hyper Text Transfer Protocol
IPC	: Inter-process communication
IdM	: Identity Management
IDMS	: Identity Management system
IPFS	: InterPlanetary File System
JSON	: JavaScript Object Notation
JSON-RPC	: JavaScript Object Notation Remote Procedure Call
JWT	: JSON Web Token
NPM	: Node Package Manager
P2P	: Peer-to-peer
PKI	: Public Key Infrastructure
POA	: Proof of Authority
POS	: Proof of Stack
POW	: Proof of Work
QR	: Quick Response
SSI	: Self-sovereignty identity
UI	: User Interface

CHAPTER 1

INTRODUCTION

1.1. OVERVIEW

The significant amount of time that people spend on the Internet keeps climbing exponentially especially, during major crises such as COVID-19 pandemic. This leads to increase the interest in the virtual world. People enter the virtual space with their digital identities to experience digital life. Digital identity refers to the combination of identifiers and credentials of entities in an appropriate context such as entity username, email address, preferences, and other entity attributes [1]. IdM is the framework and technologies used to control and manage digital identities in cyberspace [2].

The integration of the virtual and physical worlds makes digital twins, analog environments, and mixed reality a leading platform. The Metaverse market is predicted to reach $1 trillion in annual revenue [3]. However, data security and privacy have become a cause for concern. The development of Blockchain technology has made a breakthrough in achieving a Self-Sovereign Identity (SSI). At the same time, a virtual economy based on a decentralized network has enabled value attribution, distribution, and virtual identity authentication in the Metaverse.

1.2. PROBLEM STATEMENT

The digital identities of Metaverse users are still kept in traditional databases and centrally controlled by a single central authority that may have many vulnerabilities due to low security, leading hackers to exploit these vulnerabilities and causing various security breaches such as identity theft or disclosure of sensitive information [2]. This information may include addresses, telephone numbers, full names, and other sensitive information. The huge amount of personal data that we leave behind while using Metaverse services could be misused by platforms [4]. In 2016, the Cambridge Analytica scandal happened when Facebook breached the privacy of its users and leaked internal emails between the Cambridge Analytica firm and the British parliament [5]. Moreover, identity owners need to repeat registering and authenticating their identity information across different platforms in order to access their service. As result, the digital identity information will be fragmented, overshared, and unable to flow between different platforms [6]. Thus, it requires a new approach for Identity Management System (IDMS) to address these issues and meet users' privacy requirements.

1.3. RESEARCH QUESTIONS

The issue of Metaverse users having no control over their identity information is not sufficiently been solved yet [5]. Therefore, there is a need to conduct more research in this area. The overarching research question is: How to design a decentralized IDMS that enables Metaverse users to have full control over their digital identity and data sharing?

1.4. RESEARCH OBJECTIVES

Building on the success of Blockchain technology in developing decentralized solutions, it can be used to build a sustainable Metaverse system that gives users full control of their own identity and remove centralized risks. Blockchain plays a vital role in overcoming some of the issues experienced by most central repositories. Since it is a tamper-resistant ledger, ensures that the block data is trustworthy. Therefore, Blockchain helps to deliver a trust infrastructure in IdM for the Metaverse.

The main aim of this research is to design and develop a decentralized IdM and AC framework for Metaverse that leveraging of Blockchain technology and smart contracts to enable Metaverse users to have entire control over their identity information. Moreover, the research aims to assess the proposed framework based on laws of identity to reveal its strengths and weaknesses. Furthermore, the research aims to evaluate the proposed framework under different performance metrics to measure its scalability and how the framework will perform when increasing the number of transactions.

The framework has two schemes which are the IdM scheme and the AC scheme. The IdM scheme is responsible for providing users with identity services namely registration, authentication, managing attributes, identity recovery, and revoke. The AC scheme is responsible for controlling the access to users' attributes. The IdM scheme facilitates many identity services. First, it enables Metaverse users to register their identities through the mobile application after proofing their actual identity without depending on physical presence. Second, it provides recording and managing users' attributes after they are successfully authenticated. Users' attributes can be their name, age, degree, etc. Third, it facilitates users' identity recovery in case of identity loss with the ability to restore precisely stored attributes. Finally, The IdM scheme provides users' identity revoke in case of defrauding, detecting identity theft, etc. On the other hand, the AC scheme enables Metaverse users to control what of their attributes want to share and with which Metaverse platform.

1.5. RESEARCH SIGNIFICANCE

The significance that will be achieved after applying the proposed framework:

- Provide Metaverse users the ability to control what data they want to share.
- Enable secure and selectively disclose other than the complete user information. In other words, Metaverse users Metaverse users can define the necessary data that should be reveal for any transaction or interaction.
- Ensure the security and privacy of Metaverse users' sensitive data by indexing the identities attributes on Blockchain.
- Provide mapping between users' identities with their attributes.
- Enable Metaverse platforms to securely identify and authenticate their users in order to offer customized services.

1.6. RESEARCH SCOPE

The research is focused on enabling Metaverse users to have full control over their personal data and own their identity. The research proposes a framework that returns the control from Metaverse platforms back to users and presents a new level of securing Metaverse identities information by adopting the new Blockchain-based approach for building the next generation of IDMS. The framework is powered by the Consortium Ethereum Blockchain and serves the Metaverse users. Ethereum is a blockchain-based global software platform powered by blockchain technology [7]. it is used to send and receive value globally with its native cryptocurrency, known as Ether without any interference from any third party. Ethereum supports smart contract and enables developers to build and deploy DApp [8].

1.7. THESIS STRUCTURE

This thesis is organized as follows. Firstly, this part introduces the motivation behind a new IDMS approach and provides a general clarification of the proposed IdM and AC framework by specifying the problem statement and outlines the research objectives, followed by the research questions, and scope of the study.

In summary, the remaining parts of the thesis include:
Chapter 2: Literature Review provides background information and the principles that lay the foundation to understand the proposed framework. Moreover, it provides an overview of related research and applications in the areas of IdM. It presents an abstract view of similar systems; how they work and identify their features and weakness.

Chapter 3: Research Methodology covers a detailed explanation of the design science research (DSR) methodology that is being used to make the framework complete and work well.

Chapter 4: Proposed Identity Management and Access Control Framework contains the operational workflow of the proposed IdM and AC framework and the implementation details and mentions the software and hardware technologies that are utilized to build the framework. Also, it includes the analysis based on laws of identity and the performance evaluation criteria.

Finally, **Chapter 5**: Conclusion and Future Work discusses the key contributions, future directions and concludes the thesis.

CHAPTER 2

LITERATURE REVIEW

This chapter provides a background to principles, technologies, and primitives that are used throughout this thesis. Begins with Background section which provides an overview of Metaverse, Blockchain technologies, and IdM. After that, Related Work section discuss and analyze the other related work on Blockchain-based Identity Management System (IDMS) in the literature, followed by a comparison between related IDMS and the proposed framework.

2.1. BACKGROUND

In this section, the preliminaries of Metaverse and Blockchain technologies will be introduced, followed by an overview of digital identity and the evolution of IdM models.

2.1.1. Metaverse

The first version of the web is called WEB 1.0 where content of website was just read-only. Therefore, Internet users were content consumers. While in WEB 2.0, users become both content producers and consumers by facilitating content sharing. The Metaverse is considered the next generation of the web, so called WEB 3.0. Metaverse is a combination of "meta" which means transcendence and "verse" that is shorthanded from the universe, a computer-generated virtual world like the real world [3], where people represent themselves using digital avatars and they can play, work, and interact with the help of virtual and augmented reality services.

Metaverse offers unique characteristics which are immersiveness, hyper spatiotemporality, sustainability and interoperability [9]:

1. Immersiveness: Let users in the virtual space sense psychologically and emotionally immerse to create fully immersive realism through realistic images, sounds, and other sensations.

2. Hyper Spatiotemporality: Breaking the limitations of time and space exits in the real world by allowing users to seamlessly shuttle between various sub-virtual worlds with different spatiotemporal dimensions.
3. Sustainability: Metaverse should be self-sustaining by constantly getting users excited about the creation of digital content as well as open innovations. Also, it should remain persistent by being built on a decentralized architecture.
4. Interoperability: Enable users to seamlessly shuttle across sub-virtual spaces without any interruption of their digital experience. Also, it allows interchanging of digital assets between different Metaverse platforms.

2.1.2. Blockchain Technology

Blockchain technology is a distributed ledger that is widely used for recording distinct transactions. The transactions are maintained by entities on a Peer-to-Peer (P2P) network [10], [11]. Once a consensus is reached among all entities of the network, the transaction is added to a block. All blocks are bound to each other and together formed a Blockchain.

Blockchain network is a group of nodes that execute a smart contract through a consensus algorithm. The node in the Blockchain refers to a computer or client that participates in the Blockchain network. In general nodes can participate in three ways: As a full node that stores a complete copy of the distributed ledger, or as a lightweight node that stores a shallow copy of the Blockchain, or as a miner node that verifies the transactions and creates blocks [12].

2.1.2.1. Structure of Blockchain

Blockchain involves three basic concepts: block, chain, and transaction. The “block” refers to distributed data. The “chain” refers to the chronological order of blocks placed in the transaction ledger. The “transaction” is the read or write operations on the block for storing and retrieving the data. Figure 2.1 shows that the blocks are linked in a chain, so that each block holds the cryptographic hash value of the previous block [13], to give finally the criteria of de-trusted, decentralized, distributed data storage structure. The technology uses cryptographic hashes to ensure that the data of any transaction can’t be forged or tampered besides the ability to verification against integrity and security. The distributed nature is served by the distributed data across a network and P2P communication.

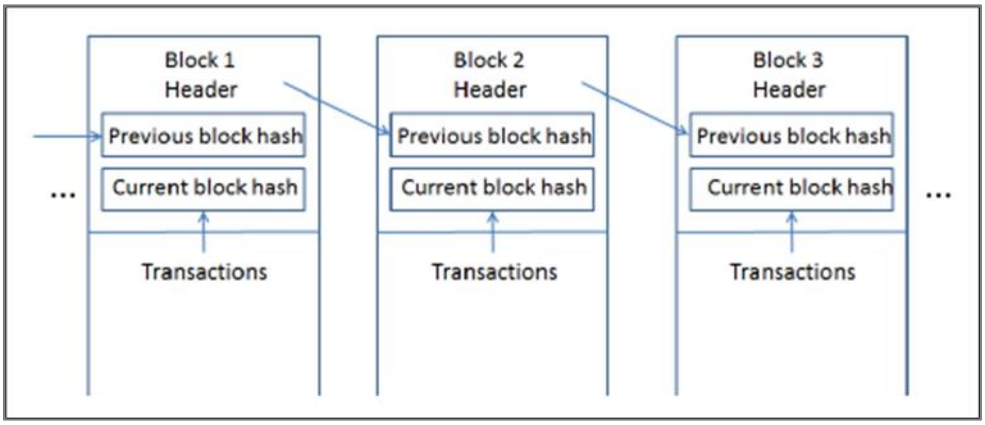


Figure 2. 1. Simple Blockchain Structure [14].

2.1.2.2. Consensus Algorithm

Consensus in the Blockchain is a strategy that a group of computers in the Blockchain network used to agree with each other on what is the truth. Thus, all nodes of the network agree on only one version of the truth about the ledger that they hold. The consensus mechanism aims to validate the transactions for appending a new block to the chain. Therefore, any tampered block will be rejected with preserving the network. There are several types of consensus algorithms such as Proof of Work (POW), Proof of Stack (POS), Proof of Authority (POA), etc. [15], [16].

2.1.2.3. Public Versus Private Blockchains

The main difference between a public and private Blockchain is the level of access granted to participants. Public Blockchain or referred to as ‘permissionless’ is completely open and allows anyone to participate by verifying or adding data to the Blockchain (a process called ‘mining’) such as Bitcoin and Ethereum and they are fully decentralized [17]. On contrary, the private Blockchain or referred to as ‘permissioned’, only allows certain authorized entities to participate in a closed network such as Corda and Hyperledger Fabric [18].

2.1.2.4. Key Characteristics of Blockchain

Most of the appeal toward Blockchain technology revolves around themes associated with its key features:

1. Decentralization: Any node in the network owns the information and has access to the data stored in Blockchain [19]. This allows network nodes to directly exchange data based on a trusted system. Thus, increasing the efficiency of data exchange and eliminating Single Point Of Failure (SPOF) [20].
2. Immutability: Means that once the data has been entered into the Blockchain, it cannot be modified [12]. The reason why the Blockchain gets this property is that of the cryptographic hash function. Thus restrict all unauthorized changes [21]and hacks in the system and removes the intermediates from the system.
3. De-trusted: The Blockchain creates linked blocks based on cryptographic hash values and uses digital signatures generated from asymmetric cryptography to ensure the security of transactions [13]. Therefore, the nodes can make transaction safely without third party control.
4. Privacy: The user is completely invisible during transmission process because the data are transmitted using public and private keys due to the digital signature algorithm [13].

2.1.2.5. Blockchain Platforms

Although Distributed Ledger Technology (DLT) is initially developed to support cryptocurrencies, the great features provided by this technology in data integrity, provenance, and authenticity have opened great new opportunities for developers to use this technology across a wide range of industry applications such as healthcare, education, the internet of things, etc [22]. With the increasing number of Blockchain-based applications, the DLT platforms has been emerged. DLT platforms facilitates to build and develop decentralized applications (DApp) on top of P2P network. Table 2.1 present a comparison of Blockchain platforms. Where DApp is the acronym for decentralized applications that enable access to the Blockchain features and services, where users can exchange information and transaction on top of a P2P network without any intermediaries. As opposed to a standard web application, a distributed application has no central server for storing data or performing computations. Instead, all computation and data storage are handled by transactions on a Blockchain network. The transactions are executed and stored in all nodes in the network [23]. The DApp consists of front end and Blockchain backend. The backend uses Blockchain infrastructure and returns any response to the web or mobile frontend [24].

Table 2. 1. Comparison between Blockchain Platforms.

	Cryptocurrency	Smart Contract	Consensus Algorithm	Trading Mechanism	Currency Issue	Technical Difference	Transaction Confirmation Time
Ethereum	ETH	Yes	POW or POS	Online trading on stock exchanges	The currency is decentralized and is not issued by a particular bank, it is a numerical value that increases based on certain criteria such as stock exchange expectations or participation in prospecting	Transactions on the Ethereum network may contain executable code	Average transaction time is around 16 seconds
Bitcoin	BTC	No	POW	Online trading on stock exchanges	The currency is decentralized and is not issued by a particular bank, it is a numerical value that increases based on certain criteria such as stock exchange expectations or participation in prospecting	Data affixed to Bitcoin network transactions are generally only for keeping notes	Average transaction time is 10 minutes
Ripple	XRP	Yes	Ripple Protocol Consensus Algorithm	A system for the direct transfer of assets (such as money, gold, and land)	Allows everybody to use the platform to create their own via RippleNet. uses a distributed consensus ledger using a network of validating servers and crypto tokens called XRP	Fast and cheap international transactions Payment ecosystem.	Average transaction time is 4 seconds

2.1.3. Identity and Identity Management

The increased usage of online services leads most of the population today to have a kind of digital identity. The digital identity refers to the personal identity that is created by individual in cyberspace [4]. The identity is established and maintained by that person. Whereby the complete personal identity is the combination of all his attributes. The user identity is the general name given to the profile information in the user's account such as username, email address, birthday, preferences, behaviors, and other information that is distinct to each user.

The involvement of digital identity in almost all online services contributes to the growing reliance on IDMS or referred to as identity and access management. IDMS broadly refers to the framework of policies and technologies designed to ensure that only authorized users have access to associated resources. Also, it facilitates managing and securing users' digital identities and provides relevant services such as authentication [25].

IDMS consists of three main entities [26]:

1. **User:** The subject or the owner of certain attributes or credentials and could use various services provided by service providers and identity providers.
2. **Service Provider:** Is an important entity of the management system, responsible for providing services to successfully authenticated users.
3. **Identity Provider:** The issuer of identity information for users. it is the core entity of the management system, responsible for holding identity information and providing users with identity services as following:
    - IdM which are registration, authentication, and managing users' attributes.
    - Identity reset in case of identity lost.
    - Identity revoke.

2.1.4. Evolution of Identity Management Models

The main IdM models will be discussed with highlights of their advantages and limitations. Also, presented the next Blockchain-based approach for IdM that targets user-centricity and eliminates the identity provider as a trusted third party.

2.1.4.1. Independent Identity Model

Also known as isolated IdM. In this model users didn't have their own identities, they only had accounts on a different service provider. Every service provider has its own identity provider as shown in Figure 2. 2. The identity provider assigns a unique identifier for each user. identifiers such as username and password [4]. Although the structure is simple, it requires a high storage capacity for each service provider. Also, the user needs to repeat the registration process which drives him to reuse the same password for many service providers. This creates a security concern as a compromise at one service provider can result in account hijacking at a different service provider. Moreover, the user needs to manage all his fragmented accounts among different service providers.

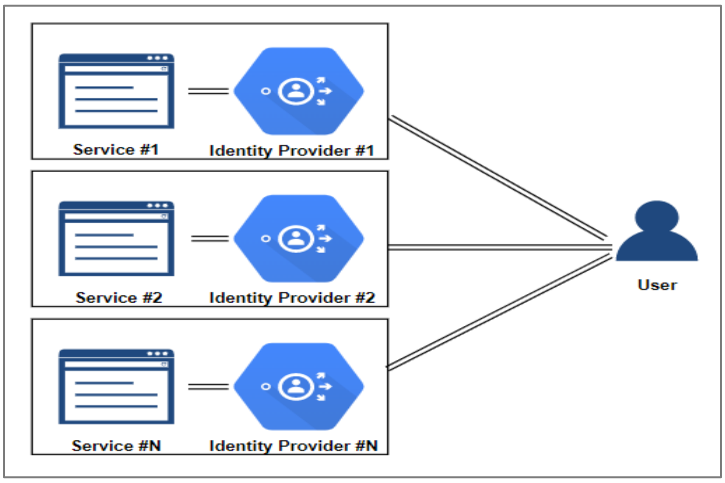


Figure 2. 2. Independent Identity Model.

2.1.4.2. Centralized Identity Model

In this model, only one identity provider as a separate entity within a trusted domain is responsible for both identification and authentication. Thus, allowing any service provider belonging to the trusted domain to share users' identities. The users' credentials are verified by a central authority. In the identification process, the user needs to identify himself to the identity provider. The identity provider verifies the user's identity through an authentication process. After completion of authentication, the user receives a token from the identity provider, and he passes the token to the service provider. Then the service provider verifies the user credentials that are carried in the user's token by querying the identity provider. After successful validation, the user can use the requested service within a certain amount of time that is determined in his token [27]. Figure 2.3 illustrate the process in the centralized identity model.

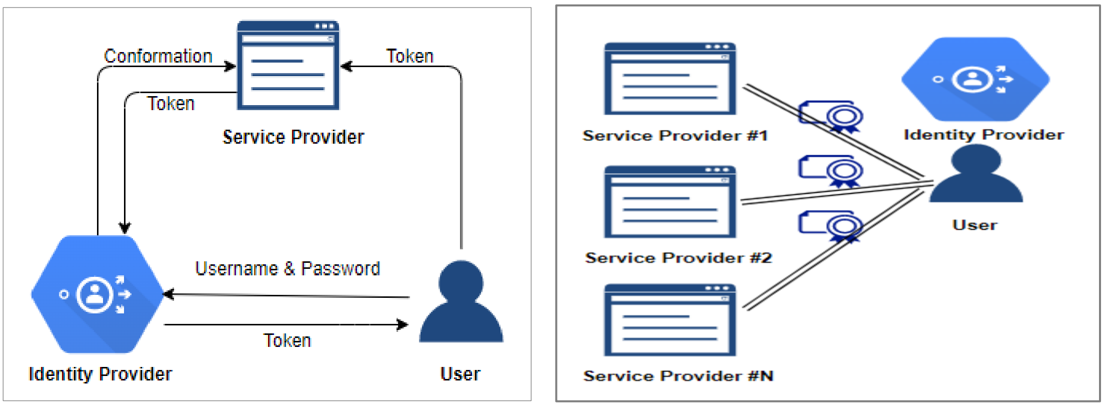


Figure 2. 3. Centralized Identity Model.

2.1.4.3. Federated Identity Model

In this model, multiple service providers within a trusted domain called federation agreed to work together to confederate and share their users' identities information [26]. Thus, allowing any service provider belonging to the federation to identify users easily. We know this on the web as social login using Google or Facebook, etc. The high-level architecture of the federated identity model is presented in Figure 2. 4. This model allows users to sign up once and carry their identity information to other service providers by using the same set of credentials. Thus, reduces the number of passwords needed to access all services down to one. In workforce scenarios, the identity provider is managed by IT staff at the enterprise and users are employees acting on behalf of their company. From a trust standpoint then there is an implicit trust relationship between the identity provider and the user. However, In the consumer landscape, single sign-on is unevenly applied. If it is available, it is characterized by centralized which makes signup and sign-in much more seamless. but on the other hand, the trust relationship between an identity provider and users little, rise to privacy and data protection concerns.

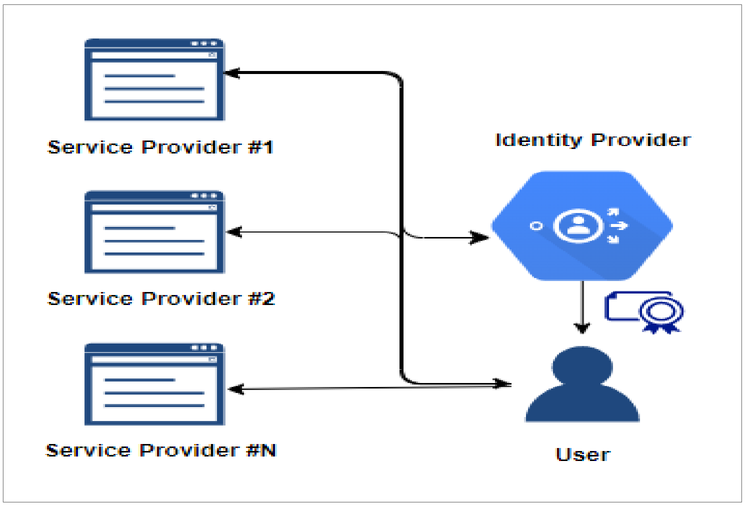


Figure 2. 4. Federated Identity Model.

