Academic Document Verification Using SSI and Blockchain Technology

by

Md. Isa Sayek Huda
19301117
Rakibul Anam Rohid
19301004
Fatiha Ishrar Chowdhury
19201015
Sharif Al Musfique
19301239
Md Shohagh Parves
23341110

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

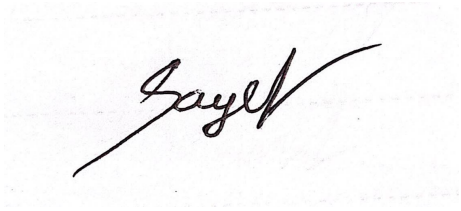
Department of Computer Science and Engineering
School of Data and Sciences
Brac University
September 2023

Declaration

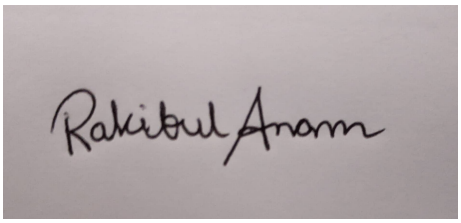
It is hereby declared that

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

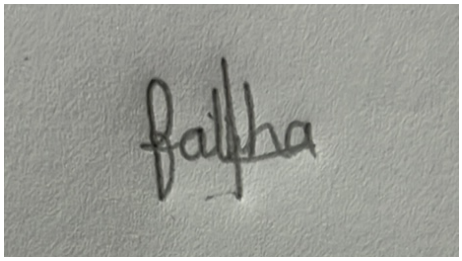
Students Full Name & Signature:



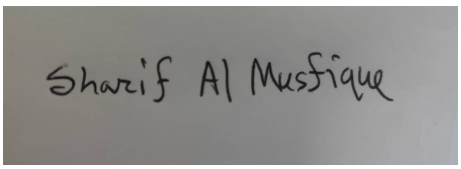
Md. Isa Sayek Huda
19301117



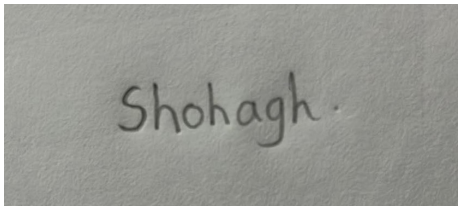
Rakibul Anam Rohid
19301004



Fatiha Ishrar Chowdhury
19201015



Sharif Al Musfique
19301239



Md Shohagh Parves
23341110

Approval

The thesis titled Academic Document Verification Using SSI and Blockchain Technology submitted by

1. Md. Isa Sayek Huda(19301117)
2. Rakibul Anam Rohid(19301004)
3. Fatiha Ishrar Chowdhury(19201015)
4. Sharif Al Musfique(19301239)
5. Md Shohagh Parves(23341110)

of Summer, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September, 2023.

Examining Committee:

Supervisor:
(Member)

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

Co-Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:

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

Head of Department:
(Chair)

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

Abstract

Academic certificates are evidence of qualification for their recipients. Although academic data is regularly transmitted by institutes, the procedure of transferring and verifying this data is usually tiring and time-consuming. This has resulted in certificate verification being unduly complex in the practices that have been traditionally implemented. These credentials are susceptible to manipulation and counterfeiting. Certificate forgery and fabrication are getting more and more difficult to catch as days pass with the customary techniques that are being adopted. More user-friendly and privacy-conscious than previous methods of storing and disseminating ID data, SSI is the latest innovation in this area by offering secure document verification methods and serving as a vital tool against document fraud and abuse. The documentation and verification process for academic certificates may be made safe, seamless, transparent, and immutable by integrating SSI with blockchain technology. Not only physical certificates, as everything is being digitized, online courses are also becoming very popular, but the legitimacy of their certificates can still be questioned. With our system, we try to contribute to SSI core motivation of building a trust layer on the internet. In this work, we offer an SSI system that uses educational institutions to give certificates to their students, who then act as holders of the certificates, keeping them safe in a wallet and being presented whenever needed. The legitimacy of the certificates in the wallet can be verified by the organizations, who are third parties. We plan to construct our system in a way that allows it to accommodate the requirement of selective information disclosure, providing us with increased total control over the verification process. Additionally, this system ensures that there are no intermediary parties which helps us to address the drawbacks of the traditional approaches to an extent. We present its design, go into the specifics of its implementation, point out its advantages and limitations, and most importantly compare it with the methods that are currently in use.

Keywords: SSI, Decentralized ID, Blockchain, Academic Records, Verification, Threat Model, Verifiable Credentials, Identity Management, Selective Disclosure, Asymmetric Encryption, Digital ID.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, many incidents have been found where people involve themselves in falsifying academic records and work. According to a newspaper daily, these kinds of actions have raised so alarmingly that a market has been found for a group of people who can earn an academic certificate for just the price of a pair of shoes. The realization that these issues are prevailing in our region makes us question to what extent morality and integrity are being compromised. Despite the fact that this type of falsification is strongly discouraged and several measures have been implemented to make people aware of the repercussions, these efforts are futile in the face of unethical desires. This malpractice rather shatters the hopes of students who spend weary nights exerting genuine effort to achieve success. After continuous monitoring and implementing several laws, this misconduct showed a slight decrease, but the offender focused on finding loopholes in the current system. In these circumstances, the necessity of implementing a system that can ensure secure verification of documents, is difficult to tamper with, and restricts unauthorized access has become mandatory for the sake of the general public. Many technologically advanced countries have introduced several security methods to ensure the safety of academic records, but there still remains a fault in the system, which makes it vulnerable to forgery in the long run [4]. According to Raman [3], there are a number of reasons why we chose SSI integrated with blockchain as a means to secure academic certificates. According to the Self-Sovereign Identity (SSI) concept, a trustworthy third party is no longer required to authenticate a person's identification online. With decentralized identity management, you own your personal information and select who may access it. The ability to revoke access to your data exists at all times. SSI technology empowers individuals to take charge of their own digital identities, eliminating the need for centralized service providers to handle sensitive information. The terms Self-Sovereign Identity and "decentralized identity" are used interchangeably at the moment. By utilizing DIDs (Decentralized Identifiers), blockchain enables the holder, issuer, and verifier to have access to the same source of truth for verifying which credentials are genuine and who issued the credentials data. It is hardly surprising that SSI has attracted such a wide number of commercial, academic, and government backers. Numerous studies examine the possible applications of SSI in domains such as healthcare [11], [17], the Internet of Things [13], [18], and others. Other studies have looked at the viability of merging SSI with OAuth [7], a common access delegation technique, and SAML [24], a widely used standard for establishing and managing identity federations.

While SSI has numerous potential online applications, little research has been conducted into how it may one day replace ordinary document and certificate authentication. Consequently, SSI's promise as a dependable source of authentication is largely unrealized. Prior to the advent of SSI technology, the traditional approach to certificate verification required a great deal of time for processing, but following the introduction of SSI, this time was drastically reduced. On the other hand, there has been a good amount of research conducted on academic document verification using the blockchain, which addresses a lot of the issues that the traditional paper-based system faces. Ezell [14] asserts that over 5,000 diploma mills operate worldwide and produce over 200,000 bogus degrees annually. Currently, the certificate fabrication industry is worth 1 billion dollars. Blockchain has recently been acknowledged as a revolutionary innovation with the potential to revolutionize the identity verification sector. However, only a few educational institutions employ blockchain technology and the bulk of these institutions use it to validate and distribute academic credentials. The MIT Media Lab Learning has developed Blockcerts, an open standard for applications that generates and verifies certificates based on blockchain technology [35]. In addition, CertChain is a certificate management platform built in British Columbia that authenticates certificates and circumvents the problem of counterfeit certificates. Using the Ethereum BC, OpenCerts [9] can also circumvent the problem of counterfeit certificates. Academic institutions may issue and publish digital certificates on the blockchain using OpenCerts [26]. These suggested methods make use of blockchain technology to run the validation process, thereby offering clarity and security.