2.1.4.4. Towards Decentralized Identity Management

Decentralized IdM aims to rectify privacy and data protection concerns by putting the control in the user's hands. The user control is enabled by shifting the transfer of identity information through users, rather than directly between service providers. [28]

2.1.5. Classification of Blockchain-Based Identity Management

According to Dunphy et al. in [29] all distributed ledger technology-based identity IdM proposals fell into one of two categories:

- **Self-Sovereign Identity** gives the individuals ownership and full control of their identities without the need for identity providers. The provided decentralized identity does not depend on any centralized registered identity provider or certificate authority (CA). Individuals can decide what to share, who to share with, and when to stop sharing their personal data. SSI enables trusted interactions to access individuals' identity information while preserving privacy. This can be enabled by an ecosystem that facilitates the collection and recording of users' attributes. Also, the ecosystem spreads mutual trust between different digital identities. Digital identities can be for institutions, individuals, and devices. Examples include uProt.

- **Decentralized Trusted Identity** relies on existing trusted credentials such as government identification cards or passports etc. Thus, the proprietary service will be able to perform identity proofing to verify these credentials. Then it stores the identity verification proofs on Blockchain for later validation by third parties. Examples include ShoCard.

2.2. RELATED WORK

The early literature on Blockchain makes frequent references to SSI and the individual's ability to own and control his or her own identity online, public Blockchains facilitate SSI by giving individuals the ability to be the final arbiter of who can access and use their data and personal information. Here in this section, some related works will be presented and briefly discuss their model, their advantages, and weaknesses. Mainly two solutions will be discussed in detail, each one belonging to a different category. The first solution is ShoCard which belongs to a decentralized trusted identity model. The second one is uPort which is the first existing identity solution that enables SSI. Furthermore, there are many other Blockchain-based identity systems in the literature, including SCPKI that relies on Public Key Infrastructure (PKI), DNS-IdM which serves online users in general while Health-ID serves patients and remote healthcare providers.

2.2.1. ShoCard

ShoCard [30] is a Blockchain-based digital identity solution that binds an existing trusted credential with a user identifier and attributes together via cryptographic hashes stored in the Bitcoin ledger. The trusted credential can be a passport, driving license, etc. ShoCard provides identity verification for both online and face-to-face interactions.

Users should first scan their identity credentials using ShoCard mobile application. The trusted credential can be a passport, driving license, etc. The uploaded credentials and the corresponding data are encrypted and kept on the user's mobile device. The signed hash of the user’s identity information is added into Bitcoin ledger to be used later for data validation. The resulting Bitcoin transaction number is the user identifier or known as ShoCardID and it is stored on the user's mobile device to be used as a pointer toward the ShoCard seal. In the certification process, the user collects additional attributes from many service providers. Then interacts with an identity provider to associate certificates to his ShoCardID. ShoCard server stores encrypted certifications or known as envelopes to give users the ability to provide their attributes to the relying parties or to retrieve them in case they lose their mobile device. Figure 2.5 illustrates the general architecture of ShoCard.

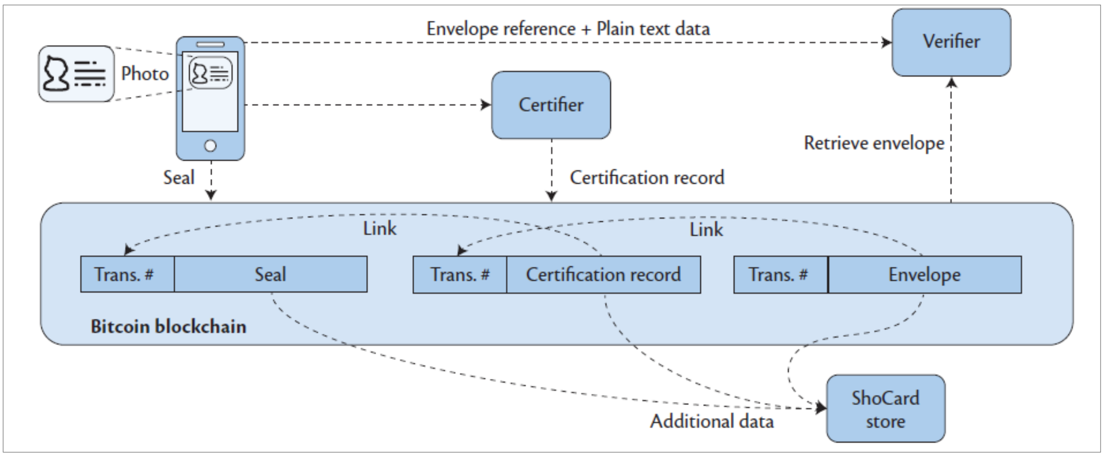


Figure 2. 5. ShoCard Architecture [31].

2.2.2. uPort

uPort [6] is an open-source decentralized IdM system that provides a digital identity to all internet users to interact with both DApp as well as traditional centralized applications.

uPort is built on the top of Ethereum ledger and relies on a set of components: smart contracts, developer libraries, and a mobile application. The developer libraries for third-party applications integration. The mobile application for cryptography asymmetric key pair management and for scanning the Quick Response (QR) code to initiate interactions with entities. Users are uniquely identified by a 160-bit hexadecimal address of the Ethereum smart contract deployed by the user and known as a proxy smart contract. Ethereum smart contracts are the core component of uPort technology. It has four main smart contracts: proxy, controller, recovery quorum, and registry smart contract. The proxy smart contract is used to forward transactions. The controller smart contract is used to maintain control access over the proxy contract. The controller contract consists of user's public key and a list of trusted entities addresses also known as recovery delegates. The recovery quorum smart contract is used to recover user's identity by triggering a vote between recovery delegates listed in the controller contract. When a quorum of delegates is reached within a specific period, meaning more than half of the recovery delegates have been positively voted. Then a new user address is replaced with the lost public key. The new address is connected to a new mobile device. The registry smart contract is used for mapping between uPort identifiers with their associated identity attributes. The attributes are stored off-chain on InterPlanetary File System (IPFS) which is a distributed storage system or on any traditional cloud service such as Microsoft OneDrive and Dropbox. The cryptographic hash of the JavaScript Object Notation (JSON) attribute data is only stored on-chain due to the high cost of large volumes. Figure 2.6 illustrates the general architecture of uPort.

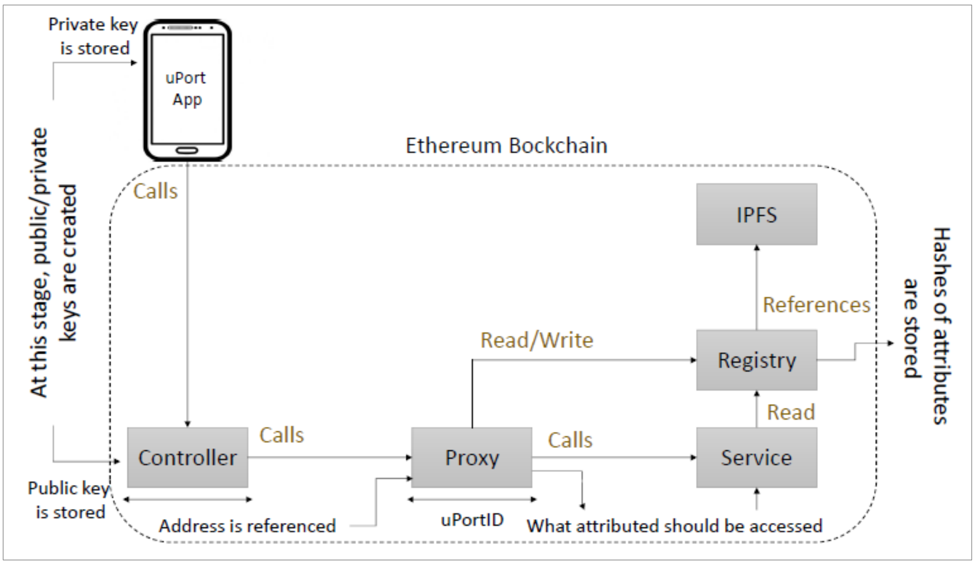


Figure 2. 6. uPort Architecture [32].

2.2.3. SCPKI

SCPKI [33] addresses the issue of rogue certificates when an attacker can get a copy of the CA private key and used it to sign certificates. An alternative to the PKI system is proposed to detect issued rogue certificates. Also, an identity system to manage the storing, retrieving, and verifying for entities attributes, signatures, and revocations in a web of trust by utilizing Blockchain technology.

SCPKI is built on the top of Ethereum, each entity is uniquely identified by an Ethereum address that is associated and controlled with the owner of a private key. However, the system has no access control on the entities' attributes, which means that it is visible to anyone in the system. Therefore, it is unsuitable for private attributes. Also, a cryptographic hash of the data with the attribute is on the Blockchain and the data itself is stored off-chain. Furthermore, all system parties must use the system in order to attach attributes to users.

2.2.4. DNS-IdM

DNS-IdM [32] is a smart contract-based IdM system that enables users to manage and trace their identity attributes. Also, facilitates the verification process for the service providers by using real-world identity attribute benefactors.

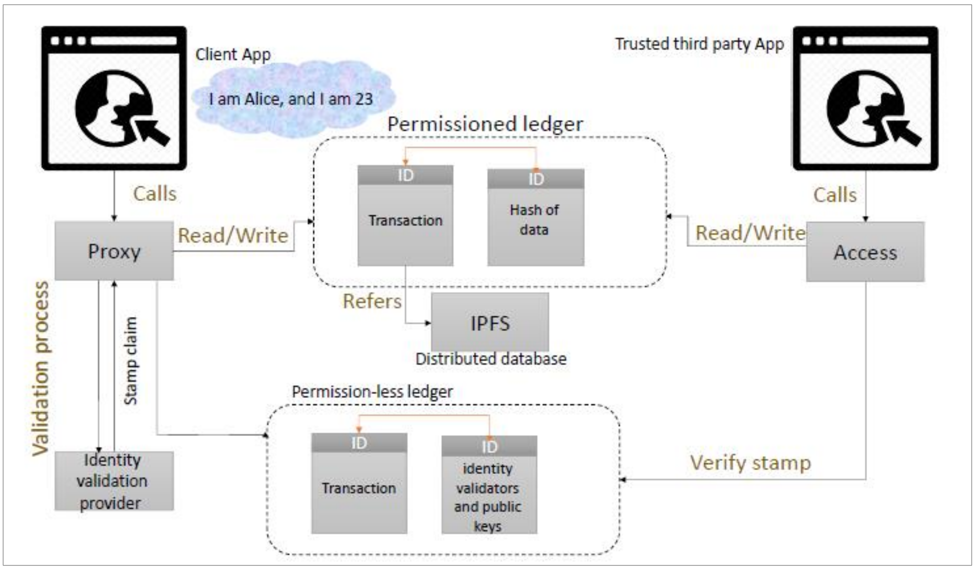


Figure 2. 7. DNS-IdM Architecture [32].

Users should register first to be able to add their attributes. The attributes itself stored on IPFS, while the hash of attributes and the identification data are stored on a permissioned Blockchain. Attributes are validated before being mined and added to the network. Therefore, DNS smart contract plays its role as a router and redirects to a specialized validation contract based on the type of attributes. The validation contracts and public keys are stored on a permissionless Blockchain. Besides that, DNS contract grants public access to the entries on a permissionless network. Figure 2.7 illustrates the design architecture of DNS-IdM.

2.2.5. Health-ID

Health-ID [34] is a Blockchain-based decentralized IdM solution that serves both patients and remote electronic healthcare providers to securely identify and authenticate themselves across different eHealth domains without relying on a central service provider.

There are three participants in the proposed system: regulators, patients, and healthcare providers. The regulators manage the Blockchain. The patients can create, store, and manage their own identity. Healthcare providers can authenticate themself to any patient before providing their health services. The participants will be registered to the Blockchain after performing off-block identity proofing. As a result, the patients and healthcare providers will have a unique identification called healthID which is the address of the smart contract deployed by each entity. The identity attributes are structured in form of a JSON object and then signed by the regulator to create a JSON Web Token (JWT). The owner uploads the encrypted JWT identity attributes over a cloud service (Dropbox, IPFS). The hash of the identity attributes and the hashID which is a unique random number assigned to that hash are further stored on the Blockchain. Figure 2.8 illustrates the general architecture of Health-ID.

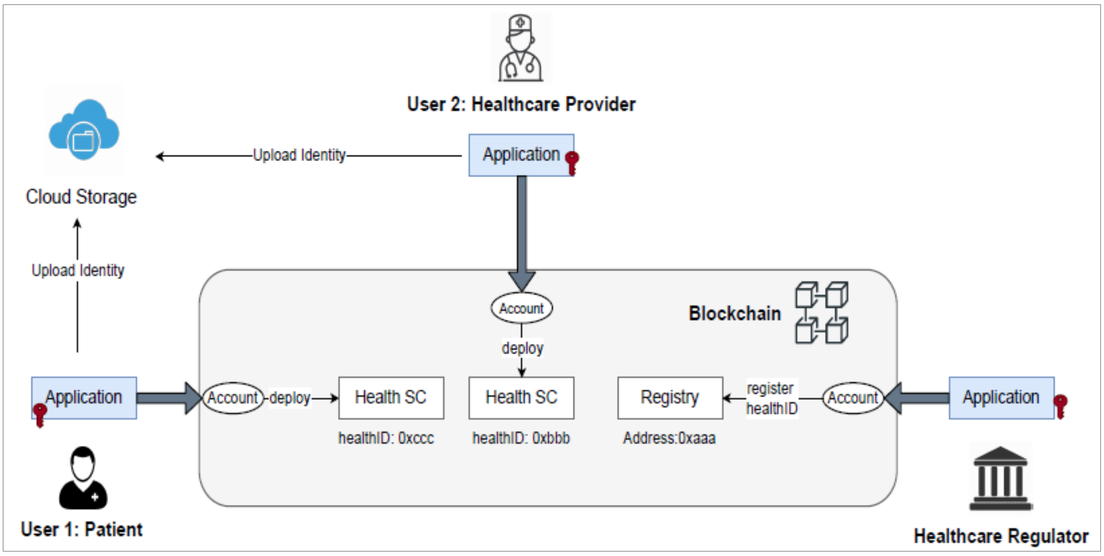


Figure 2. 8. Health-ID Architecture [34].

2.3. DISCUSSION & COMPARISON BETWEEN THE RELATED WORKS

As a conclusion of the above IdM solutions, an extensive assessment will be carried out in this section. The assessment is divided into three main parts, namely: technology used, identity services provided, and security-based assessment.

2.3.1. Technology-based Assessment

Table 2.2 presents a comparison between above IDMS based on the technology used.

Table 2. 2. Comparison between decentralized IdM solutions.

Identity Solution	**Year**	**Distributed Ledger**	**Blockchain Type**	**Consensus Algorithm**	**Identity Model**
ShoCard	2015	Bitcoin	Permission(less,ed)	≈	Decentralized Trusted
uPort	2016	Ethereum	Permissionless	POW	Self-sovereign
SCPKI	2017	Ethereum	Permissionless	≈	Self-sovereign
DNS-IdM	2019	Ethereum	Permission(less,ed)	≈	Self-sovereign
Health-ID	2021	Ethereum	Consortium	POA	Self-sovereign

≈ Not Addressed

Ethereum platform was used in all discussed IdM solutions except ShoCard which used Bitcoin. Due to the use of the Bitcoin ledger, the transaction confirmation time takes on average ten minutes compared to seconds for Ethsereum. As result, the waiting time for ShoCard users will be very high which negatively affects users' experience.

uPort and SCPKI use permissionless Blockchain which is completely open and allows anyone to participate by verifying or adding data to the Blockchain and they are fully decentralized. On the contrary, the permissioned Blockchain that only allows certain authorized entities to participate in a closed network.[13] Thus, a permissioned Blockchain is considered to be centralized but it is faster, more scalable, and transaction fees are extremely low. ShoCard and DNS-IdM get the power of the permissioned and permissionless Blockchain. Health-ID use a consortium Blockhain that considered as hybrid type of blockchain, relies on a set of authorized regulators to manage the network.

Consensus algorithm allows nodes of the Blockchain network to agree on only one version of the truth about the ledger that they hold. uPort uses POW that required expensive energy computational to reach a consensus on the state of the ledger. While Health-ID uses POA which is less energy cost than POW and significantly improves transaction throughput.

2.3.2. Identity Services-based Assessment

Identity provider is the core component of the IDMS, providing users identities and other related identity services namely identity revoke, recovery in case of loss, and IdM such as registration, authentication, and managing users' attributes [26]. Table 2.3 shows the identity services that are provided by the above identity solutions. all of them provide registration, authentication, and management. However, none of them enables identity revocation. Where revoking an identity means that the entity is no longer participating or interacting in the system. In this research, all identity services will be addressed in the proposed a framework.

Table 2. 3. Identity services comparison between identity solution.

Identity Solution	**Registration**	**Authentication**	**Manage Attributes**	**Identity Proofing**	**Identity Recovery**	**Identity Revoke**
ShoCard	✓	✓	✓	✓	×	×
uPort	✓	✓	✓	×	✓	×
SCPKI	✓	✓	✓	×	×	×
DNS-IdM	✓	✓	✓	×	×	×
Health-ID	✓	✓	✓	✓	×	×
Proposed Framework	✓	✓	✓	✓	✓	✓

A comparison between related works and the proposed framework based on identity services is discussed below.

2.3.2.1. Registration

The process of adding new identities into the system where entities could register and identify themselves to the system. All IDMS should uniquely identify entities in the system in order to distinguish between these identities. ShoCard identifies its users using ShoCardID which is the Bitcoin transaction number that is produced while storing the signed hash of the user's identity information in the Bitcoin ledger. uPort uses the address of the proxy smart contract to identify its users. Each entity in SCPKI and DNS-IdM is represented by an Ethereum address. Health-ID identifies patients and healthcare providers using healthID which is the address of the smart contract deployed by each entity. The proposed framework uniquely identifies Metaverse platforms using a cryptographic public key while Metaverse users are uniquely identified using the public key and fingerprint value.

2.3.2.2. Identity Proofing

Identity proofing is the process of verifying that the user is actually who is claimed to be [35]. This process is required to avoid any identity theft. ShoCard performs identity proofing by canning existing trusted credentials. However, the uploaded credentials may be fraudulent. Also, Health-ID performs off-block identity proofing for healthcare regulators to validate the identity information. After verifying their identity successfully, then healthcare regulators can provide physical or remote identity proofing for patients and healthcare providers. On the other hand, uPort and DNS-IdM do not perform any identity proofing.

The proposed framework performs identity proofing for both Metaverse users and platforms before adding their identities to the system. Identity proofing for users depends on email addresses while identity proofing for platforms depends on domain names. This process is performed using a consortium of trusted anchors and it does not depend on the physical presence of users and platforms to prove their identities.

Therefore, it dramatically reduces the time taken to register a new identity to the system.

2.3.2.3. Authentication

Authentication is the process of verifying the identity of an entity before it accesses the system. ShoCard verifies users before they access the relying party services using the users' envelope reference and their encryption keys. uPort works as an authentication platform for both centralized web applications and decentralized applications. However, uPort does not authenticate the owner of the mobile device, meaning if an unauthorized person has access to the user’s mobile device, he will have full control of his identity. SCPKI provides authentication by proving that the owner of a private key is associated with the user's Ethereum address (i.e. user's identity). DNS-IdM support address-based authentication using the proxy smart contract where the contract checks if the returned address matches the claimed address. Health-ID supports single sign-on using a validated user's healthID and identity token. The proposed framework utilizes a cryptographic challenge-response protocol for authenticating Metaverse users and platforms.

2.3.2.4. Manage Attributes

Recording attributes is the process of managing, storing, and retrieving users' data. ShoCard stores the encrypted form of certifications or known as envelopes on a central server. uPort, SCPKI, and DNS-IdM store the attributes data itself off-chain on IPFS, while Health-ID stores the encrypted JWT attributes over a cloud service. Only the hash value of attributes is stored on the Blockchain. That’s useful for verifying the integrity of the attributes but cannot guarantee recovery of attributes in case of loss or damage. The proposed framework stores users' attributes in the blockchain instead of storing just the hash value of attributes, and that guarantees the availability and integrity of data.

2.3.2.5. Identity Recovery

Identity recovery mechanism is the process of restoring users' identities with all previous attributes associated with those identities. Identity recovery is performed when users lost their identities, which means they lost their keys or lost their mobile devices. This allows users to maintain a persistent identifier even in case of lost cryptographic keys. uPort enables users to recover their identity if their mobile device that holds the user's private key is lost or theft. uPort provides an identity recovery mechanism by triggering a vote between recovery delegates listed in the controller smart contract. The recovery delegates list can be friends, family members, institutions, etc. However, this mechanism would be vulnerable when the trusted recovery delegates themselves are attackers or malicious entities by replacing the recovery delegate's address with their own identities leads to compromising the user’s device key permanently and stealing his identity. Moreover, there is a potential for leakage of attributes in the registry.

The proposed framework utilizes fingerprint biometrics to perform the identity recovery process. Once the user's fingerprint has been scanned during the registration process, it will be stored in the Blockchain in encrypted form and linked to the user's identity. When the user lost his mobile device, which means he is lost his cryptographic keys, he can send a request to the identity manager smart contract to recover his identity.

2.3.2.6. Identity Revoke

Identity revoke is the process of preventing users from accessing, participating, or benefiting from system services. None of the above-discussed identity solutions addresses identity revocation completely. The proposed framework enables trusted anchors to revoke users' identities in case of detected bad behavior from users such as defrauding, violating, harm to another, or identity theft. If any user violates the policies or his bad behavior has been reported, then the trusted anchor will take appropriate action and block him from accessing or participating in the system. This is done when a trusted anchor sends a request to the identity manager smart contract to revoke the user's identity. The identity manager contract lookup the identity of the user and make it no longer valid.

2.3.3. Security-based Assessment

Several distributed IDMS are geared toward taking advantage of intermediaries instead of eliminating them by reshaping their roles. For example, uPort relies on trusted attribute providers and uses a registry that stores the mapping between uPort identifiers with their associated identity attributes. Also, ShoCard uses a central server as an intermediary to manage the exchange of user certifications between users and different relying parties. However, if any security breaches happened or if the company no longer existed, users would be unable to exchange certifications between different relying parties.

Some Blockchain-based IDMS support creating multi-unlinkable identities for the same user. For example, ShoCard, uPort, SCPKI, and DNS-IdM provide creating multiple identities for one user while Health-ID supports only a single identity for the user.

CHAPTER 3

RESEARCH METHODOLOGY

This chapter covers a detailed explanation of DSR methodology that is being used to make the propped framework complete and work well. This methodology is used to achieve the objective of the research. Moreover, technical knowledge and used software are presented to underline the main technologies used to build the framework.

3.1. RESEARCH METHODOLOGY

DSR methodology was followed by a set of well-defined steps to develop an Information Technology (IT) artifact in form of a framework that hands back the control of the digital identity to the individuals. DSR is a set of principles, techniques, and procedures that provides an organized path to producing objects known as artifacts [36]. Where the artifact represents a solution to a problem at hand. In this context, the DSR methodology is divided into an iterative and cyclical six phases involve (1) problem identification and motivation; (2) definition of solution objectives; (3) solution design and development; (4) solution demonstration; (5) solution evaluation; (6) communication. Figure 3.1 depict the DSR phases of the proposed framework.

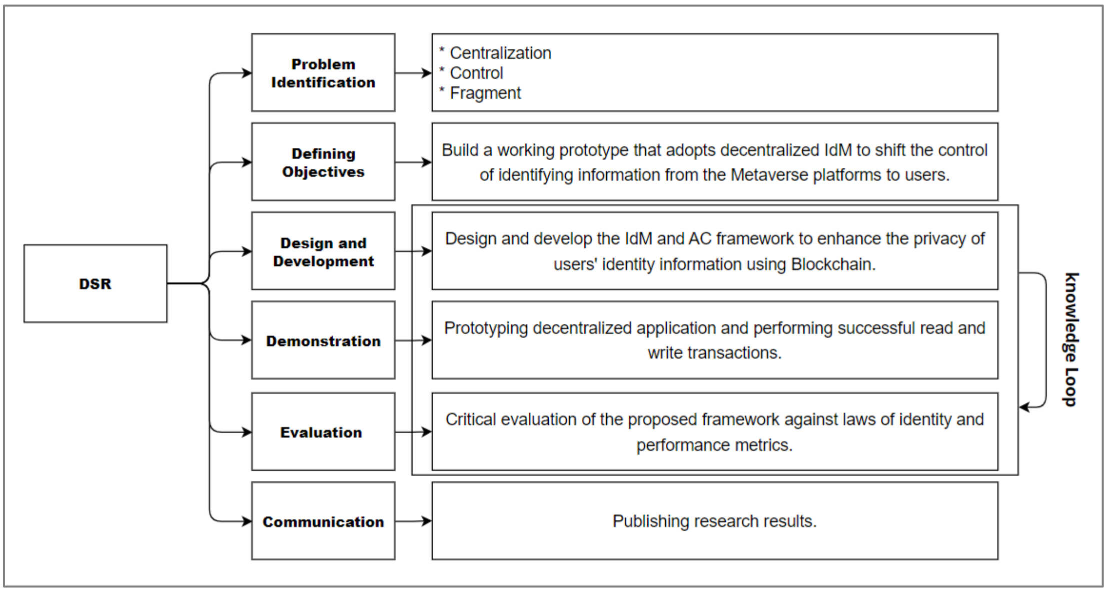

DSR
Problem Identification
* Centralization
* Control
* Fragment
Defining Objectives
Build a working prototype that adopts decentralized IdM to shift the control of identifying information from the Metaverse platforms to users.
Design and Development
Design and develop the IdM and AC framework to enhance the privacy of users' identity information using Blockchain.
Demonstration
Prototyping decentralized application and performing successful read and write transactions.
Evaluation
Critical evaluation of the proposed framework against laws of identity and performance metrics.
Communication
Publishing research results.
knowledge Loop

Figure 3. 1. DSR Phases of the Proposed Framework.

3.3.1. Problem Identification

The problem has been defined after conducting literature reviews which is one way to gather data and gain insights into the problem. It was observed that individuals' digital identities are stored in central repositories. Centralization repositories may have significant security flaws causing various security breaches such as SPOF, identity theft, disclosure of sensitive information [2], and control by third-party entities that have the entire control of our personal information. The powerful entities could gather and abuse users' information without their knowledge or permission [4]. Furthermore, many Metaverse platforms provide immersive experiences in many domains. for example, gaming, education, and shopping. However, each of these Metaverse platforms has a centralized identity management system. Therefore, the user needs a username and password for every Metaverse platform as shown in Figure 3.2. Consequently, users won't be able to shuttle across various Metaverse platforms or sub-metaverses with their avatars and digital assets. After identifying the problem, the next step is to identify potential solutions.

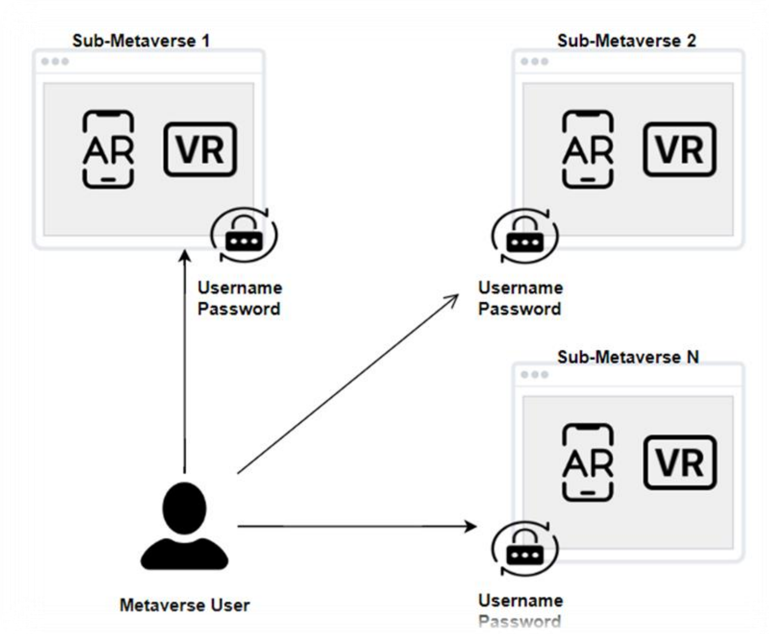


Figure 3. 2. Centralized Identity Management within Metaverse Platforms.

3.3.2. Definition of Objectives

The primary objective of this research is to is to build a working prototype that adopt decentralized IdM approaches for future Metaverse systems by shifting the control of identifying information from the Metaverse platforms to users to have the ownership of their own identity and increasing confidence in data immutability and availability.

3.3.3. Design and Development

To resolve the problem mentioned above, this research propose IdM and AC framework to enhance the privacy of users' identity information using Blockchain. In this section, the framework architecture will be described by defining the main components and their interaction.

3.3.3.1. Framework Architecture

The IdM and AC framework workflow is simple to use and its meets with user experience. Framework architecture is illustrated in Finger 3.2. First the Metaverse users and platforms should register in order to get benefit from identity services. Then the framework performs identity proofing to verify the real identity and to avoid any identity theft. This is done through communication with the trusted anchors who approve to continue the registration process. After successful registration, the user can freely add his attributes and determine his own AC policy that controls who from the registered platforms is allowed to access his attributes.

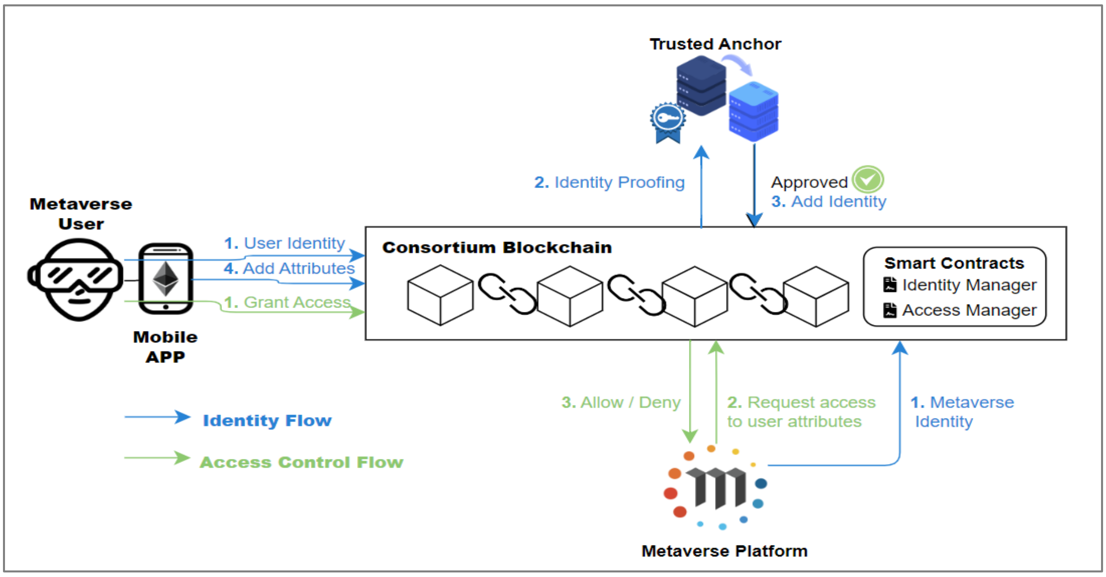


Figure 3. 3. Framework Architecture.

The framework components are described in detail below:

- **Metaverse User:** The owner of SSI that has full control to specify which of his stored information wants to share with any Metaverse platform.

- **Metaverse Platform:** The service provider that offers Metaverse services for verified users after they are identified and authenticated successfully.

- **Trusted Anchor:** A consortium of trusted anchors manages the Blockchain. They are responsible for the user's identity proofing process to verify that the user is actually who is claimed to be. This process is required to avoid any identity theft. Moreover, they are responsible for revoking the identity of users.

- **Mobile Application:** used as an interface to interact with Blockchain.

- **Blockchain Network:** The proposed framework uses a PoA consensus agreement to deploy the consortium Blockchain. in the PoA consensus algorithm, the blocks and transactions are verified by pre-approved participants that are selected based on their reputation. Thus, a specific predefined validation node should be selected first to validate each transaction. Each member of trusted anchors consortium manages a validator node in the network. There are two smart contracts deployed over Ethereum network, namely as followed:

a. Identity Manager contract used for registering and administering for both users and Metaverse platforms accounts. Furthermore, it is responsible for identity recovery.
b. Access Manager contract responsible for storing users' attributes in Blockchain. Also, it manages platform's permissions. Users can allow or deny Metaverse platforms to access their information.

3.3.3.2. Framework Development

The propped framework has been implemented in a consortium Blockchain using clique as a POA consensus algorithm. the consortium Blockchain was used instead of a private Blockchain because the network is controlled by a group of trusted anchors in contrast to a single entity. The simulated Blockchain network present in Figure 3.3. consists of the following node:

a) Bootnode Node: a private cluster predefined bootstape node that is used for connects participant network nodes to each other. Thus, any node should connect to this bootnode first to join the network and find the other nodes.
b) Non-Miner Nodes: local Ethereum clients used to receive data from Blockchain network.
c) Trusted Anchor Mining Node: this node is responsible for mining a new block which verify transactions and broadcast it to peer nodes.

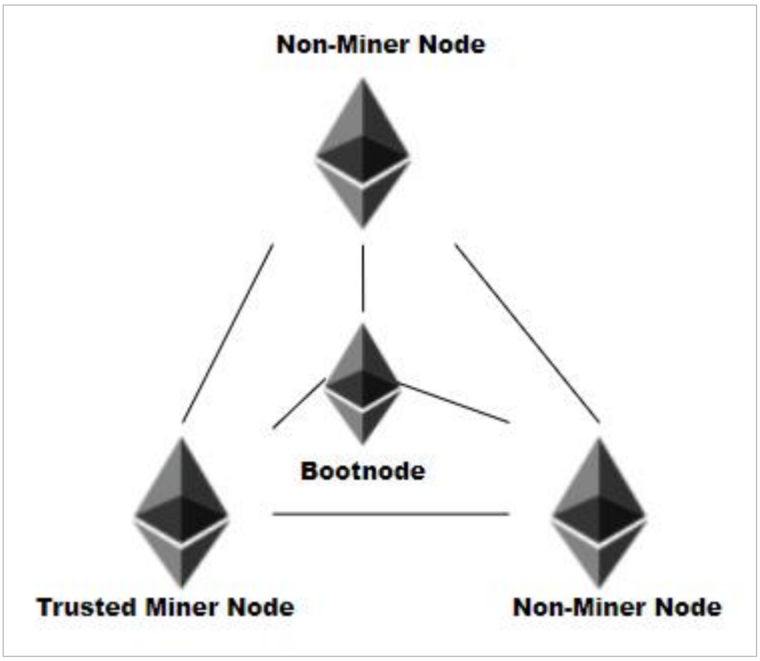


Figure 3. 4. Blockchain Network.

After initializing the network, the identity manager and access manager smart contracts were created and deployed over the Ethereum network. The architecture of the smart contract includes main two parts: contract deployment and contract interaction as shown in Figure 3.4. The first part aims to deploy the smart contract on Ethereum private network. The deployment process begins with compiling the smart contract source code using the Truffle framework. Figure 3.5. shows the solidity compilation process where it takes a Smart Contract source code as a parameter. The result of this process will be Application Binary Interface (ABI) which is a JavaScript interpretation layer of what the contract is and also the contract bytecode which is what actually deployed as instances of Smart Contracts in the EVM.

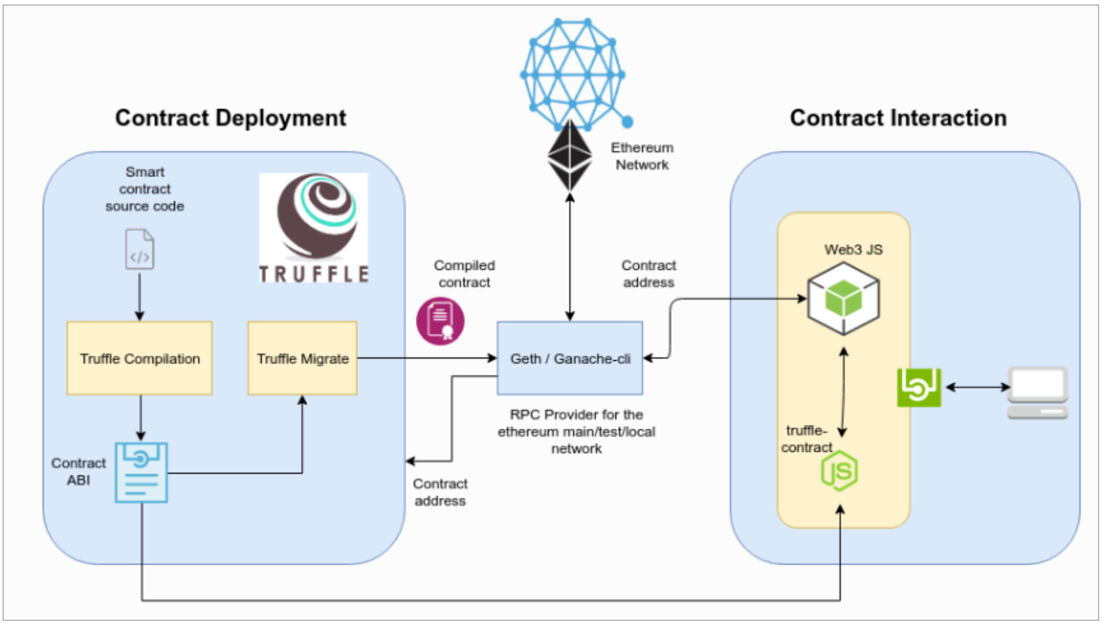


Figure 3. 5. Ethereum Smart Contract Deployment Architecture [38].

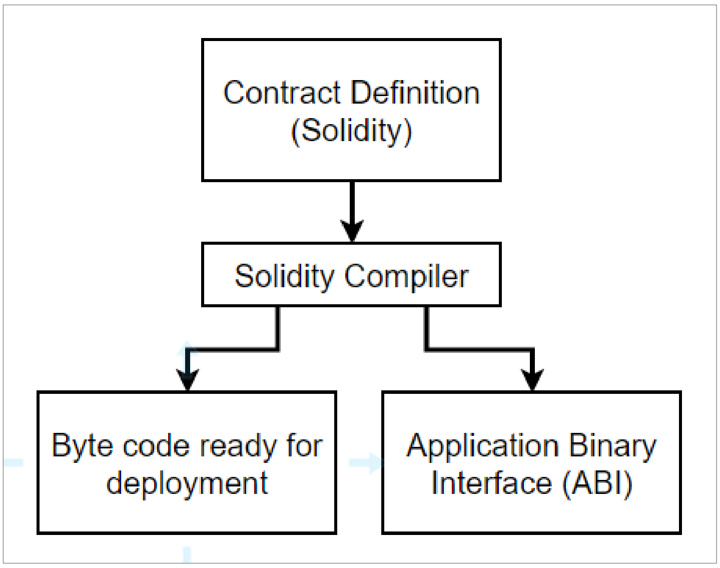

Contract Definition (Solidity)
Solidity Compiler
Byte code ready for deployment
Application Binary Interface (ABI)

Figure 3. 6. Solidity Compilation Process.

The Contract ABI which is going to be sent to Truffle migrate process that will deploy the smart contract into Ethereum private network. The contract address will be the output of the migration process. The contract interaction as part aims to interact with the Ethereum Smart Contract. This done by injecting Web3 Js library into UI pages and passing the address of deployment contact. Then any method on this contact can be called. Figure 3.6 shows how the Smart Contract can interact with DApp.

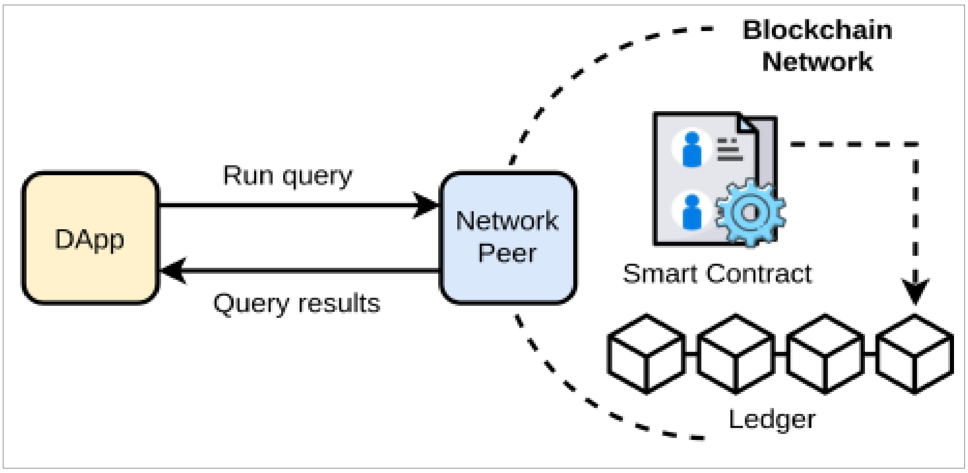


Figure 3. 7. Interaction Between Smart Contract and DApp [37].

After deploying the smart contracts into the built Blockchain network, a user-friendly mobile application was developed and linked to the Blockchain database. The users can easily interact and take advantage of the identity services that the framework provided.

3.3.4. Demonstration

This phase involves the use of prototype DApp to solve the problem described in the first phase. The created prototype is built to demonstrate that Metaverse users have the full control over their own identities by enforcing their AC policies. Thus, a Blockchain network was setup and DApp application was developed and linked to the Blockchain network. Appendix A shows the DApp mobile application user interfaces for the framework. After building the application, dummy transactions were submitted over the Blockchain network to be tested and evaluated.

3.3.5. Evaluation

The performance of the IdM and AC framework was analyzed on a consortium Blockchain using Ethereum. Clique network was used as the POA consensus algorithm. Windows 11 home operating system with 8GB RAM and Intel core i5 was used to implement the experiment. The identity and access managers smart contracts are written in solidity language version 0.8.11. Web3.js version 1.7.5 was used to send the identity transactions through the framework mobile application UI. Geth version 1.10.8-stable was used to set up the network of three nodes: the trusted anchor miner node and two receiver nodes i.e., receiving transactions. All nodes are connected and configured to regulate the Blockchain.

The eth-netstats and eth-net-intelligence-api were used to track and monitor the status of the built Ethereum network. The network status is shown through a visual interface using eth-netstats. while the eth-net-intelligence-api is the backend service to fetch network information from running Ethereum nodes. Figure 3.7. shows the monitor architecture and Figure 3.8 dashboard of our experimental network.

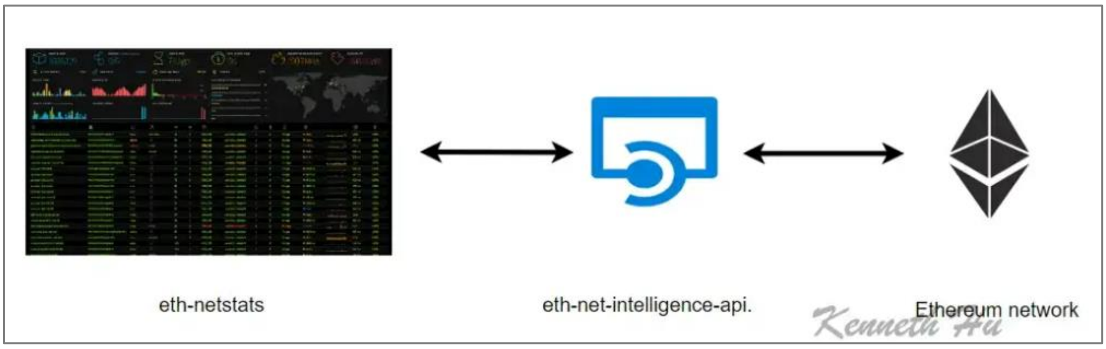
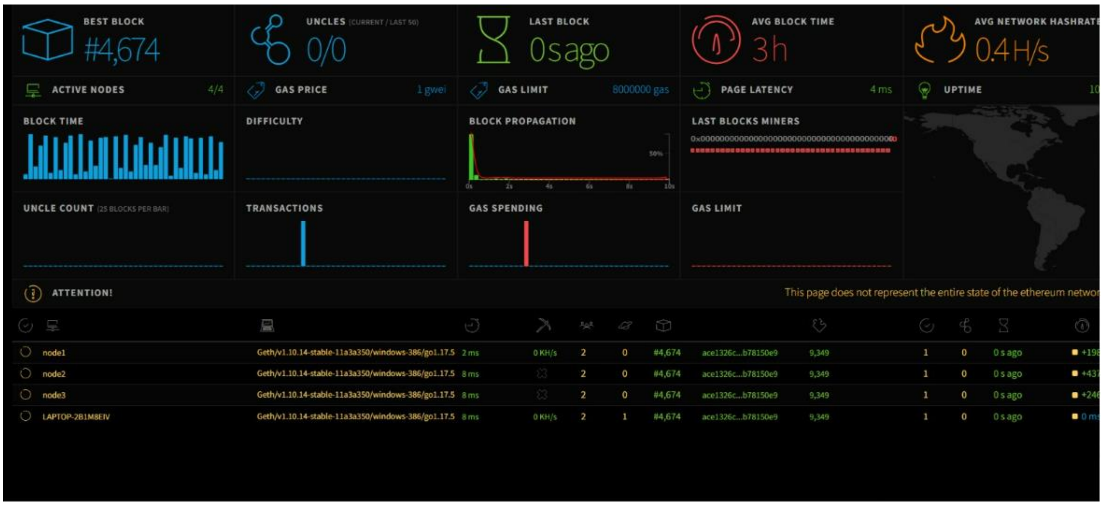


Figure 3. 8. Ethereum Mointor Architecture [38].