1.1 Problem Statement

Academic transcripts and certificates are significant to a person, but the validity of the documents is more essential. Due to several inadequacies in the current system, there is always a possibility that academic references would be tainted. Many universities still verify certificates by hand, but most organizations are moving toward digital verification. Despite the fact that digital validation is making the system seem more effective, there are still challenges.

The reliability of centralized services, database systems, and protocols are questionable, yet, different degrees of trust should be incorporated into the infrastructures activity and communication levels. This is because practically everyone has a central administration system in each environment, whether its a large tech company or a tiny university database. Few colleges are now making an effort to adopt a unique identification for every student in order to track their credentials. Many institutes of higher learning are implementing a central database platform where they can manage the learning management systems components. Most often, they decide to employ a centralized database to streamline operations and stay within their budget. Database management can be done efficiently and economically on a server that is centrally placed. Nevertheless, it can also be broken, once more as a result of a privacy violation. It can be challenging to identify any unlawful adjustments when they are committed in conventional academic transcript storage systems [1].

In [7], it says that by embracing the cutting-edge capabilities of blockchain, education providers would be able to produce official certificates that serve as proof of achievement. Individuals can gather these credentials and give them straight to those who need formal credentials. However, the challenge of scalability that such a blockchain-based system faces renders it impractical for large use cases when we want a system that can run seamlessly on a national or even worldwide scale. Furthermore, privacy issues arise since certificate data on the blockchain cannot be altered or removed, limiting the user's control over their data. However, the expense and resources needed raise concerns about its widespread deployment. Since certificates are issued on a large scale and the verification process must be interoperable between parties; blockchain faces challenges in providing a scalable, modular, and economical solution due to its immutable and expensive nature [23]. All of these issues are addressed by Self-Sovereign Identity, which eliminates the need to store certificates on the blockchain while maintaining the same degree of transparency, security, and decentralization.

1.2 Research Objectives

Academic record transfer and verification are regarded as critical components of both academic and career trajectories. The extraction of academic credentials using the current manual or even digital technique has been found to be particularly vulnerable to fraud or forgery due to its inadequate safety measures. Modern technology makes it much easier to modify the unique identifiers associated with academic papers as the primary form of protection, making the documents more prone to forgery. This fabrication leads to a multitude of unethical activities, such as the wrongful admission of a less qualified student who takes the seat of a deserving talented student.

SSI allows users a decentralized identity and complete control over their identity and individual information. It only shares relevant data with a third party, a practice known as selective disclosure [20]. The primary goal of self-sovereign identity is to issue identification credentials based on a trusted network between two parties.

Users may govern their own identity by adopting Self-Sovereign Identity, which is supported by Blockchain technology. This will aid in achieving the following goals:

- **R01** - Based on SSI (Self-Sovereign Identity) technology, the system prioritizes limiting the information users must disclose in order to avoid needless or excessive disclosure of personal credentials and information. Users can gain complete control over their digital identities and data using SSI, maintaining privacy and security throughout the sharing process.
- **RO2** - Information that is vital for certificate issuance, verification, and sharing is recorded in an immutable and auditable blockchain, ensuring transparency and creating a trustworthy environment for all parties involved.
- **RO3** - The system should allow the issuer to determine who is qualified for a credential and then issue one. The holder shall keep the credentials in their

(digital) wallet and can validate the credential's claims if requested.

- **RO4** - At a later situation, the holder will be required to validate the credential's claims by giving the verifier a verifiable presentation. Holder may also be able to demonstrate that the granted credential has not been revoked.
- **RO5** - This authentication process ensures that only authorized parties can access the blockchain records.
- **RO6** - Developing a Proof of Concept (PoC) in our system using a number of tools and framework that guarantees to provide a secure, private channel with another person or organization but has no central server and requires no login.