BEST BLOCK #4,674

UNCLES (CURRENT / LAST 50) 0/0

LAST BLOCK 0s ago

AVG BLOCK TIME 3h

AVG NETWORK HASHRATE 0.4H/s

ACTIVE NODES 4/4

GAS PRICE 1 gwei

GAS LIMIT 8000000 gas

PAGE LATENCY 4 ms

UPTIME 10

BLOCK TIME

DIFFICULTY

BLOCK PROPAGATION

LAST BLOCKS MINERS

0x0000000000000000000000000000000000000000

UNCLE COUNT (25 BLOCKS PER BAR)

TRANSACTIONS

GAS SPENDING

GAS LIMIT

ATTENTION!

This page does not represent the entire state of the ethereum networ

node1	Geth/v1.10.14-stable-11a3a350/windows-386/go1.17.5	2 ms	0 KH/s	2	0	#4,674	ace1326c...b78150e9	9,349	1	0	0 s ago	■ +198
node2	Geth/v1.10.14-stable-11a3a350/windows-386/go1.17.5	8 ms		2	0	#4,674	ace1326c...b78150e9	9,349	1	0	0 s ago	■ +437
node3	Geth/v1.10.14-stable-11a3a350/windows-386/go1.17.5	8 ms		2	0	#4,674	ace1326c...b78150e9	9,349	1	0	0 s ago	■ +246
LAPTOP-2B1M8EIV	Geth/v1.10.14-stable-11a3a350/windows-386/go1.17.5	8 ms	0 KH/s	2	1	#4,674	ace1326c...b78150e9	9,349	1	0	0 s ago	■ 0 ms

Figure 3. 9. Mointor Dashboard of Built Ethereum Network.

The proposed framework will be analyzed under different performance metrics, including transaction gas cost, gas limit, block period, and throughput.

- **Transaction Gas Cost:** The Ethereum platform uses a system of costs called gas where any transaction that changes the state of the network will be charged. Gas refers to the cost needed to submit a transaction or contract over the Blockchain network, and it shows the complexity of the smart contracts.

- **Gas limit**: The maximum amount of gas consumers to send a write transaction on Blockchain.

- **Block Period:** In the PoA consensus, essential configurations must be made first before running the network including the block period which is the time needed to add a new block in a Blockchain. These seconds that the users should wait until the transaction data is stored in a new block. If the transaction could not be added to an existing block, the transaction will be included in the next block period.

- **Transactions Throughput:** Indicates the number of transactions that are successfully committed in network blocks within a defined period as shown in the equation below [39]. It is measured in transactions per second (TPS).

$$TPS = \frac{Count\ (T_x\ in\ (t_s, t_e))}{t_e - t_s}$$

Where T_x is the overall submitted transactions, t_s is the beginning submission time and t_e is the ending submission time where all sent transactions are committed successfully.

CHAPTER 4

IDENTITY MANAGEMENT AND ACCESS CONTROL FRAMEWORK

This chapter illustrates the operational workflow of the proposed IdM and AC framework and presents the main practical implementation of developing the simulation environment of the framework. After that, the environment setup is explained in detail. Also, this part analyzes the framework based on the laws of identity and discusses the performance of the deployed consortium Blockchain.

4.1. FRAMEWORK NETWORK

The proposed framework is geared toward taking advantage of intermediaries instead of eliminating them by reshaping their roles. Thus, a group of trusted anchors plays an important role in managing the consortium Blockchain network. They provide administrating services for Metaverse users including, identity-proofing for verifying users' identities and revoking the identity from users in case of violation of user's privacy. Examples of trusted anchors include universities, international non-governmental organizations, and public sectors. Each trusted anchor will manage a node in the consortium Blockchain network. A new trusted anchor can be included at any time based on the majority decision of the consortium.

4.2. IDENTITY MANAGEMENT SCHEME

The IdM scheme is designed to adopt Blockchain technology to provide identity services.

4.2.1. Registration

In the registration process, a new user or a Metaverse platform is registered in the system to have an account. The registration process for the user is different than the Metaverse platform. Figure 4.1 present user registration sequence diagram. The registration process for each one is presented separately below and described in Algorithms 1 – 2 respectively.

4.2.1.1. User Registration

a) A new user needs to download the mobile application.
b) The user chooses his trusted anchor for performing the identity proofing process.
c) Asymmetric keys will be generated for the user. The private key will be locally stored in the user device because is no means of exporting the private key out of the user's device.
d) The user sends a registration request along with his generated public key, trusted email address and fingerprint value to be used later for account recovery in case he lost his identity.
e) After conducting the identity proofing process successfully. The trusted anchor sends a request along with the user’s public key and his fingerprint value to the Identity Manager contract (IDMc) in order to add a new identity to the system.
f) The IDMc checks if the user's identity which is the generated public key and fingerprint value has already been registered in the system or not. If yes, the registration request will be rejected. Otherwise, it will store the new identity in the Blockchain network.

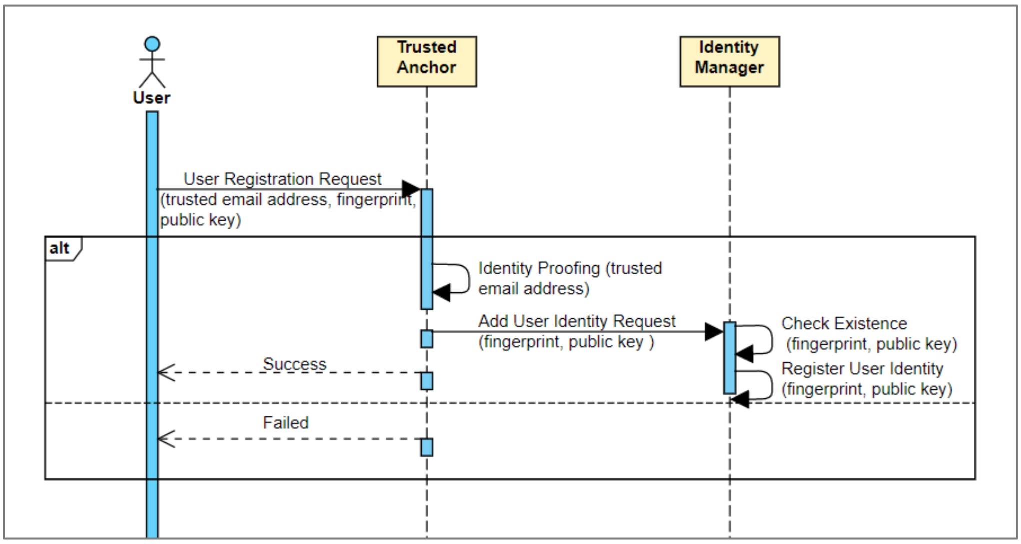

User
Trusted Anchor
Identity Manager
User Registration Request (trusted email address, fingerprint, public key)
alt
Identity Proofing (trusted email address)
Add User Identity Request (fingerprint, public key)
Check Existence (fingerprint, public key)
Register User Identity (fingerprint, public key)
Success
Failed

Figure 4. 1. User Registration Sequence Diagram.

Algorithm 1: User Registration

Input: Email address, fingerprint value, public key

Output: Add new identity to the system

```
Procedure Add user identity.
    1.  Identity proofing (email address)
    2.  If Identity proofing success
    3.      If identity does not exist (fingerprint value, public key)
    4.      Add user to the Blockchain network.
    5.      Else
    6.      Registration failed.
    7.      End if
    8.  Else
    9.  Registration failed.
    10. End If
End Procedure
```

4.2.1.2. Metaverse Platform Registration

The Metaverse platform registration is less complicated than the user registration because they just want to access and read users' attributes. Figure 4.2. shows the Metaverse platform registration sequence diagram.

a) Asymmetric keys will be generated for the platform. The private key will be stored locally in platform’s device.
b) The platform sends a registration request to the trusted anchor along with its domain and generated public key.
c) After conducting the identity proofing process successfully. The trusted anchor sends add platform request along with the generated public key to the IDMc.
d) The IDMc checks if the received public key has already been registered to an existing Metaverse platform account or not, if yes, it rejects the request. Otherwise, it will store the verifiable claim to the Blockchain network.

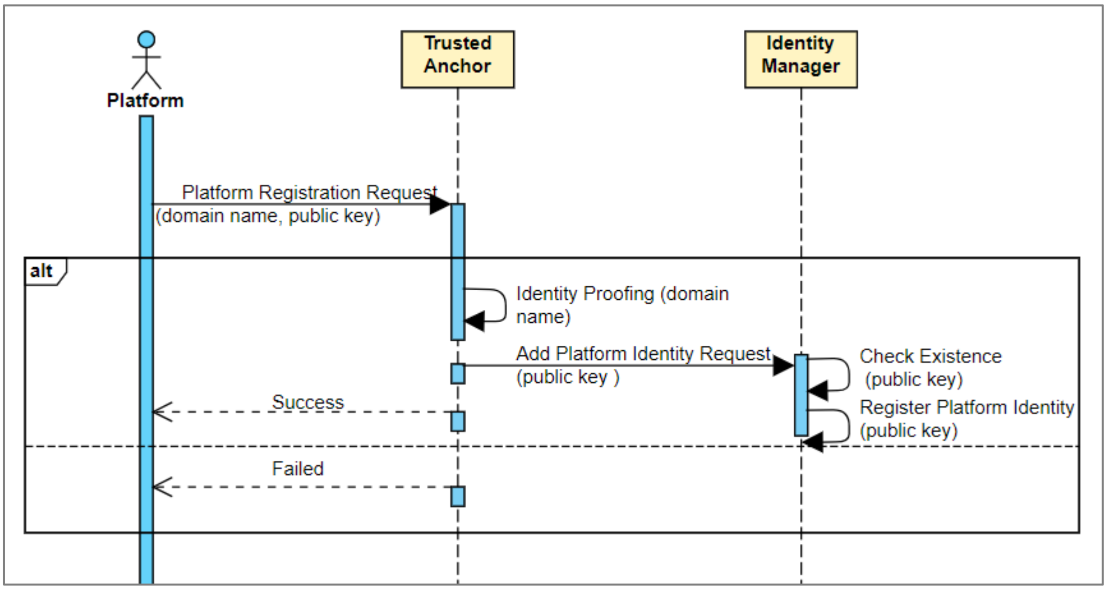

Platform
Trusted Anchor
Identity Manager
Platform Registration Request
(domain name, public key)
alt
Identity Proofing (domain name)
Add Platform Identity Request
(public key)
Check Existence
(public key)
Register Platform Identity
(public key)
Success
Failed

Figure 4. 2. Metaverse Platform Registration Sequence Diagram.

Algorithm 2: Metaverse Platform Registration

```
Input: Public key
Output: Add new identity to the system
Procedure Add Metaverse platform identity.
    1. If identity does not exist (fingerprint value, public key)
    2. Add Metaverse platform to the Blockchain network.
    3. Else
    4. Registration failed.
    5. End if
End Procedure
```

4.2.2. Identity Proofing

Identity proofing is the process of verifying the identity of a real-world entity by collecting and validating personal information [35]. This is typically done through the use of documents such as passports, driver's licenses, or other forms of government-issued identification. In this process, the proposed framework checks if a claimed identity matches his real identity. This is done off-chain by the trusted anchor. However, all identity proofing transactions are transmitted via a secure channel. Figure 4.3. shows identity proofing architecture.

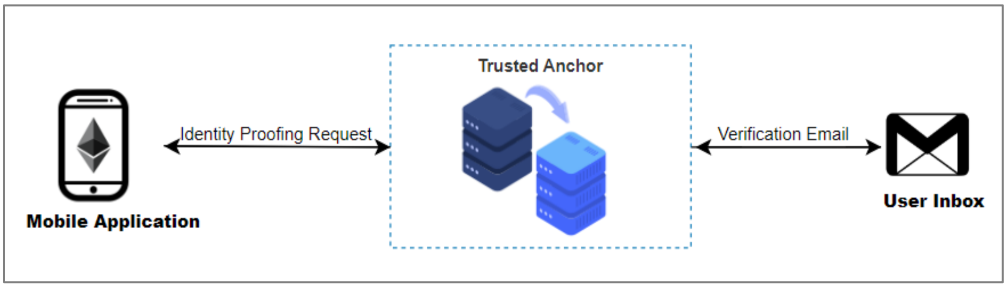


Figure 4. 3. Identity Proofing Architecture.

The identity proofing process is performed for both users and Metaverse platforms to approve adding their identities to the system. The platform provides its domain to the trusted anchor. Then the trusted anchor checks if the domain is already verified in its database or not. On the other hand, the user provides his email address using mobile application. Then the framework communicates with the user's chosen trusted anchor. The trusted anchor first checks if the user is already identified in its database or not. If there is no email address matched with the user-provided email, then the trusted anchor sends a not found response. Otherwise, the trusted anchor will send a verification email to this user to guarantee that he is the actual owner of the email address.

4.2.3. Authentication

Authentication indicates the process of verifying the identity of an entity that requests to gain access to data in the system [40], [41]. The authentication process used in the proposed framework is based on challenge-response rather than a classical password. The password-based authentication is less safe from a security perspective. Passwords are vulnerable to various attacks such as brute force and dictionary attacks. Therefore, using public key cryptography in the authentication process is a much more secure alternative to using passwords. The users will not have to remember any of their passwords.

Public key cryptography or known as asymmetric cryptography generates two mathematically related keys, public key, and private key [42]. The public key must be available for everyone to verify the user's identity. On the other hand, the private key must be kept secret and not be shared anywhere. Because if anyone gets access to the user's private key, then he could steal that user's identity. The plain message could be encrypted using either a public key or a private key. However, the only way to decrypt it is by using the corresponding private key or public key. Public key cryptography provides confidentiality. Also, it is used for digital signatures.

The authentication process for a user and a Metaverse platform is similar. The user has to authenticate himself before adding his attribute to the system. Also, the Metaverse platform has to authenticate itself before accessing the user's attributes. Framework authentication utilizes a cryptographic challenge-response protocol. The transformation of the challenge/response message will be done using an SSH connection to avoid eavesdropping attacks. The authentication operations for each one are described in depth in Algorithms 3 - 4 respectively. The following is showing how the user authentication process is performed:

4.2.3.1. User Authentication

a) The user initiates an SSH connection with the consortium Blockchain network.
b) The user sends an authentication request along with his fingerprint in an encrypted form to the IDMc.
c) The IDMc gets its associated public key from stored identities in the consortium Blockchain network. This ensures that the user's fingerprint is already registered in the system. Then checks if this identity is valid or not.
d) The IDMc will encrypt a challenge message (random number) under the user’s public key and send it back to the user.
e) The user decrypts the challenge using his private key. Then calculates the SHA-256 hash value of the challenge.
f) The user encrypts the hash value using his private key (digitally signs the hash value) and responds to the IDMc.
g) The IDMc also calculates the SHA-256 hash value of the challenge message that is sent to the user. If the response value is matched with the calculated hash value. The authentication process is performed successfully, and now the user can add his attributes to be stored on the chain. Otherwise, the authentication process is failed.

Algorithm 3: User Authentication

Input: Fingerprint value

Output: User is valid to access the system

```
Procedure Verify user (fingerprint)
    1. Send challenge (fingerprint)
    2. If solve challenge correctly
    3. Authentication success
    4. Else
    5. Authentication failed.
    6. End if
End Procedure
```

4.2.3.2. Metaverse Platform Authentication

a) The platform initiates an SSH connection with the consortium Blockchain network.

b) The platform sends a request along with the platform's public key to the IDMc.

c) The IDMc checks if the platform's public key is already valid and registered in the system.

d) The IDMc will encrypt a challenge message (random number) under the platform's public key and send it back to the platform.

e) The platform decrypts the challenge using its private key. Then calculates the SHA-256 hash value of the challenge.

f) The platform encrypts the hash value using its private key (digitally signs the hash value) and responds to the IDMc.

g) The IDMc also calculates the SHA-256 hash value of the challenge message that is sent to the platform. If the response value is matched with the calculated hash value. The authentication process is performed successfully, and now the platform can send an access request to read the user's attributes. Otherwise, the authentication process is failed.

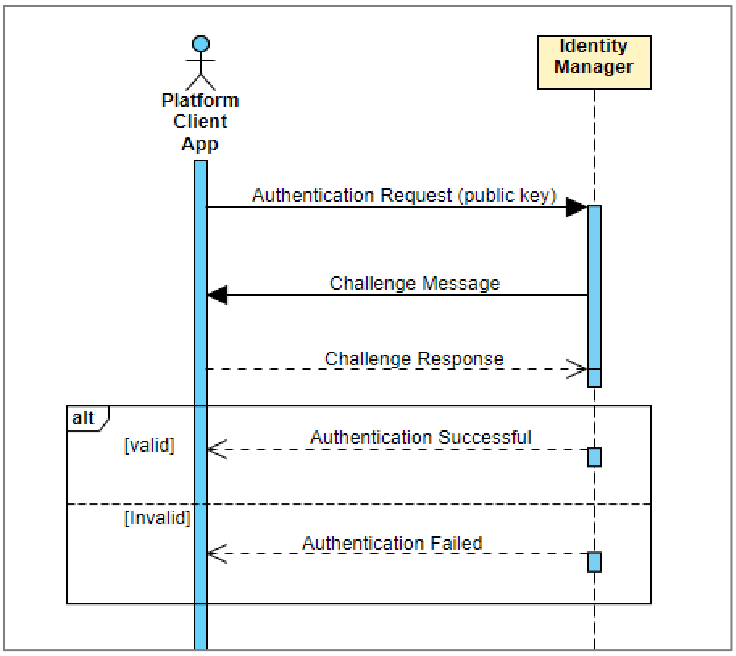


Figure 4. 4. Metaverse Platform Authentication.

Algorithm 4: Metaverse Platform Authentication

Input: Public key

Output: Metaverse platform is valid to access the system

```
Procedure Verify Metaverse platform (public key)
    Send challenge (public key)
    If solve challenge correctly
        Authentication success
    Else
        Authentication failed.
    End if
End Procedure
```

4.2.4. Recording Attributes

Recording attributes is the process of collecting and storing data about an individual [43], such as their name, address, date of birth, and other personal information. This data is used to create a digital profile of the individual, which can be used for identity verification and authentication purposes. After successful login into system, the authenticated user has the ability to store his attribute on the chain. Algorithm 5 depicts the mechanics of this process. Adding new attribute takes the following parameters: user's fingerprint for future identity recovery, attributer key, attribute value, timestep to know when the user adds this attribute. Figure 4.5 shows user's authentication and recording attributes sequence diagram.

a) The user sends a request along with his fingerprint, attribute key and value to IDMc.
b) The IDMc sends a write transaction to the Ethereum network for adding new attribute.

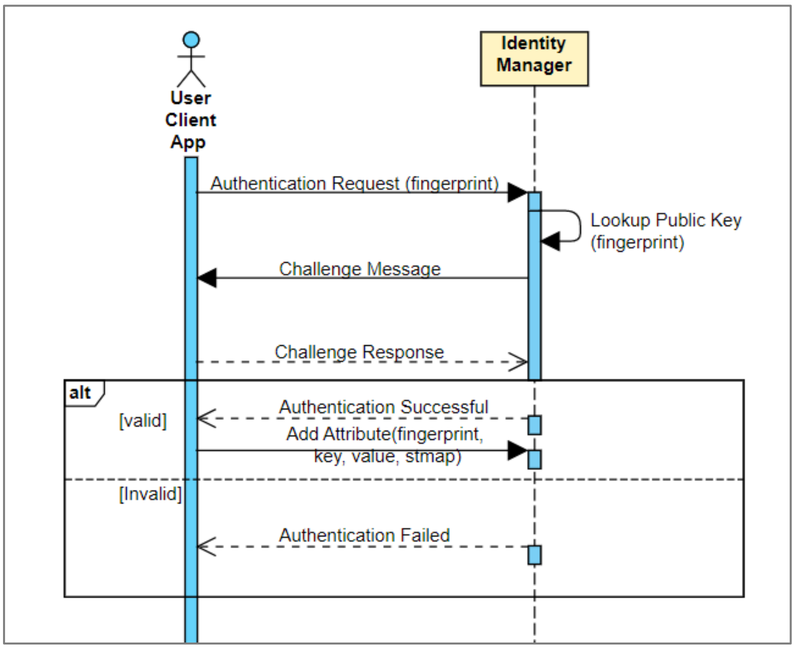

User Client App
Identity Manager
Authentication Request (fingerprint)
Lookup Public Key (fingerprint)
Challenge Message
Challenge Response
alt
[valid]
Authentication Successful
Add Attribute(fingerprint, key, value, stmap)
[Invalid]
Authentication Failed

Figure 4. 5. User Authentication & Recording Attributes Sequence Diagram.

Algorithm 5: Recording Attributes

```
Input: fingerprint value, attribute key, attribute value, timestamp
Output: Add new user attribute
Procedure Add attribute (fingerprint value, attribute key, attribute value, timestamp)
    1. If authenticated user
    2. Add user attribute to the Blockchain network.
    3. Else
    4. Recording attribute failed
    5. End if
End Procedure
```

4.2.5. User Identity Recovery Mechanism

Identity recovery is the process of restoring an individual's identity after it has been compromised or stolen [44]. This can involve restoring access to accounts, recovering lost or stolen documents, and restoring access to personal information. This process preformed when user lost his mobile device, means he is lost his identity. The identity manager will recover his account and all his previously stored attribute. The steps for the identity recovery process are shown in Figure 4.6 and discussed in Algorithm 6.

a) The user requested to recover his account through the mobile application with his fingerprint.

b) New asymmetric keys will be generated for the user. The private key will be stored locally. As a result, the new identity connects to the new device. Then he sends a request along with the user’s fingerprint value and the generated public key to the IDMc for recovery.

c) The IDMc will make his old identity not valid. Then it will create a new identity for the user by storing his new public key and fingerprint value in the Blockchain network.