1.3 Structure

We briefly examined the problems with the present system and the goals of our research in Chapter 1. The background of SSI technology is discussed in Chapter 2, along with some of its core concepts. Some relevant articles are reviewed in Chapter 3. In Chapter 4, we examined the requirements of our system as well as the threat model. We delve deeply into the macro-level system design of our proposed model in Chapter 5, which also has a full breakdown of how certain use cases are implemented with detailed Protocols Flow. In Chapter 6, we give an analysis of the system requirements with the fulfillment of the study purpose, some comparison with studies in our field, the benefits and limitations, and with potential future scopes. Finally, we emphasized the significance of our method in Chapter 7 to wrap up our paper.

Chapter 2

Background

2.1 Self-Sovereign Identity

Self-Sovereign Identity (SSI) is a revolutionary approach to digital identity that empowers individuals with full control over their personal information. It is built upon decentralized technologies, such as blockchain, to provide a secure and user-centric identity management system. In SSI, individuals can create and manage their digital identities independently, without relying on centralized authorities or intermediaries [8].

In an SSI framework, individuals own and control their identity data, which is stored in a decentralized manner across a network of nodes. Digital credentials, issued by trusted entities, are cryptographically secured and linked to the individual's identity. These credentials can include personal attributes, qualifications, or other relevant information. The individual can selectively disclose these credentials to third parties, enabling identity verification without revealing unnecessary personal details.

Unlike traditional identity systems, SSI eliminates the need for centralized databases and any need of contacting the issuer organization, reducing the risk of data breaches and unauthorized access. It provides a privacy-preserving solution by allowing individuals to share only the necessary information for specific transactions or interactions [6], [27].

There are three main participants or roles in the SSI system as depicted in Figure 2.1:

- **Holder:** Individuals can generate their decentralised identification by utilising a digital wallet application. They can then acquire Verifiable Credentials from various issuing organisations and securely store them within their mobile wallet. These credentials serve as evidence that can be presented at a later time to substantiate their claims.
- **Issuer:** The entity with the requisite authorization to give Verifiable Credentials to users, specifically organisations.
- **Verifier:** Entity who demands proof of control of a credential from a user and then verifies the person's credentials.

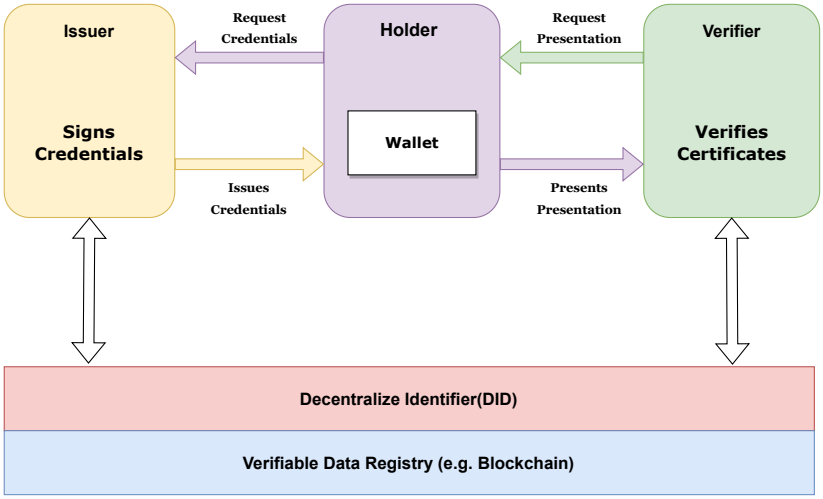


Figure 2.1: SSI Roles and their relations [16]

2.1.1 Principles of SSI

SSI holds significant importance in today's digital landscape due to several key reasons [10].

- **User Control and Privacy:** SSI enables individuals to have complete control over their personal data, allowing them to choose what information to disclose and to whom. This enhances privacy and reduces the risk of identity theft and surveillance.
- **Security and Trust:** By leveraging blockchain technology, SSI ensures the integrity and immutability of identity data, making it highly secure and tamper-proof. It eliminates the reliance on central authorities, promoting trust in identity verification processes.
- **Interoperability and Portability:** SSI promotes interoperability by enabling individuals to use their digital identities across multiple platforms, services, and organizations. This seamless integration enhances user experience and simplifies identity management.
- **Inclusion and Accessibility:** SSI has the potential to provide digital identities to underserved populations, including those without traditional identity documents. This inclusion can enhance access to essential services, financial inclusion, and social empowerment.
- **Reduced Identity Friction:** SSI aims to streamline identity verification processes by eliminating the need for repetitive identity checks and paperwork. This frictionless experience can enhance efficiency, reduce costs, and improve user satisfaction.

2.2 Building Blocks

SSI is a newcomer on the internet, having debuted in 2016. In part, SSI is a collection of standards for handling identities and private information in cyberspace. Self-Sovereign Identity (SSI) is constructed upon several key building blocks that form the foundation of this innovative digital identity framework. These building blocks comprise a combination of decentralized technologies, cryptographic protocols, and standards that work together to establish a secure, user-centric, and privacy-preserving identity ecosystem.

2.2.1 Decentralized Identifier (DID)

Decentralized Identifiers (DIDs) are a new kind of identifier that is crucial for Self-Sovereign Identity (SSI). They allow for verifiable credentials and operate without any centralized authority. According to the DID specification, DIDs are globally unique, highly accessible, and cryptographically verifiable. They are user-generated, much like an HTTP address, and can be created by anyone at any time [36].

Similar to URLs, DIDs can be resolved. When a valid DID is submitted to a DID Resolver, it responds with a DID Document (DIDDoc) that contains JSON data. This data often includes the public keys linked to the entity controlling the DID and service endpoints for communication with that entity. A visual representation of a DID example is given in Figure 2.2.

Figure 2.2: A simple example of a decentralized identifier (DID) [28]

DIDs are globally unique, meaning each one is unique worldwide when properly created. They are highly available, ensuring they can be resolved even when some servers are offline, as they are commonly resolved by reading from a distributed ledger. Cryptographic verification of a DID involves the DID controller demonstrating their control over the DID by showing control of a private key associated with a public key in the DIDDoc. This allows you to use a DID to obtain a DIDDoc, locate a public key, send a message to the entity controlling the DID, and request proof that they possess the private key associated with the public key. Once received, you can then verify this proof [19].

Public DIDs are broadly visible and typically utilized by organizations. They are stored on blockchains to ensure global accessibility and are used in verifiable credentials, enabling the verifier to check the issuer's identity. Conversely, private DIDs allow select individuals (i.e., holders, verifiers) to connect without being globally visible. These DIDs are shared directly between the involved parties and are not

listed on any public ledger. Writing data to a blockchain can be quite costly. Therefore, by avoiding the addition of DIDs that don't need to be publicly available on the blockchain, we can decrease the associated expenses, such as resources, time, and transaction fees [43].

2.2.2 Verifiable Credential (VC)

Verifiable Credentials (VCs) are vital components of Self-Sovereign Identity (SSI), enabling individuals to manage their digital identities. They are digital counterparts of physical credentials, with the advantage of being tamper-evident and having cryptographically verifiable authorship.

VCs involve three main participants: the Issuer (the entity issuing the VC), the Holder (the entity owning the VC), and the Verifier (the entity verifying the VC's validity) [33]. This setup mirrors the traditional physical credentialing system in a digital context. One of the key aspects of VCs is their use of public key cryptography. The issuer signs the VC with their private key, and the verifier checks the signature with the issuer's public key, ensuring the credential's authenticity and the issuer's authorship.

VCs contain claims or statements about the holder, which are the information pieces the verifier is interested in. Privacy is also a fundamental aspect of VCs. Techniques such as zero-knowledge proofs can be used to disclose only minimal required information, ensuring the holder's privacy. Finally, VCs are integrated with Decentralized Identity (DID), ensuring they are globally unique, highly available, and cryptographically verifiable [46].