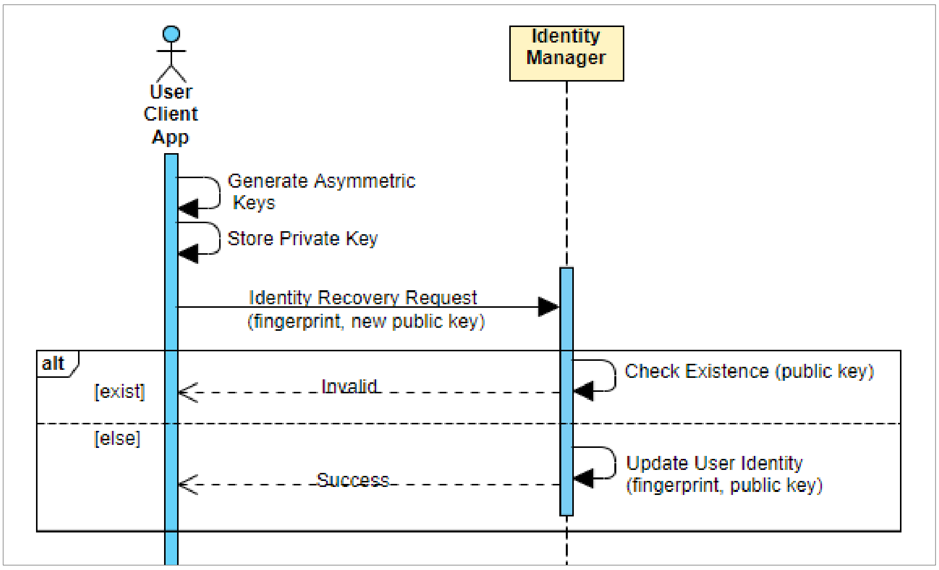

User Client App
Identity Manager
Generate Asymmetric Keys
Store Private Key
Identity Recovery Request (fingerprint, new public key)
alt
[exist]
Invalid
Check Existence (public key)
[else]
Success
Update User Identity (fingerprint, public key)

Figure 4. 6. User Identity Recovery Sequence Diagram.

Algorithm 6: User Identity Recovery

Input: Fingerprint value, new public key

Output: Recovered user identity

```
Procedure Recover user identity.
    1. Update user identity (fingerprint, public key)
    2. If new user identity does not exist (fingerprint value, public key)
    3. Add new user identity to the Blockchain network.
    4. Else
    5. Recover failed.
    6. End If
End Procedure
```

4.2.6. User Identity Revoke Mechanism

Identity revoke is the process of abolishing an entity from accessing the system [45]. This process is performed by the trusted anchor in case of bad behavior occurs, defrauding, detecting identity theft, etc. The user will not be able to login and participate in the system again. The identity revokes operations are shown in Algorithm 7.

a) The trusted anchor requested to revoke an identity from a specific user that will send a request along with the user's identity to IDMc
b) The IDMc lookup the identity and make it no longer valid.

Algorithm 7: Revoke User Identity

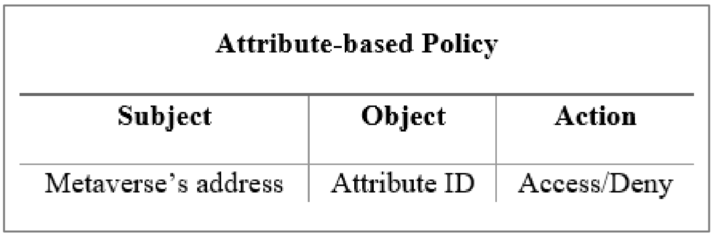
Input: Fingerprint value, public key

Output: Revoked user identity

```
Procedure Revoke user identity
Revoke user identity (fingerprint, public key)
End Procedure
```

4.3. ACCESS CONTROL SCHEME

The proposed framework allows Metaverse users to enforce their attribute-based AC policies, enabling users to control what of their attributes want to share with which Metaverse platform. After registering and authenticated successfully, the user has the ability to record his AC behavior on the Blockchain. By default, the framework will deny all Metaverse platforms from accessing the user's attributes unless the user adds an access policy to allow a specific platform to read his attributes. The policy contains a platform address and attributes list with their specified actions, as shown in Figure 4.7.



Attribute-based Policy

Subject	Object	Action
Metaverse's address	Attribute ID	Access/Deny

Figure 4. 7. Access Control Policy.

Following steps shows how user can add an AC policy:

a. The user sends a request to the Access Manager contract (ACMc) along with registered Metaverse platform address, a list of attributes, and an access flag that indicates if this platform is allowed or denied.

b. The ACMc sends a write transaction to Ethereum network for adding new AC policy.

Algorithm 8: Add AC policy

```
Input: Attribute Id, Metaverse platform address, permission flag
Output: Add AC policy to the system
Procedure Add AC policy (Attribute Id, platform address, permission)
Add AC policy to the Blockchain network.
End Procedure
```

After storing the AC policy, a registered platform can access user attributes as appeared in Figure 4.8.

a. The Metaverse platform sends an access request to the user's attributes using mobile application.

b. The ACMc handles the request to assess platform eligibility to access this attribute based on the user's AC.

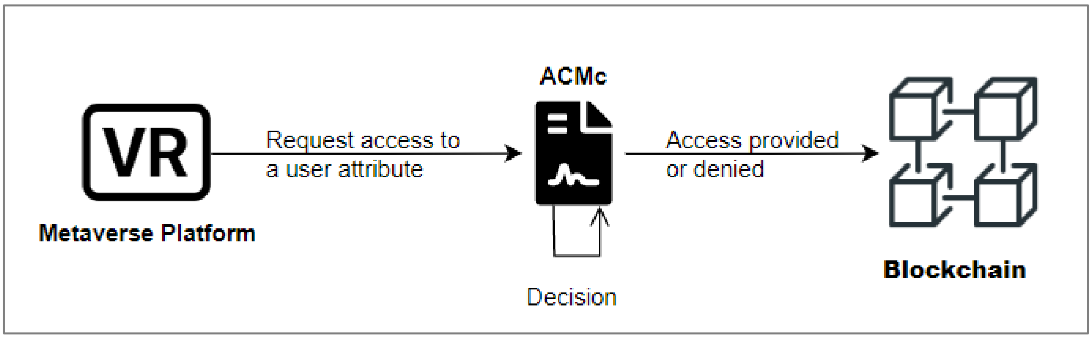


Figure 4. 8. AC Management.

4.4. HARDWARE AND SOFTWARE

The proposed framework has been implemented in a simulation environment. In this section, the used hardware and software used to build the proposed framework will be presented.

4.4.1. Used Hardware

Mobile device and a laptop with Windows 11 Home operating system with 8GB RAM, 11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz 2.42 GHz specifications has been used for the implementation of the proposed framework.

4.4.2. Used Software

The framework is powered by the Consortium Ethereum Blockchain. In order to build the framework, Ethereum dependencies needed to be used. The dependencies are as follows:

- Node Package Manager (NPM) to install packages.
- Solidity is a programming language which is like other object-oriented languages [40], it is designed to target the Ethereum Virtual Machine (EVM), and it is used for writing smart contracts on the Ethereum Blockchain.
- Truffle Framework for creating and deploying the smart contracts. A framework provides a suite of tools for compilation, linking, migrations, and deploying Ethereum smart contracts to any Blockchain networks. It is the environment for DApp based on Blockchain and Ethereum technology. Truffle is a ready environment with all the requirements to create DApp applications so that it gives the possibility to collect smart contracts, test, publish and launch them in the program. Also, allows the work of interfaces [41].
- Geth to setup Ethereum network. Which is Go Ethereum which is an Ethereum software client implemented in Google's Go programming language. Geth is used to turns a computer into an Ethereum node that runs on P2P network [46].

- Web3 is a set of JavaScript libraries that allow interaction with a local or remote Ethereum node using Hyper Text Transfer Protocol (HTTP), Inter-process communication (IPC) connection, or WebSocket. The web3 JavaScript library enables interaction with the Ethereum Blockchain network. Mainly used to interact with smart contracts and send read or write transactions [47].
- Remix which is an online Ethereum editor for testing the smart contracts.

The software used to simulate the trusted anchor network:

- MongoDB Database: that represents the trusted anchor database.
- Nodemailer: Node Js module designed for sending emails.
- Node Js: used to run JavaScript code on the server.
- Express Js REST API to handle requests.
- Postman: use to test API requests.

The software used to develop a mobile application are:

- React Native used for building mobile application. Which is an Open-source JavaScript framework created by Facebook, designed to develop mobile applications using native UI elements on multi-platform such as iOS, android, and web applications [37].
- Visual Studio Code: Code editor
- Node Js: web server for running React native mobile application.

4.5. FRAMEWORK IMPLEMENTATION

This section explains the environment setup that is used to build the Blockchain-based IdM and AC framework. The section includes three main parts: Ethereum network implementation, trusted anchor network implementation, and mobile application implementation.

4.5.1. Ethereum Network Implementation

The proposed framework is implemented in a local Ethereum Blockchain network using clique as a POA consensus algorithm.

4.5.1.1. Set up Ethereum Network

A consortium Ethereum Blockchain network is used to implement the IdM and AC framework. All network nodes are built using Geth. Nodes are assigned an Ethereum account defined by a private and public key. The assigned account used to interact with Ethereum network by sending transactions. All nodes run a Geth client and are connected by their eNode addresses. The P2P network consists of the following node:

- One boot node that listens on port 30301, used to connect all nodes together.
- One miner node that used to mine a new block and broadcast it to the peer nodes. The time used to create a new block has been determined in the network configuration. This time will affect how long users will wait for every write transaction like adding new attribute.
- Two non-miner nodes expose JavaScript Object Notation Remote Procedure Call (JSON-RPC) Application Programming Interface (API) which works over HTTP endpoint. Expose ports 8545 - 8546 for each node respectively. These nodes used as host machine to allow external interaction with this Ethereum network.

After creating the network node, A genesis configuration file was created to initialize the genesis block which is the first block in the network. The genesis file defines the initial behavior and stores all network information. the genesis file parameters were configured to build the Clique network which is PoA consensus in Ethereum.

4.5.1.2. Write Smart Contracts

Smart Contract is a computer code that determines the policy of decentralizing applications based on Blockchain technology. the smart contract will automatically execute when specified conditions are met [48]. It is necessary before deploying the smart contract on Blockchain to test it and ensure it does not have any bugs that can be exploited because the is no way to rewrite the underlying code once it is deployed into the Blockchain network. The aim of smart contracts is to facilitate negotiation and conduct business requirements away from the need for a third party. Moreover, it aims to provide security that is superior to traditional contract law and to reduce other transaction costs associated with contracting [49].

The Truffle framework sets up three directories. The first is called contracts directory which contains all the contracts files. The second is the migrations directory that is used to deploy the smart contract. The last directory is the Test directory which is used to test the functionality of the smart contract. The proposed framework consists of two main smart contracts: identity manger, and access manager smart contracts. The smart contracts files have been created in the contracts directory. These contracts are written in Solidity programming language. Tables 4.1 - 4.2 presents functions details in the main smart contracts respectively.

Table 4. 1. Functions description in identity manager smart contract.

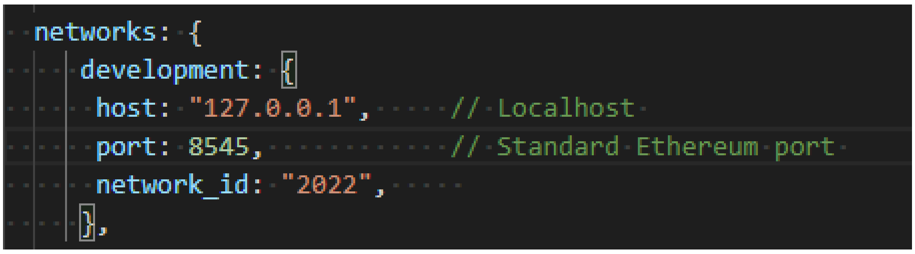
Function Name	**Parameter/s**	**Explanation**
addUserIdentity	name: username and it should be unique. fingerprint: user fingerprint value for future identity recovery. public_key: user public key address.	Add new identity to the system by keeping an array of all registered identities. There is a modifier is activated automatically on this function that is used to avoid reregistering same identity again to the system.
addPlatformIdentity	name: platform name and it should be unique. public_key: platform public key address.	Add new identity to the system.
recoverUserIdentity	new_public_key: new user public key address. fingerprint: user fingerprint value.	Recover user identity.
RevokeUserIdentity	fingerprint: user fingerprint value.	Revoke identity form user.
getUserPublicKey	fingerprint: user fingerprint value.	Get user public key to perform user authentication.
validPlatform	public_key: platform public key address.	Checks if the platform public key is registered before to platform authentication.
getAllUsers	None	Retrieve all registered users.
getAllPlatforms	None	Retrieve all registered platforms.

Table 4. 2. Functions description in access manager smart contract.

Function name	Parameter/s	Explanation
addAttribute	fingerprint: user fingerprint value. attribute_key: attribute label. attribute_value: actual value. stamp: timestamp that the user adds this attribute.	Add attribute to the user identity
getUserAttributes	fingerprint: user fingerprint value.	Retrieve all user attributes.
UserattributesCount	fingerprint: user fingerprint value.	Get the number of added attributes.
getAttribute	attributeID: a unique identifier for the attribute.	Retrieve the actual value of the attribute.
AddAccessPolicy	attributeID: a unique identifier for the attribute. platform: public key of registered platform permission: flag dedicate if access is allowed or denied	Add an access policy to a specific platform to read user attribute.
getPlatformPermission	attributeID: a unique identifier for the attribute platform_ public _ key: public key of platform identity	Check permission of a specific platform on this attribute.

4.5.1.3. Deploy the Smart Contract

In order to deploy this contract onto a local Blockchain network using Truffle, it’s needed to add Ethereum network port into the Truffle configurations. The port number in the Truffle framework should be the same as the port number from the Ethereum private network. Moreover, it’s needed to create a migration file which containing the path of smart contract file. Figure 4.9. shows the Truffle Configuration.

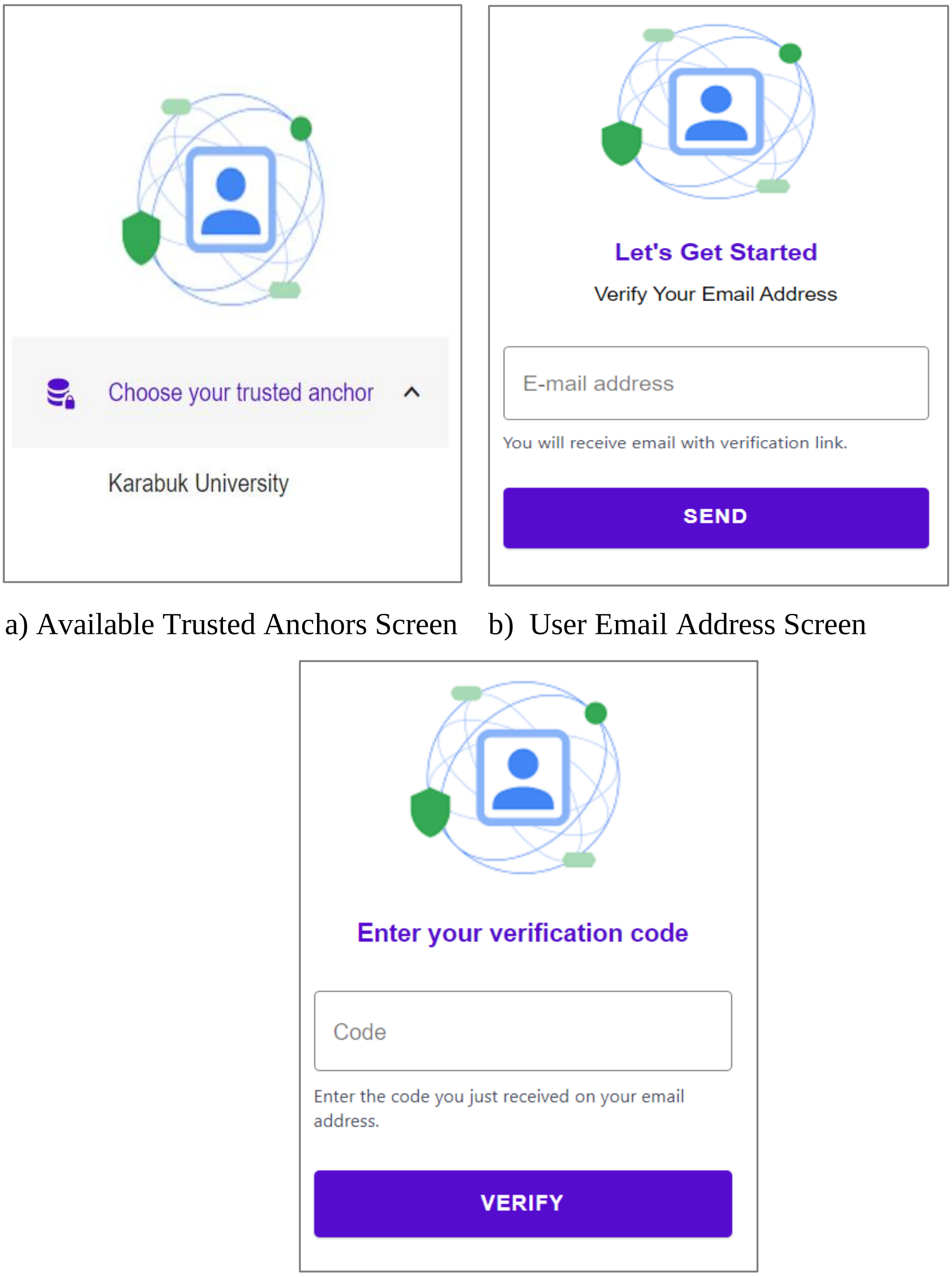


```
  networks: {
    development: {
     host: "127.0.0.1",     // Localhost
     port: 8545,            // Standard Ethereum port
     network_id: "2022",
    },
```

Figure 4. 9. Truffle Configuration.

4.5.2. Trusted Anchor Network Implementation

The overall objective of developing the trusted anchor network is to use it in the identity-proofing and identity revoke services. Every trusted anchor should have a database and mail server. The database contains the trusted users' email addresses. The mail server is used to send a confirmation email to the user's email address. The user should first choose which trusted anchor will perform the identity-proofing process as shows in Figure 4.10, a When the user provides his email address in Finger 4.10, b the trusted anchor checks if this email already exists in its database or not. if it exists then a confirmation email will be sent by the mail server to guarantee that he is own this email address. The sent email contains a verification code which is a random 4-digit number and used once by the user. The user should enter this code into the mobile application as presents in Figure 4.10, c MongoDB is used to simulate the trusted anchor database while nodemailer Node Js module is used for sending confirmation emails. Database API requests are handled via port 3000. The mail server is running on Port 4000 to receive incoming requests from the mobile application.


Choose your trusted anchor
Karabuk University
Let's Get Started
Verify Your Email Address
E-mail address
SEND
Enter your verification code
Code
VERIFY

You will receive email with verification link.

Enter the code you just received on your email address.

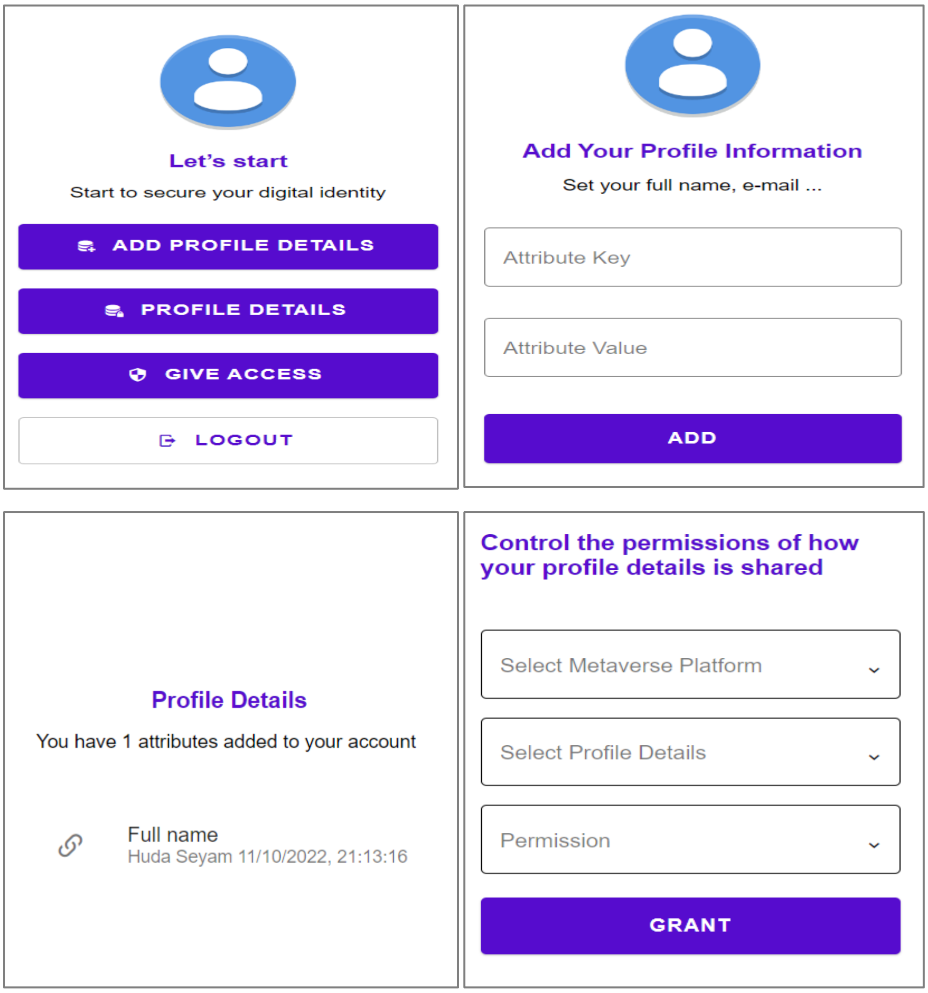
a) Available Trusted Anchors Screen b) User Email Address Screen

c) Verfication Code Screen

Figure 4. 10. Identity Proofing Screens.

4.5.3. Mobile Application Implementation

The proposed framework provides a DApp which is a mobile application that is used as an interface to interact directly with Blockchain. The user and Metaverse platforms interact and send transactions via the mobile application. Figure 4.11. presents the main functionality for the user while Figure 4.12. for the platform. React Native front-end programming language has been used to develop user interfaces, with React Native Paper which gives the pages a great look without writing a lot of CSS code. Node JS server was used to run the react native front-end pages also used for routing which is in simple words: taking a URL and deciding what content should user show. Appendix B shows the web3 mobile application for the framework.



Let's start

Start to secure your digital identity

Add Your Profile Information

Set your full name, e-mail ...

Profile Details

You have 1 attributes added to your account

Full name

Huda Seyam 11/10/2022, 21:13:16

Control the permissions of how your profile details is shared

Figure 4. 11. User Profile Screens.

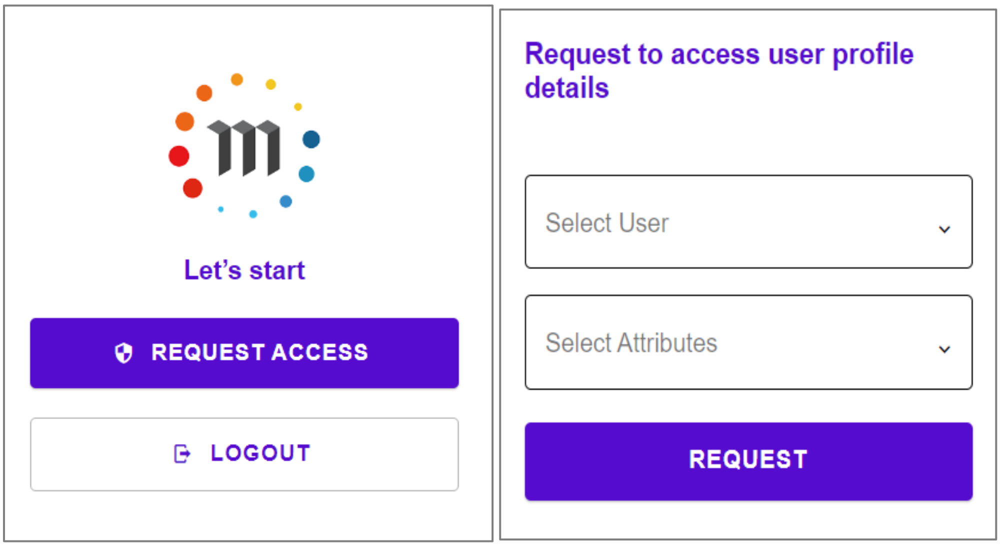


Figure 4. 12. Platform Profile Screens.

4.6. FRAMEWORK EVALUATION

The analysis of the proposed framework will be based on the laws of identity and performance evaluation.

4.6.1. Identity Laws-based Evaluation

laws of identity provide a guideline on how Blockchain should be used in IDMS to ensure user control. Those principles are set by Kim Cameron, Microsoft's Architect of Identity in 2005 [31], [50]. The seven laws are briefly introduced as follows:

1. User Control and Consent: Every identification information related to the user should only be revealed with his consent.
2. Minimal Disclosure for a Constrained Use: IDMS should only collect a minimal amount of user identification information for legal purposes.
3. Justifiable Parties: Any identification information about users should be limited to legitimate parties that have the right to access this information in a transaction.

4. Directed Identity: IDMS should support sharing the user identification information both publicly and secretly.
5. Pluralism of Operators and Technology: IDMS must enable the inter-working and linked-up across identity schemes used by different identity providers.
6. Human Integration: IDMS must ensure that the human user is integrated within the system and understands the implications of his interactions to be protected against identity attacks.
7. Consistent Experience across Contexts: IDMS must provide a simple consistent experience over different security contexts through multiple platforms.

Table 4.3 discusses the evaluation of IdM, and AC framework based on Cameron's laws of identity.

Table 4. 3. Framework Evaluate based on Laws of Identity.

Law	IdM and AC framework
1)	The proposed framework set the control in the hands of the user by enforcing attribute-based access control. Therefore, platforms could not access users' attributes without their consent.
2)	User needs to disclose only one personal data to create an account on the system which is the user's email address that is required to conduct identity proofing.
3)	User's attributes are revealed to allowed Metaverse platforms via a secure channel. Trusted anchor knows the identity of relying parties.
4)	Supports unidirectional identifier which intended to that Metaverse platform and no other. But users might broadcast their identifiers out of application.
5)	Supports integration with a consortium of trusted anchors. But the Metaverse platforms could not add attributes to a specific user.
6)	Provides a mobile application and maintains a persistent user's identifier through an identity recovery mechanism. However, users are not familiar

	with cryptographic key management, and also, they are not educated about the implications of storing their identification information on Blockchain.
7)	Provides a consistent user experience through directed user interfaces that simplify the process of using the mobile application.

4.6.2. Performance-based Evaluation

In this section, the proposed framework is analyzed under different performance metrics, including transaction gas cost, gas limit, block period, and throughput.

In order to evaluate the proposed framework, a consortium Blockchain network is set up. The consortium network consists of 3 network node or peers as shown in Figure 4.13. The trusted anchor node that responsible for mining new transactions into the network. Two Metaverse nodes as receiver nodes that hole a copy of the network ledger. The consortium network is tested by varying block period and gas limit and the result block information is extracted from the Geth console.

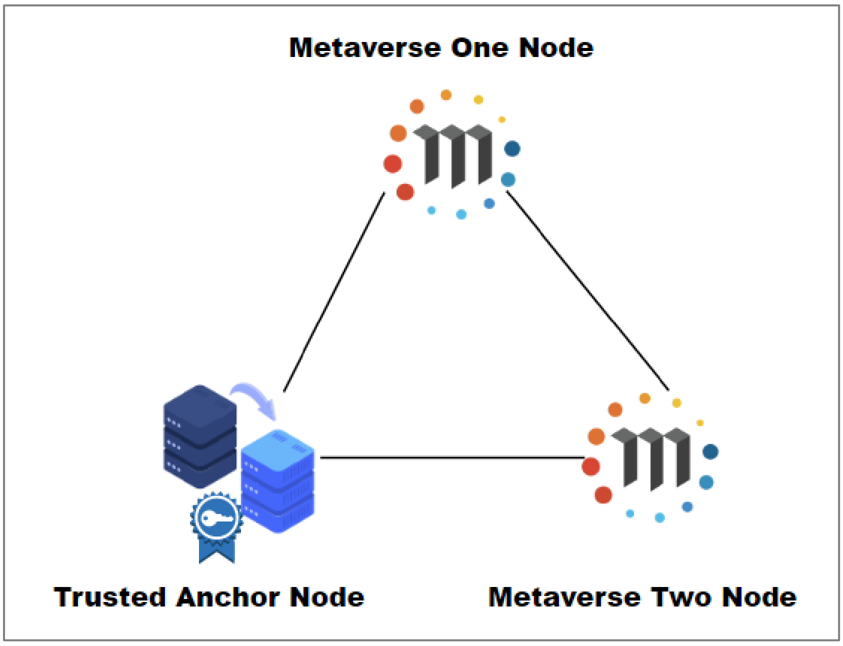


Figure 4. 13. Consortium Blockchain Network.

The proposed framework uses a PoA consensus agreement to deploy the consortium Blockchain. There are important consequences to the choice of consortium Blockchain. The limited number of validator nodes in the consortium network makes the network highly scalable and improves transaction throughput. Moreover, there is no need for expensive mathematical computation to reach an agreement on the state of the ledger. PoA does not require any cost to be paid. Therefore, the user attributes data are stored in the blockchain network instead of storing just the hash value of attributes, and that guarantees the availability and the integrity of data.

4.6.2.1. Transaction Gas Cost

The complexity of the proposed framework smart contracts is shown by computing the transaction gas cost. The deploying cost is required only once when initializing the Blockchain network. If the gas limit of 15 M is the maximum block gas limit and a fee of 1,768,754 wei is required to deploy the identity manager smart contract over the consortium Blockchain network, then one block can have at most eight same-cost smart contracts. On the other hand, the access manager smart contract cost 1,083,074 wei, then one block can contain at most thirteen same-cost smart contracts. Table 4.4 presents the gas used for deploying identity and access manager smart contracts.

Table 4. 4. Gas cost for proposed framework smart contracts.

Smart Contract	Gas Used
Identity Manger	1,768,754
Access Manger	1,083,074

Tables 4.5 - 4.6 present the transaction gas cost of executing the functions of the identity and access manager smart contract. Transaction cost computes the performance of smart contracts in terms of complexity i.e., the higher gas cost, the function takes more time to be executed on the Blockchain. Therefore, recording user attributes is the slowest process in the system.

Table 4. 5. Gas cost for identity manager contract transaction.

Identity Manager Contract Transaction	Transaction Cost
addUserIdentity	134,425
recoverUserIdentity	110,588
addPlatformIdentity	99,451
revokeUserIdentity	38,634

Table 4. 6. Gas cost for access manager contract transaction.

Access Manager Contract Transaction	Transaction Cost
addAttribute	156,158
AddAccessPolicy	107,298

4.6.2.2. Block-Period

Currently, the average block period in public Ethereum network is between 12 to 15 seconds as shown in Figure 4.14 [51]. The setting should be changed by setting the number of seconds should wait before mining the new block. The evaluation process was carried out based on compared of Health-ID since it is the most solution similar to the proposed framework. Health-ID and the proposed framework use consortium Blockchain to provide a secure and distributed IDMS and they both utilize PoA as a consensus agreement for transaction validation and creation. Health-ID is proposed to serve patients and remote healthcare providers. However, the proposed framework serves Metaverse users. In order to compare our block period with those of Health-ID, several experiments were done under different block periods of 3, 5, 10, 15, 20, and 30 seconds. For each experiment, 1500 write transactions were committed to the consortium Blockchain.

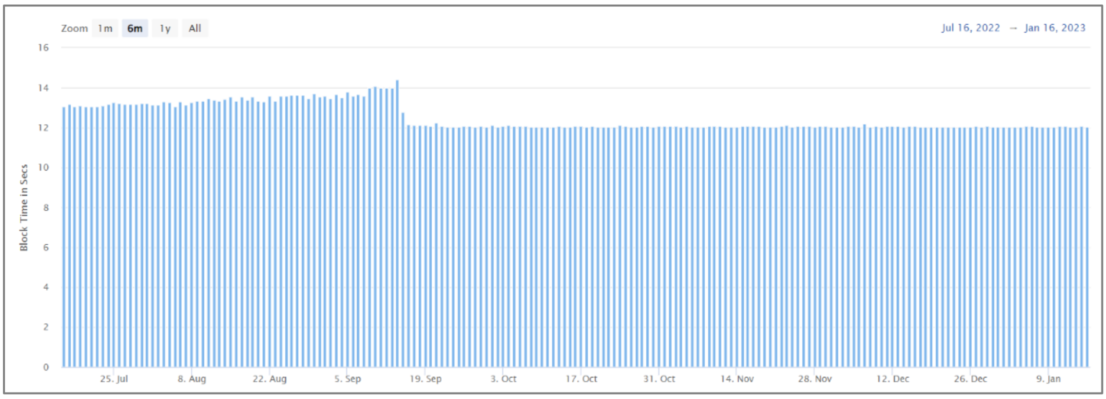


Figure 4. 14. Ethereum Average Block period [51].

Figure 4.15 demonstrate the effect of block-period on transaction throughput. Transactions per second is an indicator that measures the scalability and reliability of the Ethereum network. With higher throughput, the network can be more scalable, and transactions are faster and more secure. The experiments confirmed that the throughput decreases when increasing of block period. Also, the number of transactions per block is decreased with higher block period. The given throughput values in all experiments are greater than the values in Health-ID, i.e., the proposed framework outperforms Health-ID. Also, framework throughput values are around nine times greater than the throughput values in the public Ethereum network as illustrated in Figure 4.16. Although, the gas limit used in the public network is double the target gas limit used in the experiments.

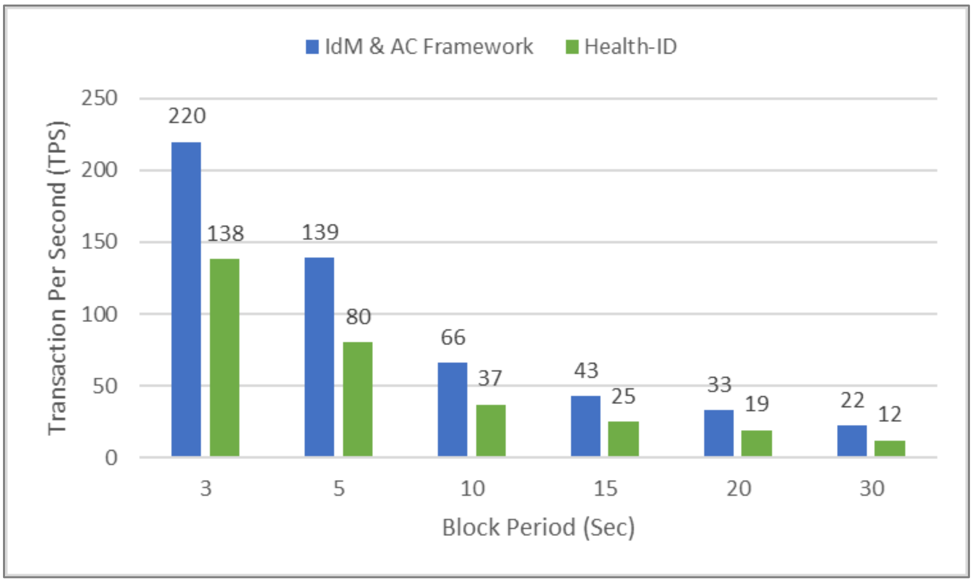


Figure 4. 15. Transactions Per Second Against Block-Period.

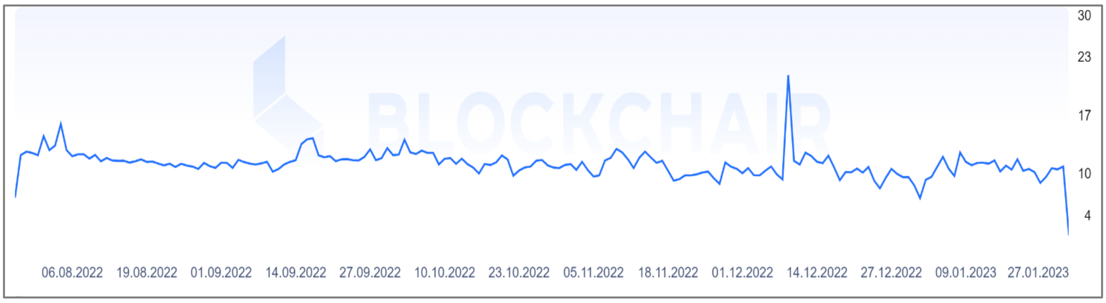


Figure 4. 16. Ethereum Transactions Per Second [52].

4.6.2.3. Gas-Limit

Currently, the average gas limit used in the public Ethereum network is 30,000,000 in Figure 4.17.[53]. However, a higher gas limit can be used to increase network throughput. The proposed framework used a limited number of nodes due to utilizing the consortium network. Thus, the gas limit value can be higher than the public Ethereum network. The gas limit of the experiment was varied from 60M to 200M while the block period is set to 7 seconds. This setting is applied to see the effect of gas limit on network throughput. The gas limit values are the same values used in Health-ID. Also, the block period is the same that is used in Health-ID.

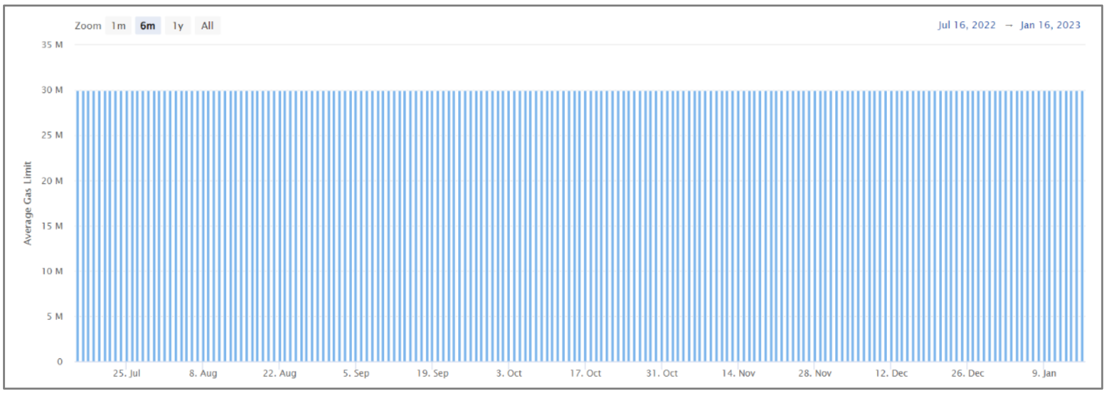


Figure 4. 17. Ethereum Average Gas Limit [53].

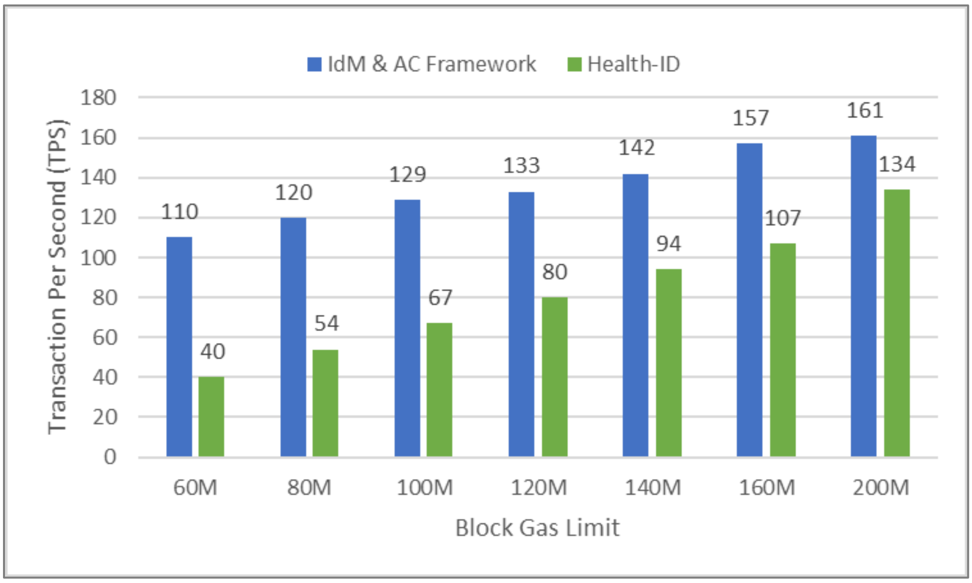


Figure 4. 18. Transactions Per Second Against Gas-Limit.

Figure 4.18 demonstrate the effect of gas-limit on transaction throughput. When using the gas limit of 200M. The throughput is reach more than 160 TPS. Therefore, the experiments confirmed that the number of committed transactions inside the block increases when increasing the gas limit value. Thus, increasing network throughput. The given throughput values in all experiments are greater than the values in Health-ID.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This chapter This part describes the main contributions of this research in the IdM area. Then the future work section explores the enhancements that provide important recommendations for next researchers in the future. The part ends with a conclusion derived from the findings of this study.

5.1. KEY CONTRIBUTIONS

This research reviewed the principles of Metaverse, Blockchain technology, and digital identity, then discussed the main IdM models and highlighted their advantages and drawbacks. Also, the research explored in-depth the recent IDMS that enable decentralized identity: ShoCard, uPort, SCPKI, DNS-IdM, and Health-ID by describing their architecture, components, and their interaction. Additionally, comparative assessments have been described for all discussed IDMS. The assessments were based on the technology used, identity services provided, and security assessment. After the comparative assessments, a decentralized IdM and AC framework for Metaverse users is proposed in an attempt to overcome the identified recent IDMS challenges and meets the user's privacy. The laws of identity are adopted to validate the user's control over his data sharing. The proposed framework facilitates identity services by providing a DApp. Users and Metaverse platforms interact and send transactions via mobile application. The proposed framework is analyzed under different performance metrics, including transaction gas cost, gas limit, block period, and throughput.

A decentralized IdM and AC framework has been proposed. IdM scheme is designed to adopt Blockchain technology to provide identity services. AC is designed to enforce user attribute-based AC policies, enabling users to control. The proposed framework has several advantages over existing Blockchain-based IDMS in the literature:

- Perform identity proofing without depending on physical presence to prove their identity.
- All identity services are provided in the proposed framework. The identity services are registration, authentication, managing users' attributes, identity recovery, and identity revoke.
- Propose an identity new recovery mechanism that recovers users' accounts while preserving their previously added attributes.
- Propose an AC mechanism that enables users to control access policies that specify which of their attributes want to share with any Metaverse platform.

5.2. FUTURE WORK

In the future, we aim to enhance the framework by enabling the Metaverse platform to add attributes for a specific user. In other words, enable addition of two different types of user attributes. The first type is the user's personal information. The second type is the data that is generated while the user used the provided service.

5.3. CONCLUSION

IdM plays a vital role for Metaverse users. IdM and AC framework was proposed that leveraging of Blockchain technology without relying on a central authority or third party for identity verification. The framework enables the Metaverse users to control and manage their own digital identities while preserving data privacy. They can decide what to share, who to share with, and when to stop sharing their personal data, enabling trusted interactions to access users' identification information. Furthermore, enables Metaverse platforms securely identify and authenticate users before offering a customized service to them. The proposed framework was assessed by Cameron's laws of identity, and it was clarified how the framework relates to each law of identity. The performance evaluation has been computed based on transaction gas cost, gas limit, block period, and throughput. In the future, we aim to enable the Metaverse platform to add attributes for a specific user which are generated while the user used the provided service.

5.1. KEY CONTRIBUTIONS

This research reviewed the principles of Metaverse, Blockchain technology, and digital identity, then discussed the main IdM models and highlighted their advantages and drawbacks. Also, the research explored in-depth the recent IDMS that enable decentralized identity: ShoCard, uPort, SCPKI, DNS-IdM, and Health-ID by describing their architecture, components, and their interaction. Additionally, comparative assessments have been described for all discussed IDMS. The assessments were based on the technology used, identity services provided, and security assessment. After the comparative assessments, a decentralized IdM and AC framework for Metaverse users is proposed in an attempt to overcome the identified recent IDMS challenges and meets the user's privacy. The laws of identity are adopted to validate the user's control over his data sharing. The proposed framework facilitates identity services by providing a DApp. Users and Metaverse platforms interact and send transactions via mobile application. The proposed framework is analyzed under different performance metrics, including transaction gas cost, gas limit, block period, and throughput.