2.2.3 Agent

In the Self-Sovereign Identity (SSI) ecosystem, an agent is a software entity that represents an individual or organization. It is the component of SSI infrastructure that is composed of software elements that represents an individual or an organization in the real world. It consists of several key components [43] as illustrated in Figure 2.3:

1. **Key Management Service:** The KMS is basically a secured storage for the agent to store all the sophisticated information it collects. Aries KMS is essentially an encryption layer around a standard database for storing and managing cryptographic keys, DIDs, connections, credentials etc. It also carries out many functions like encrypts, decrypts, creates, stores, signs data.

2. **Agent Messaging Interface:** It defines the protocols and standards for exchanging messages and data related to credentials, and proof-requests. The interface allows agents to securely send and receive messages as plain-text format even because it does not rely on transport encryption. Any information exchanged between the aries agents are protected by three layers of encryption:

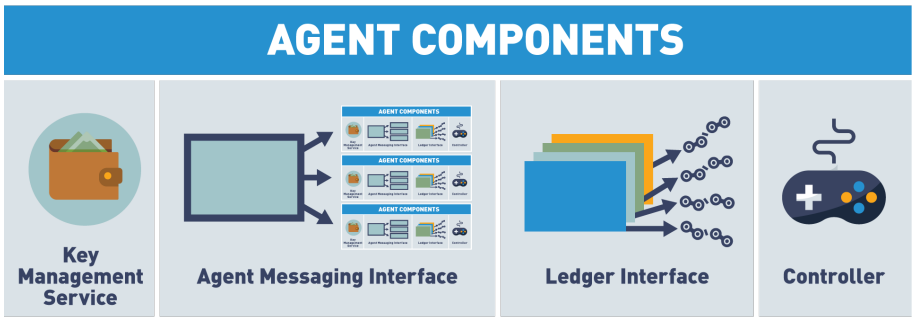


Figure 2.3: Agent Components [43]

3. **Ledger Interface:** The Ledger Interface facilitates interaction between the agent and the underlying distributed ledger technology (DLT) or blockchain. It enables the agent to read from and write to the ledger, allowing for the storage and retrieval of verifiable credentials and other identity-related data. The ledger interface ensures the immutability and transparency of the recorded transactions.

4. **Controller:** The Controller component manages the overall behavior and operations of the agent. It handles tasks such as identity verification, credential issuance, and consent management. The controller interacts with the key management service, agent messaging interface, and ledger interface to perform these operations securely and efficiently. It acts as the brain of the agent, making decisions based on predefined rules and policies.

   These components work together to give an agent in SSI the ability to manage digital identity operations securely and efficiently, providing individuals and organizations with self-sovereign control over their identities and data.

2.2.4 Digital Wallet

In the realm of Self-Sovereign Identity (SSI), a digital wallet serves essentially as an agent for users to store and manage their digital identities and credentials. It uses cryptographic keys to guarantee the privacy and security of data. The wallet also facilitates user interaction with other entities within the SSI ecosystem, enabling them to authenticate themselves as required. It offers portability and interoperability, permitting users to transport their digital identities across multiple services and platforms. Essentially, it's a user-focused approach to managing identity in the digital world. It is essential that the wallets contents can be transferred conveniently and securely preserved [15].

2.2.5 Verifiable Data Registry (VDR) & Blockchain

A Verifiable Data Registry (VDR) in Self-Sovereign Identity (SSI) is a key component of the SSI infrastructure. It is a system for managing the creation, reading,

updating, and deletion of decentralized identifiers (DIDs), which are unique identifiers tied to a digital identity.

The VDR is responsible for storing DIDs and their associated public keys, service endpoints, and other metadata. This stored information is used to establish and verify digital identities and their credentials. When a verifier needs to validate a digital identity or credential, they can refer to the VDR to check the associated DID and public keys [45].