A decentralized IdM and AC framework has been proposed. IdM scheme is designed to adopt Blockchain technology to provide identity services. AC is designed to enforce user attribute-based AC policies, enabling users to control. The proposed framework has several advantages over existing Blockchain-based IDMS in the literature:

- Perform identity proofing without depending on physical presence to prove their identity.
- All identity services are provided in the proposed framework. The identity services are registration, authentication, managing users’ attributes, identity recovery, and identity revoke.
- Propose an identity new recovery mechanism that recovers users' accounts while preserving their previously added attributes.
- Propose an AC mechanism that enables users to control access policies that specify which of their attributes want to share with any Metaverse platform.

5.2. FUTURE WORK

In the future, we aim to enhance the framework by enabling the Metaverse platform to add attributes for a specific user. In other words, enable addition of two different types of user attributes. The first type is the user's personal information. The second type is the data that is generated while the user used the provided service.

5.3. CONCLUSION

IdM plays a vital role for Metaverse users. IdM and AC framework was proposed that leveraging of Blockchain technology without relying on a central authority or third party for identity verification. The framework enables the Metaverse users to control and manage their own digital identities while preserving data privacy. They can decide what to share, who to share with, and when to stop sharing their personal data, enabling trusted interactions to access users' identification information. Furthermore, enables Metaverse platforms securely identify and authenticate users before offering a customized service to them. The proposed framework was assessed by Cameron’s laws of identity, and it was clarified how the framework relates to each law of identity. The performance evaluation has been computed based on transaction gas cost, gas limit, block period, and throughput. In the future, we aim to enable the Metaverse platform to add attributes for a specific user which are generated while the user used the provided service.

REFERENCES

[1] M. Shuaib *et al.*, “Self-Sovereign Identity Solution for Blockchain-Based Land Registry System: A Comparison,” *Mobile Information Systems*, vol. 2022. Hindawi Limited, 2022. doi: 10.1155/2022/8930472.

[2] M. A. Bouras, Q. Lu, F. Zhang, Y. Wan, T. Zhang, and H. Ning, “Distributed ledger technology for ehealth identity privacy: State of the art and future perspective,” *Sensors (Switzerland)*, vol. 20, no. 2. MDPI AG, Jan. 02, 2020. doi: 10.3390/s20020483.

[3] Y. K. Dwivedi *et al.*, “Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy,” *Int J Inf Manage*, vol. 66, p. 102542, Oct. 2022, doi: 10.1016/j.ijinfomgt.2022.102542.

[4] T. Rathee and P. Singh, “A systematic literature mapping on secure identity management using blockchain technology,” *Journal of King Saud University - Computer and Information Sciences*. King Saud bin Abdulaziz University, 2021. doi: 10.1016/j.jksuci.2021.03.005.

[5] B. Faber, G. Michelet, N. Weidmann, R. Rao Mukkamala, and R. Vatrapu, *BPDIMS:A Blockchain-based Personal Data and Identity Management System*. [Online]. Available: https://hdl.handle.net/10125/60121

[6] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena, “UPORT: A PLATFORM FOR SELF-SOVEREIGN IDENTITY,” 2016. Accessed: May 26, 2022. [Online]. Available: https://blockchainlab.com/pdf/uPort_whitepaper_DRAFT20161020.pdf

[7] S. Aiman, S. Hassan, A. Habbal, A. Rosli, A. Hanis, and M. Shabli, “Smart Electricity Billing System Using Blockchain Technology”.

[8] V. K. Vemuri, “Blockchain: a practical guide to developing business, law, and technology solutions,” *Journal of Information Technology Case and Application Research*, vol. 20, no. 3–4, pp. 161–163, Oct. 2018, doi: 10.1080/15228053.2019.1588546.

[9] Y. Wang *et al.*, “A Survey on Metaverse: Fundamentals, Security, and Privacy,” Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.02662

[10] T. Keerthana, R. Tejaswini, V. Yamini, and K. Hemapriya, “Integration of Digital Certificate Blockchain and Overall Behavioural Analysis using QR and Smart Contract,” *International Journal of Research in Engineering*, vol. 2, no. 3, 2019.

[11] R. Al-Amri, N. H. Zakaria, A. Habbal, and S. Hassan, “Cryptocurrency adoption: current stage, opportunities, and open challenges,” *International Journal of Advanced Computer Research*, vol. 9, no. 44, pp. 293–307, Sep. 2019, doi: 10.19101/ijacr.pid43.

[12] M. N. M. Bhutta *et al.*, “A Survey on Blockchain Technology: Evolution, Architecture and Security,” *IEEE Access*, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 61048–61073, 2021. doi: 10.1109/ACCESS.2021.3072849.

[13] H. Sun, X. Wang, and X. Wang, “Application of blockchain technology in online education,” *International Journal of Emerging Technologies in Learning*, vol. 13, no. 10, pp. 252–259, 2018, doi: 10.3991/ijet.v13i10.9455.

[14] J. Pan, Y. Liu, J. Wang, and A. Hester, “Key Enabling Technologies for Secure and Scalable Future Fog-IoT Architecture: A Survey,” Jun. 2018, [Online]. Available: http://arxiv.org/abs/1806.06188

[15] N. Chaudhry and M. Yousaf, “Consensus Algorithms in Blockchain: Comparative Analysis, Challenges and Opportunities,” in *2018 International Conference on Open Source Systems and Technologies (ICOSST)*, 2018.

[16] G. T. Nguyen and K. Kim, “A survey about consensus algorithms used in Blockchain,” *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 101–128, 2018, doi: 10.3745/JIPS.01.0024.

[17] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” in *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*, Institute of Electrical and Electronics Engineers Inc., Sep. 2017, pp. 557–564. doi: 10.1109/BigDataCongress.2017.85.

[18] S. M. Karim, A. Habbal, S. A. Chaudhry, and A. Irshad, “BSDCE-IoV: Blockchain-Based Secure Data Collection and Exchange Scheme for IoV in 5G Environment,” *IEEE Access*, 2023, doi: 10.1109/ACCESS.2023.3265959.

[19] N. A. Askar, A. Habbal, A. H. Mohammed, M. S. Sajat, Z. Yusupov, and D. Kodirov, "Architecture, Protocols, and Applications of the Internet of Medical Things (IoMT)," *Journal of Communications*, vol. 17, no. 11, pp. 900–918, Nov. 2022, doi: 10.12720/jcm.17.11.900-918.

[20] S. Tanwar, K. Parekh, and R. Evans, "Blockchain-based electronic healthcare record system for healthcare 4.0 applications," *Journal of Information Security and Applications*, vol. 50, Feb. 2020, doi: 10.1016/j.jisa.2019.102407.

[21] S. M. Karim, A. Habbal, S. A. Chaudhry, and A. Irshad, "Architecture, Protocols, and Security in IoV: Taxonomy, Analysis, Challenges, and Solutions," *Security and Communication Networks*, vol. 2022. Hindawi Limited, 2022. doi: 10.1155/2022/1131479.

[22] M. J. M. Chowdhury *et al.*, "A comparative analysis of distributed ledger technology platforms," *IEEE Access*, vol. 7, pp. 167930–167943, 2019, doi: 10.1109/ACCESS.2019.2953729.

[23] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. M. Leung, "Decentralized Applications: The Blockchain-Empowered Software System," *IEEE Access*, vol. 6, pp. 53019–53033, Sep. 2018, doi: 10.1109/ACCESS.2018.2870644.

[24] "Dapp Defined - Decentralized Applications (Dapps) | Coursera." https://www.coursera.org/lecture/decentralized-apps-on-blockchain/dapp-defined-1RKam (accessed May 19, 2022).

[25] M. K. Hamza, H. Abubakar, and Y. M. Danlami, "Identity and Access Management System: a Web-Based Approach for an Enterprise," *Path of Science*, vol. 4, no. 11, pp. 2001–2011, Nov. 2018, doi: 10.22178/pos.40-1.

[26] Y. Liu, D. He, M. S. Obaidat, N. Kumar, M. K. Khan, and K. K. Raymond Choo, "Blockchain-based identity management systems: A review," *Journal of Network and Computer Applications*, vol. 166. Academic Press, Sep. 15, 2020. doi: 10.1016/j.jnca.2020.102731.

[27] I. Milenković, O. Latinović, and D. Simić, "Using Kerberos protocol for Single Sign-On in Identity Management Systems," *JITA - Journal of Information Technology and Applications (Banja Luka) - APEIRON*, vol. 5, no. 1, Jun. 2013, doi: 10.7251/jit1301027m.

[28] O. Dib and K. Toumi, “Decentralized identity systems: Architecture, challenges, solutions and future directions,” *Annals of Emerging Technologies in Computing*, vol. 4, no. 5. International Association for Educators and Researchers (IAER), pp. 19–40, 2020. doi: 10.33166/AETIC.2020.05.002.

[29] P. Dunphy and F. A. P. Petitcolas, “A first look at identity management schemes on the blockchain,” *IEEE Secur Priv*, vol. 16, no. 4, pp. 20–29, Jul. 2018, doi: 10.1109/MSP.2018.3111247.

[30] “ShoCard.” https://www.shocard.com/en.html (accessed May 26, 2022).

[31] P. Dunphy and F. A. P. Petitcolas, “A first look at identity management schemes on the blockchain,” *IEEE Secur Priv*, vol. 16, no. 4, pp. 20–29, Jul. 2018, doi: 10.1109/MSP.2018.3111247.

[32] J. A. Kassem, S. Sayeed, H. Marco-Gisbert, Z. Pervez, and K. Dahal, “DNS-IdM: A blockchain identity management system to secure personal data sharing in a network,” *Applied Sciences (Switzerland)*, vol. 9, no. 15, 2019, doi: 10.3390/app9152953.

[33] M. Al-Bassam, “SCPKI: A smart contract-based PKI and identity system,” in *BCC 2017 - Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, co-located with ASIA CCS 2017*, Association for Computing Machinery, Inc, Apr. 2017, pp. 35–40. doi: 10.1145/3055518.3055530.

[34] I. T. Javed, F. Alharbi, B. Bellaj, T. Margaria, N. Crespi, and K. N. Qureshi, “Health-id: A blockchain-based decentralized identity management for remote healthcare,” *Healthcare (Switzerland)*, vol. 9, no. 6, Jun. 2021, doi: 10.3390/healthcare9060712.

[35] M. Shimaoka and N. Sonehara, “Modeling the cost structure of identity proofing,” in *Proceedings - IEEE 38th Annual International Computers, Software and Applications Conference Workshops, COMPSACW 2014*, Institute of Electrical and Electronics Engineers Inc., Sep. 2014, pp. 180–185. doi: 10.1109/COMPSACW.2014.34.

[36] M. Esteves, M. Esteves, A. Abelha, and J. Machado, “A proof of concept of a mobile health application to support professionals in a portuguese nursing home,” *Sensors (Switzerland)*, vol. 19, no. 18, Sep. 2019, doi: 10.3390/s19183951.

[37] C. T. B. Garrocho, C. M. S. Ferreira, C. F. M. da C. Cavalcanti, and R. A. R. Oliveira, “Blockchain-Based Process Control and Monitoring Architecture for Vertical Integration of Industry 4.0,” Jul. 2020, [Online]. Available: http://arxiv.org/abs/2007.05788

[38] “How to build Ethereum Dashboard and to monitor your Ethereum Network Status | by 胡家維 Hu Kenneth | Singapore Blockchain-Dapps | Medium.” https://medium.com/singapore-blockchain-dapps/how-to-build-ethereum-dashboard-and-to-monitor-your-ethereum-network-status-a06c5ac18d15 (accessed Dec. 02, 2022).

[39] H. H. Pajooh, M. A. Rashid, F. Alam, and S. Demidenko, “Experimental Performance Analysis of a Scalable Distributed Hyperledger Fabric for a Large-Scale IoT Testbed,” *Sensors*, vol. 22, no. 13, Jul. 2022, doi: 10.3390/s22134868.

[40] Y.-G. Kim and M.-S. Jun, “A Design of User Authentication System Using QR code Identifying Method,” 2011.

[41] D. Dasgupta, A. Roy, and A. Nag, *Advances in User Authentication*. 2017. [Online]. Available: http://www.springer.com/series/13554

[42] S. Malviya and H. Lohiya, “AN ANALYSIS OF AUTHENTICATION ATTACKS WITH COUNTERMEASURES AND VARIOUS AUTHENTICATION METHODS IN A DISTRIBUTED ENVIRONMENT,” *International Research Journal of Modernization in Engineering Technology and Science*, vol. 4, no. 12, 2022, doi: 10.56726/IRJMETS31960.

[43] X. Yang and W. Li, “A zero-knowledge-proof-based digital identity management scheme in blockchain,” *Comput Secur*, vol. 99, p. 102050, Dec. 2020, doi: 10.1016/J.COSE.2020.102050.

[44] W. Lee, J.-H. Jin, and M.-J. Lee, “A Blockchain-based Identity Management Service Supporting Robust Identity Recovery,” *International Journal of Security Technology for Smart Device*, vol. 4, no. 1, pp. 29–34, Apr. 2017, doi: 10.21742/ijstsd.2017.4.1.04.

[45] M. A. Bouras, Q. Lu, S. Dhelim, and H. Ning, “A lightweight blockchain-based iot identity management approach,” *Future Internet*, vol. 13, no. 2, pp. 1–14, Feb. 2021, doi: 10.3390/fi13020024.

[46] “Getting Started with Geth | Go Ethereum.” https://geth.ethereum.org/docs/getting-started (accessed May 19, 2022).

[47] “web3.js - Ethereum JavaScript API — web3.js 1.0.0 documentation.” https://web3js.readthedocs.io/en/v1.7.3/ (accessed May 19, 2022).

[48] M. HOCAOĞLU and A. HABBAL, “NFT based model to manage educational assets in Metaverse,” *European Journal of Science and Technology*, Oct. 2022, doi: 10.31590/ejosat.1189373.

[49] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F.-Y. Wang, “An Overview of Smart Contract: Architecture, Applications, and Future Trends; An Overview of Smart Contract: Architecture, Applications, and Future Trends,” *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, doi: 10.0/Linux-x86_64.

[50] K. Cameron, “The Laws of Identity,” 2005. [Online]. Available: www.identityblog.com

[51] “Ethereum Average Block Time Chart | Etherscan.” https://etherscan.io/chart/blocktime (accessed Jan. 24, 2023).

[52] “Ethereum transactions per second chart — Blockchair.” https://blockchair.com/ethereum/charts/transactions-per-second (accessed Jan. 27, 2023).

[53] “Ethereum Average Gas Limit Chart | Etherscan.” https://etherscan.io/chart/gaslimit (accessed Jan. 24, 2023).

APPENDIX A.

USER INTERFACE DESIGN

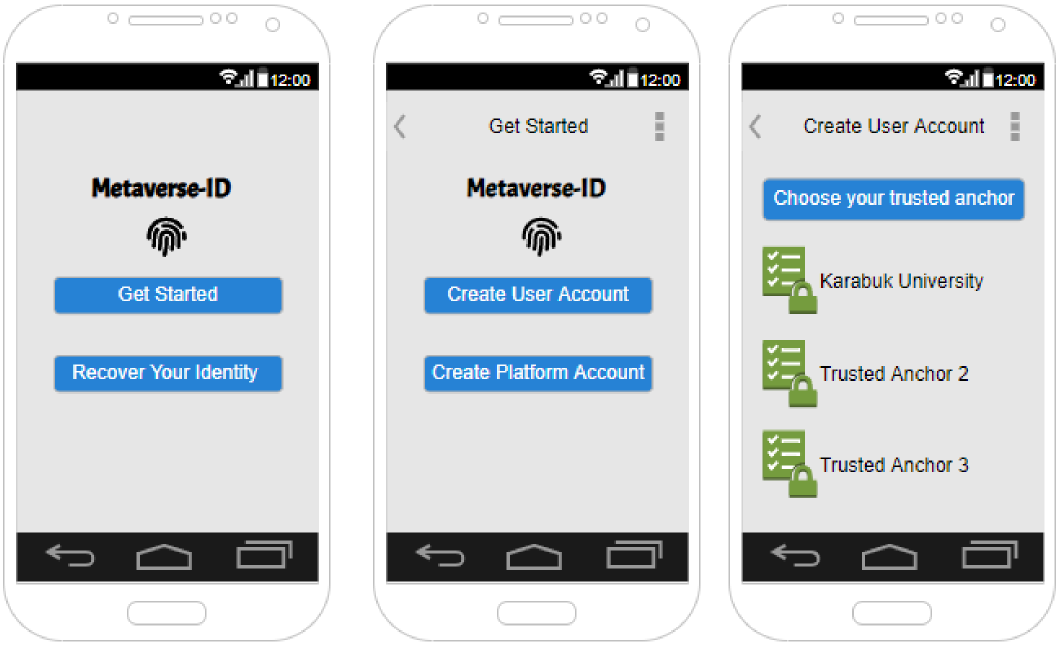
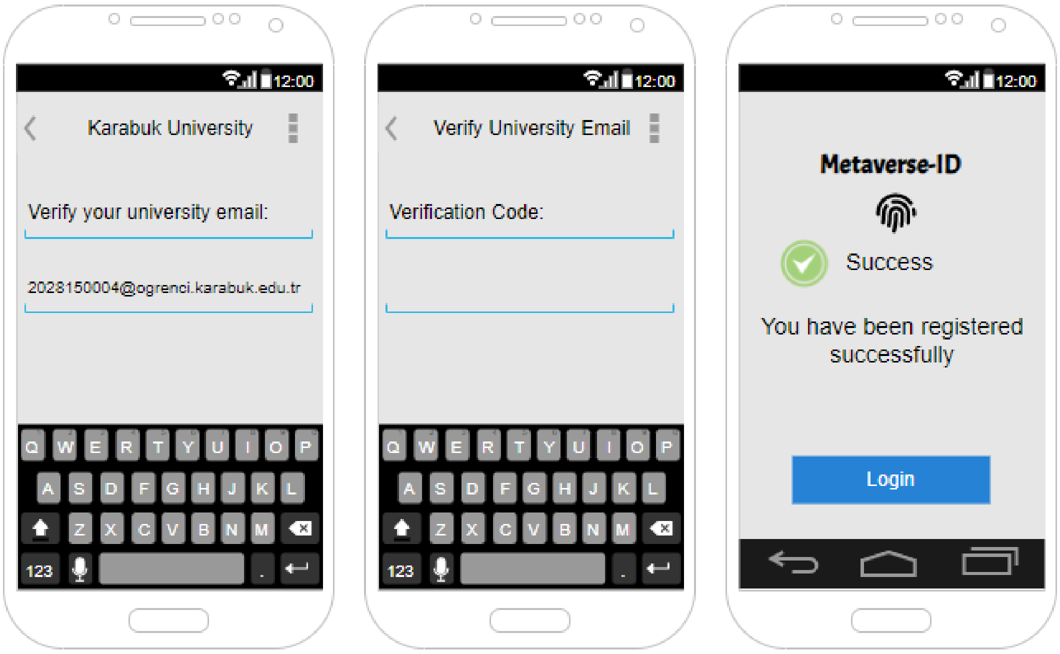


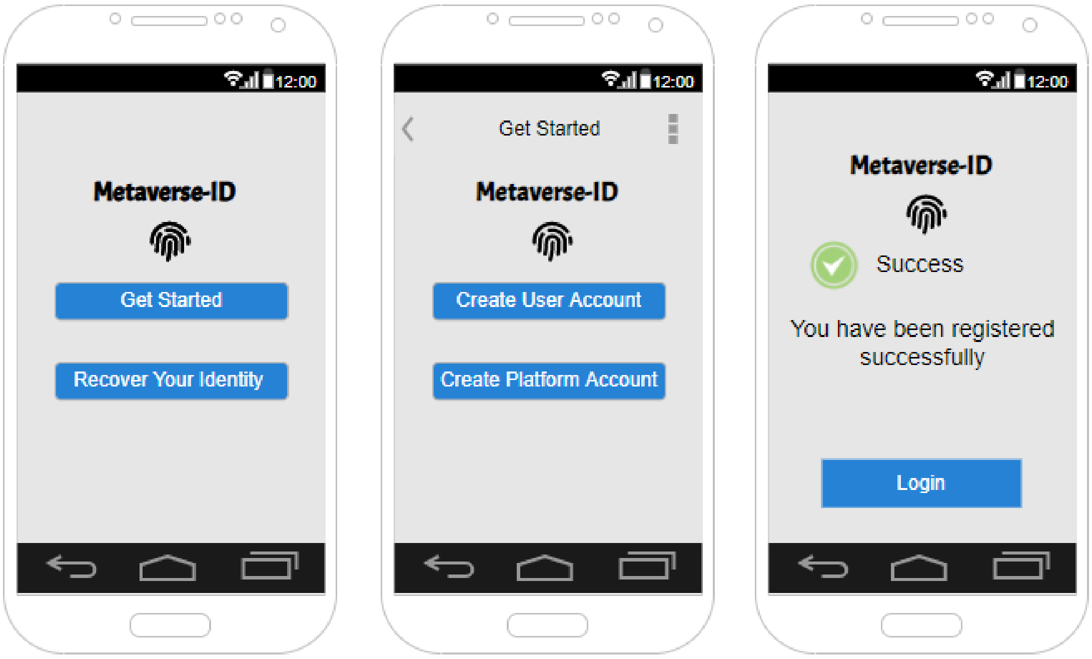
Figure Appendix A.1. User Registration.



2028150004@ogrenci.karabuk.edu.tr

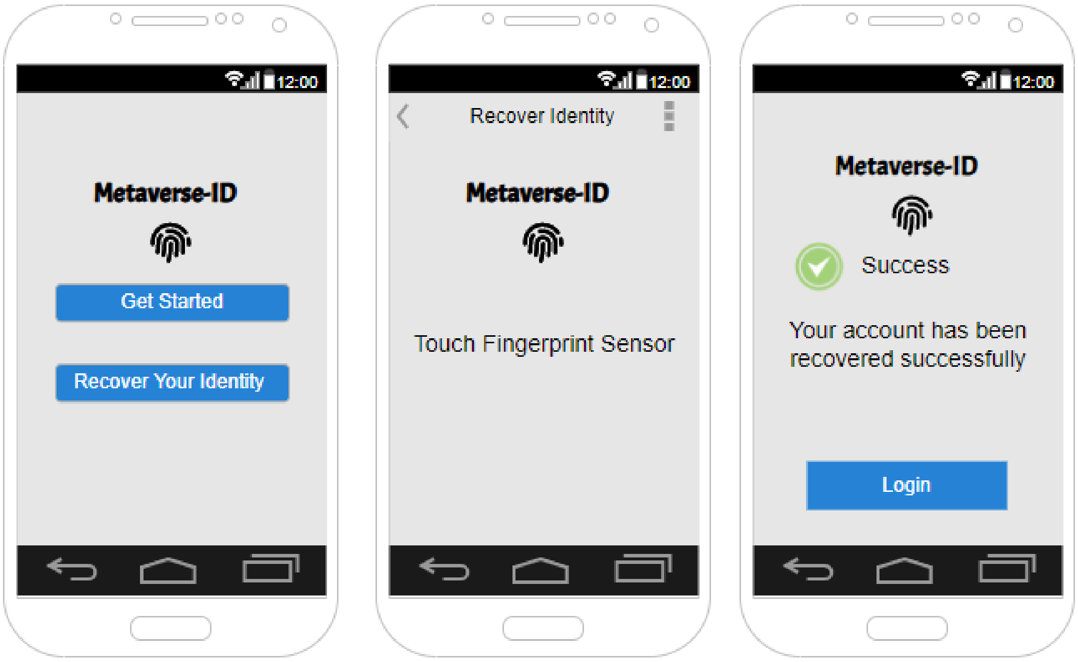
You have been registered successfully

Figure Appendix A.1. User Registration (Continuing).



You have been registered successfully

Figure Appendix A.2. Metaverse Platform Registration.



Touch Fingerprint Sensor

Your account has been recovered successfully

Figure Appendix A.3. User Identity Recovery.

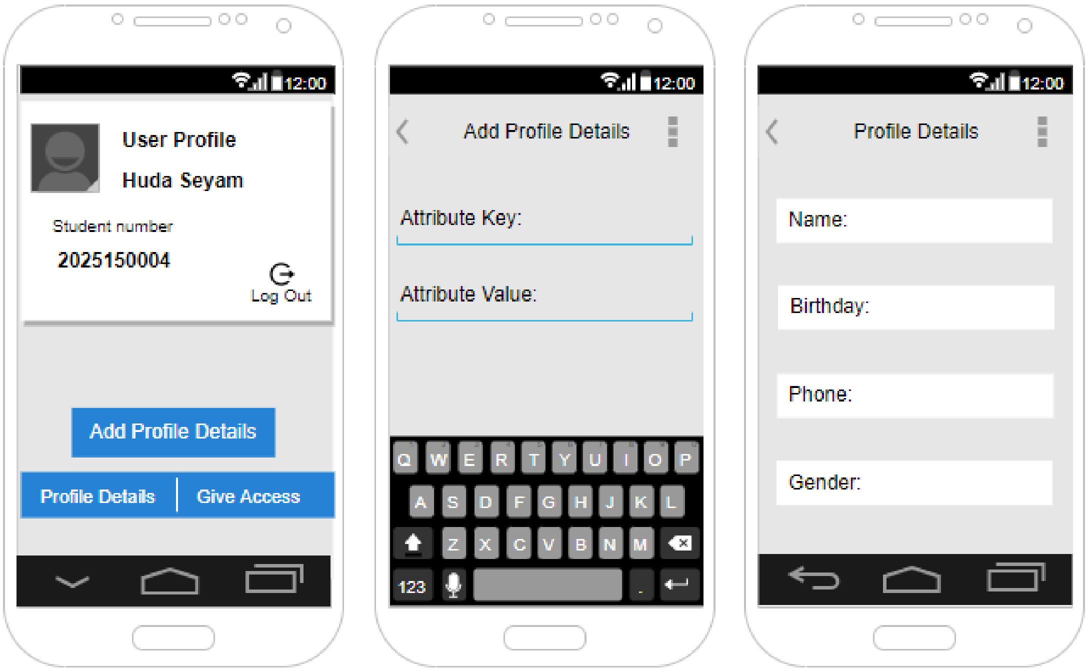


Figure Appendix A.4. Recording User Attribute.

APPENDIX B.

WEB3 MOBILE APP

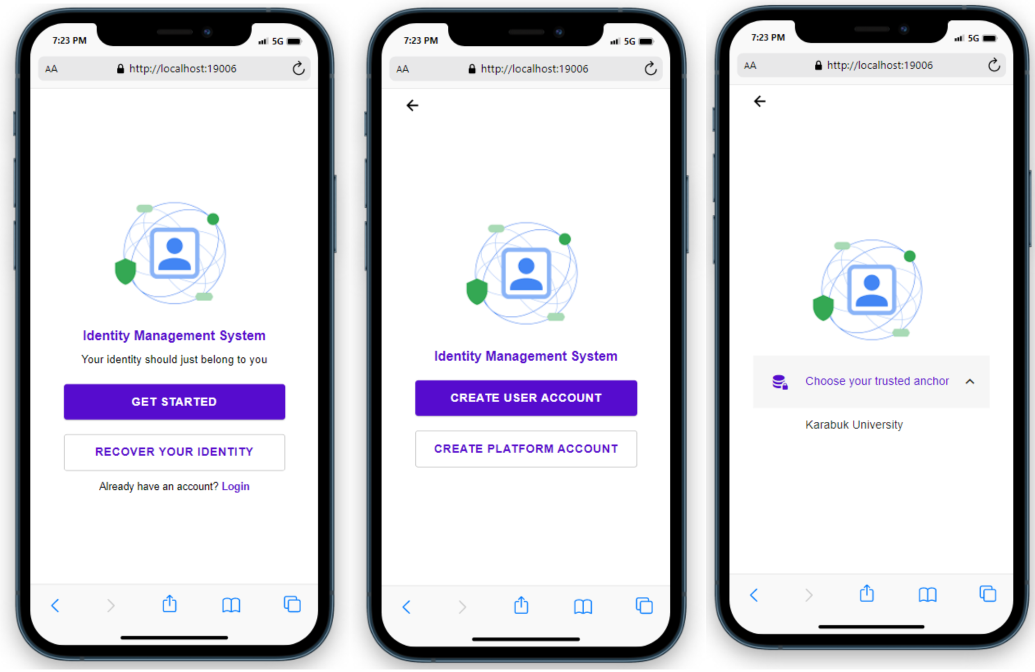


Figure Appendix B.1. User Registration.

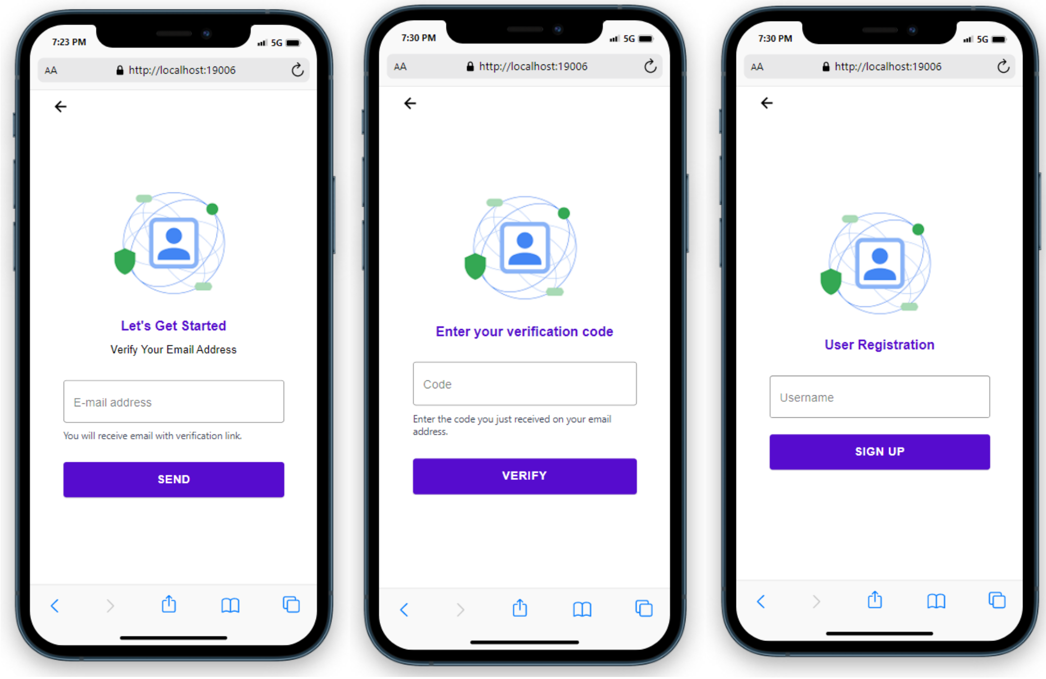


Figure Appendix B.1. User Registration (Continuing).

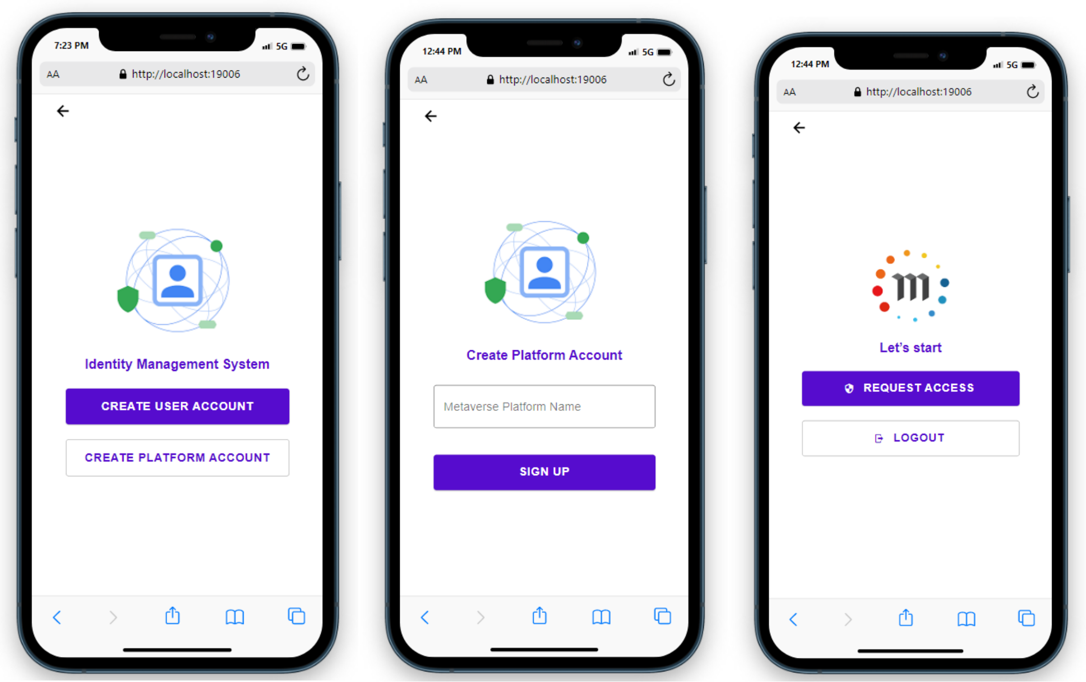
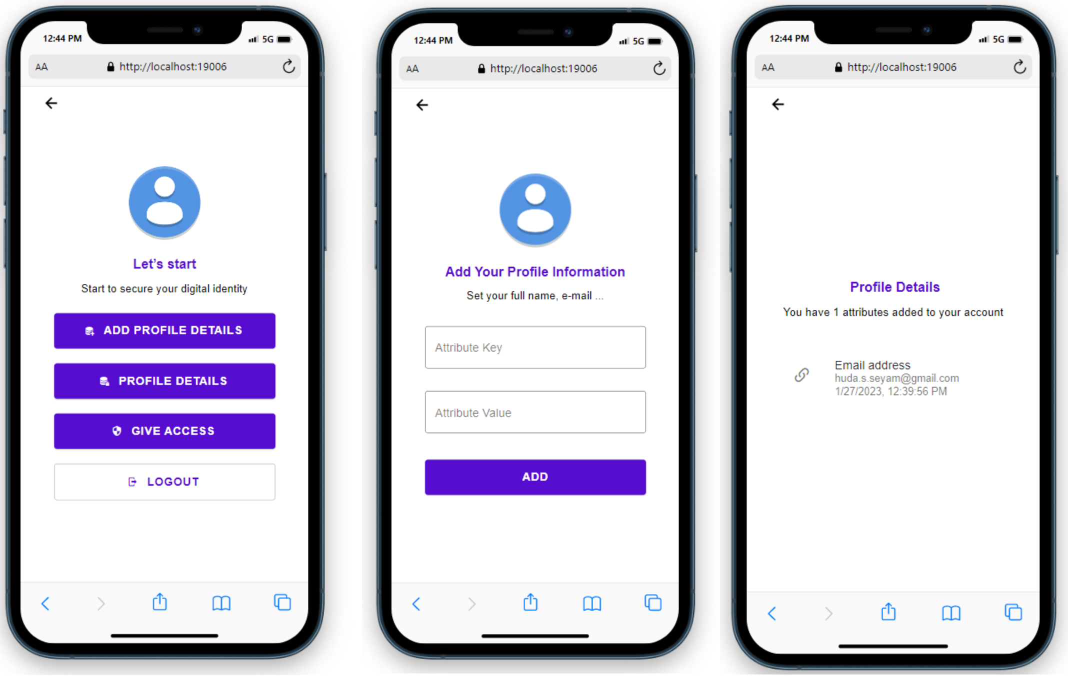


Figure Appendix B.2. Metaverse Platform Registration.



Start to secure your digital identity

Set your full name, e-mail ...

You have 1 attributes added to your account

Email address
huda.s.seyam@gmail.com
1/27/2023, 12:39:56 PM

Figure Appendix B.3. Recording User Attributes.

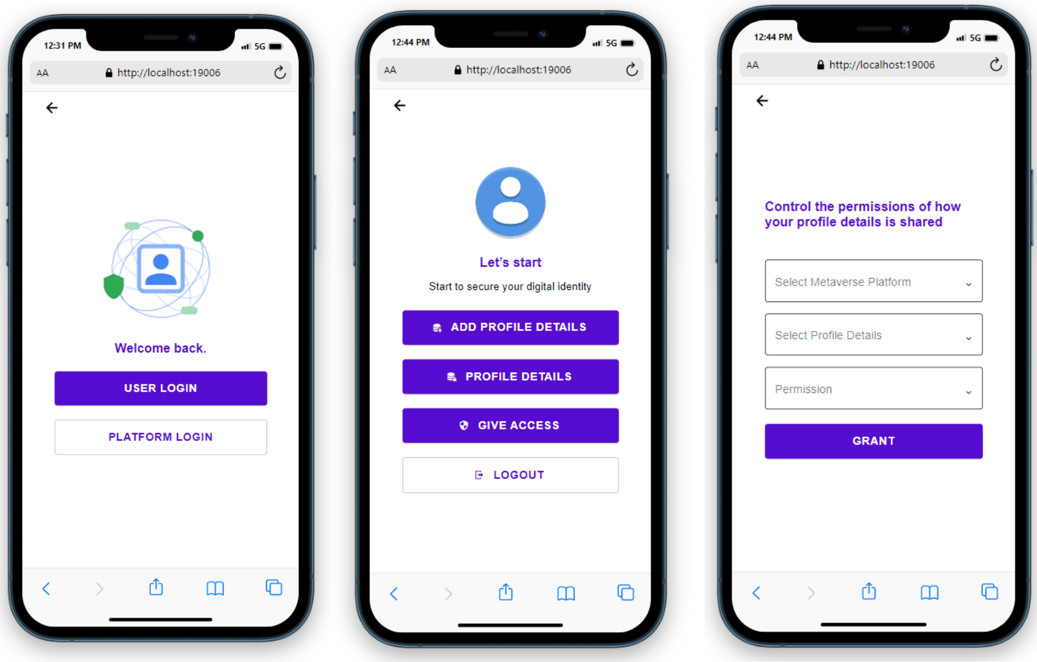


Figure Appendix B.4. User AC Policy.

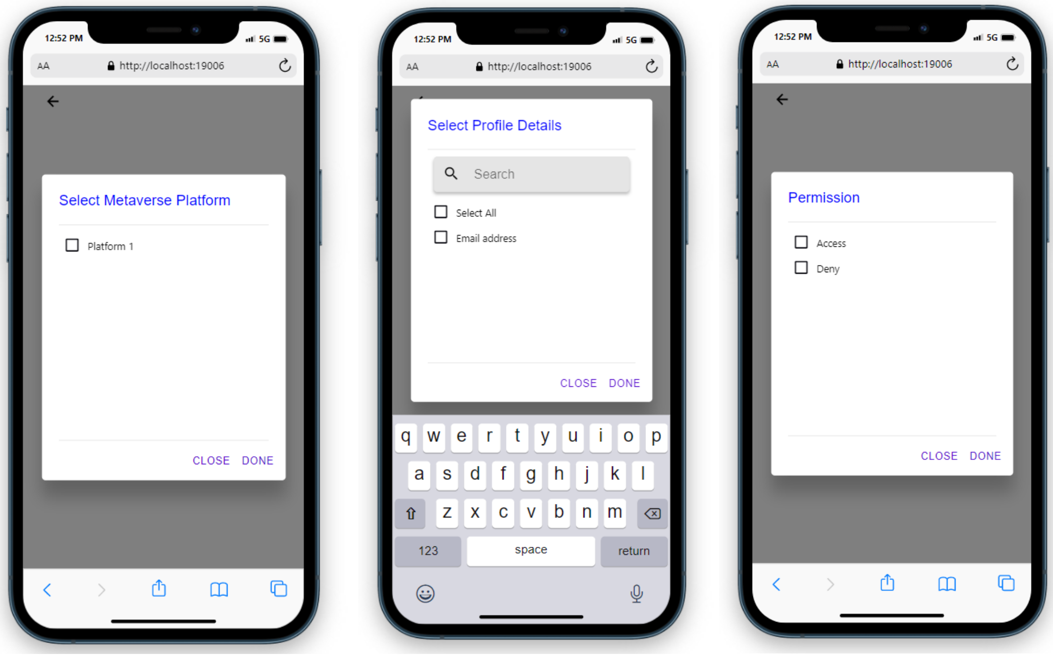
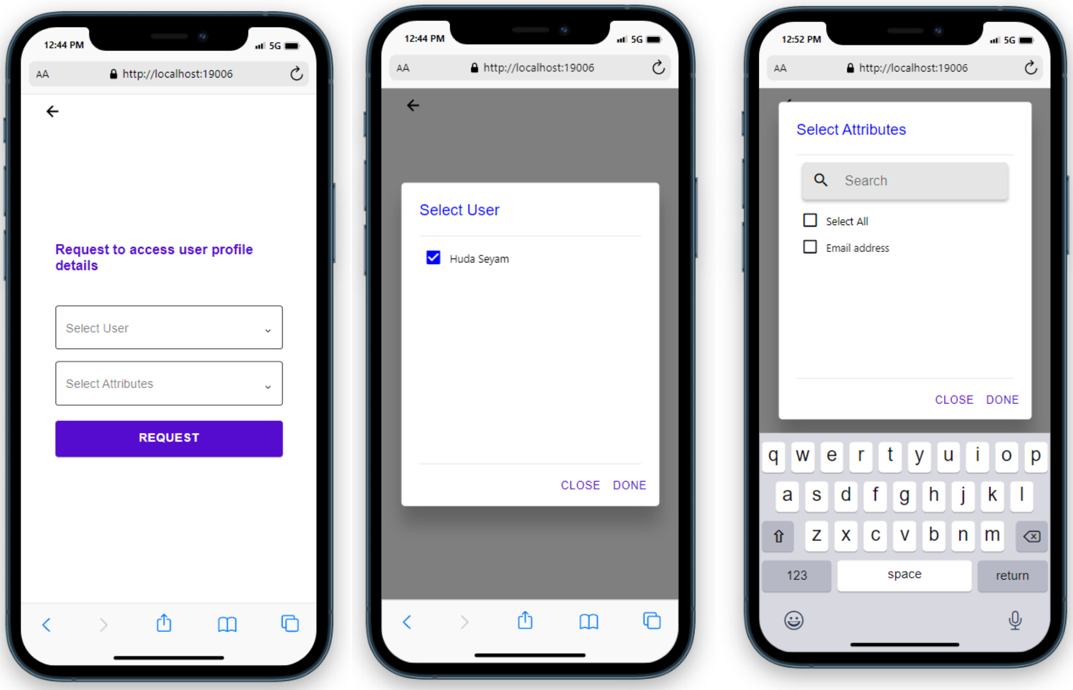


Figure Appendix B.5. User AC Policy (Continuing).



12:44 PM

http://localhost:19006

Request to access user profile details

Select User ⌄

Select Attributes ⌄

REQUEST

Select User

☑ Huda Seyam

CLOSE DONE

12:52 PM

Select Attributes

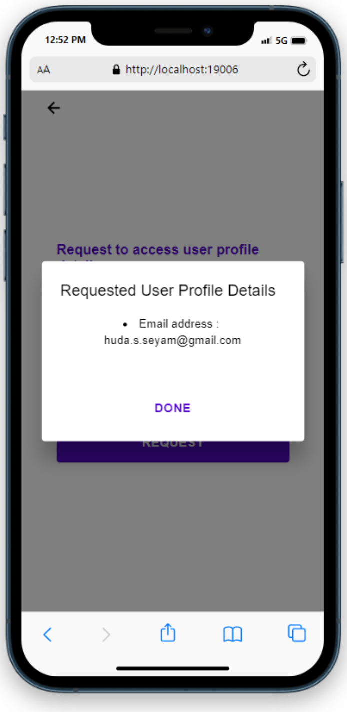
Search

□ Select All

□ Email address

CLOSE DONE

Figure Appendix B.6. Request User Attribute.



12:52 PM

http://localhost:19006

Request to access user profile

Requested User Profile Details

- Email address : huda.s.seyam@gmail.com

DONE

Figure Appendix B.7. Request User Attribute (Continuing).

RESUME

Huda Suhail Seyam finished her elementary education in Gaza, Palestine. She completed her high school education in Arafat secondary school for gifted, after that, in 2015 she started the undergraduate program in Information Security Engineering at the University College of Applied Sciences (UCAS), Department of Engineering and Information System. Then in 2020, she started the graduate program at Karabuk University Department of Computer Engineering to complete her M.Sc. education. Her research interests include Cyber Security, Cryptography, and Blockchain technology.