Figure 2.4: Blockchain for Verifiable Data Registry(VDR)

Let's explore how blockchain operates as a VDR in SSI. Blockchain, a distributed digital ledger that records transactions on numerous computers, is inherently decentralized. This characteristic aligns perfectly with SSI's principle of independence from centralized authorities. By adopting blockchain as a VDR, it's assured that DIDs and DID Documents are universally accessible and not under the control of a single organization [15].

Furthermore, blockchain's cryptographic security increases the dependability of digital identities. The blockchain data is immutable, implying that once a DID is saved, it can't be modified or erased. This immutability offers a solid audit trail, boosting confidence in digital identities. Blockchain also helps accomplish SSI objectives by facilitating peer-to-peer transactions. In the SSI framework, individuals can share their credentials directly with others, eliminating the need for an intermediary. Blockchain's decentralized nature enables this direct exchange, increasing user privacy and control [37].

Finally, blockchain's consensus algorithms guarantee data integrity. They confirm and agree on the data's state before incorporating it into the blockchain. This is crucial in preserving the accuracy of DIDs and preventing fraudulent actions. In summary, blockchain's decentralization, cryptographic security, support for direct transactions, and consensus processes are crucial in its role as a VDR in SSI. They closely align with SSI's principles and goals, delivering a secure, transparent, and user-focused solution for managing digital identities [21].

There are four key variables that must be taken into consideration when evaluating a blockchain as a potential Verifiable Data Registry within the framework of Self-Sovereign Identity (SSI). These factors pertain to the nature of the blockchain, specifically whether it is categorised as public, private, permissioned, or permissionless, as illustrated in Figure 2.5 .

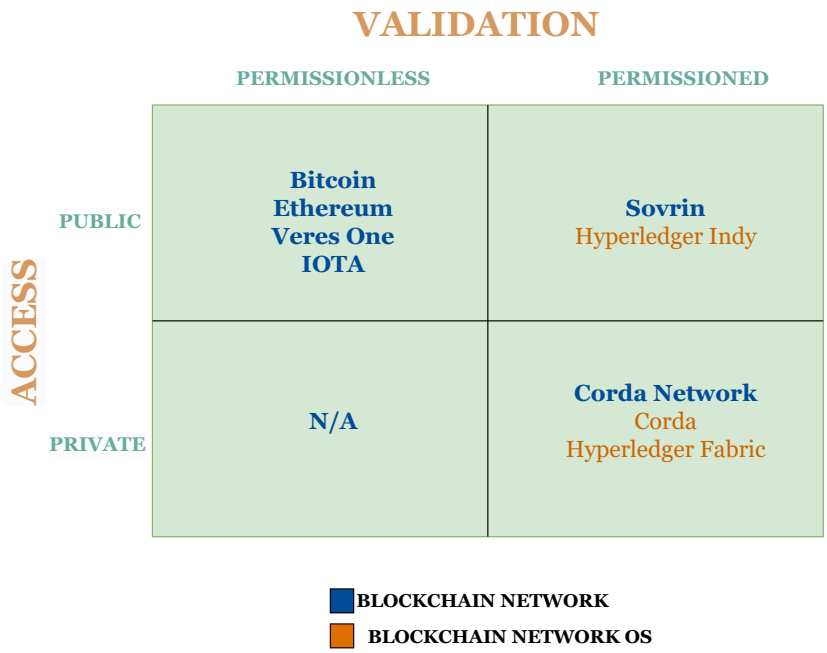


Figure 2.5: Types of Blockchain

1. **Public Blockchains:** These are open to anyone, allowing participants to access the blockchain freely. In SSI, they can serve as independent and permanent records of Decentralized Identifiers (DIDs) and related data. However, they have limitations concerning scalability, data privacy, and energy consumption.

2. **Private Blockchains:** These are accessible only to specific participants. They offer higher scalability and data privacy compared to public blockchains but are less decentralized. Hyperledger Fabric is an example. These blockchains can provide a controlled environment for identity management in SSI, ensuring data privacy and efficiency.

3. **Permissioned Blockchains:** These are a type of private blockchain where network participation and validation is restricted. While they allow for greater control and privacy, they can also provide a level of decentralization if governed by a consortium of different entities. Hyperledger Indy, designed for identity management in SSI, is an example.

4. **Permissionless Blockchains:** These are typically public blockchains where anyone can validate transactions. While they offer high levels of decentralization and transparency, they may not be ideal for SSI due to potential privacy and scalability issues.

The choice of blockchain type for SSI as a VDR largely depends on the specific requirements for decentralization, scalability, privacy, and control.

Chapter 3

Literature Review

Barros, Schardong and Custodio discuss the use of SSI utilizing Blockchain for creating COVID-19 vaccination certification that take care of different privacy and data protection concerns [44]. The vaccination certification scheme posits a similar problem to identity verification where it is necessary to check if people are who they actually say they are. With this, comes the added responsibility of providing protection to the data in two paradigms. One such that, a vaccinated person can verify his/her vaccination without revealing personal data, vaccination dosage or vaccine name. Another is that of not creating a central data repository of all vaccinated persons, which can potentially be a single point of failure and moreover be used to extract data about individuals. Compared to other identity management techniques and ways to solve this specific problem, the authors propose a framework which not only solves the problem in a better way but also provides interoperability between different governments and authoritative entities. The paper outlines the underlying technologies used in the proposed system, including the use of blockchain, specifically Hyperledger Indy, to store decentralized identifiers (DIDs) and issue verifiable credentials (VCs). It also utilizes an abstraction layer called Hyperledger Aries to handle the creation, transfer, and storage of VCs and verifiable presentations (VPs) independent of the underlying blockchain solution. The proposed framework uses SSI so that users can own their data and give access to other parties according to their wish. A blockchain system using Hyperledger Indy is used to create the DID issuers which are the vaccination authorities. Using DIDComm a vaccinated person gets the vaccination certificate which is a Verifiable Credential (VC) in his/her digital wallet. The digital wallet can be any secure personal assistant device. This facilitates no single central repository as people own their personal vaccination certificate and also gives control of data using Zero Knowledge Proof and Verifiable Presentation(VP). Using this, people can control what presentation or what part of the certification attributes they want to give access to other parties and Zero Knowledge Proof scheme helps verifiers identify vaccinated people from those Verifiable Presentations (VP). The verifier can verify the VP using Hyperledger Indy to check if it originated from a legitimate vaccinator and if the VC is still valid and unrevoked. The paper compares its approach with similar initiatives, such as the Good Health Pass Collaborative, the WHO's Digital Documentation of COVID-19 Certificates, and the EU Digital COVID Certificate. The proposed system mitigates some of the problems in these methodologies, which include lack of interoperability and scalability. Overall, the proposed architecture offers a more privacy-preserving

and user-centric solution compared to other initiatives. Using the combination of SSI principles with blockchain technology to achieve a globally scalable and secure system for proving vaccination status while protecting individual data privacy makes it a better candidate system.

Feulner et. al examines the usage of SSI alongside blockchain technology to help in the prevention of event ticket scalping, fraud and bot activities with a view to ensuring efficient identity verification and privacy protection [38]. Event ticketing sites now have a problem of bots creating 40% of their whole traffic as well as bots buying up tickets for reselling in the secondary market. To bind identity strongly to the buyers different techniques have been suggested such as dynamic QR code and blockchain technologies. The paper studies the use of SSI to manage identity for mitigating these challenges. The study used a design science research approach where they identified the mentioned problems, set objectives to mitigate those using SSI alongside blockchain, created artifacts as proof of concept and showcased the artifacts to different stakeholders for gaining their insights of the system. For ensuring and binding identity, the system uses either government authority based or any other KYC based identity providing system which binds identity with people and generates Verifiable credentials for them to store in their own digital wallets. In the ticketing site, using Verifiable presentations from their own digit wallet, users buy tickets which generates another verifiable credential of the ticket for them. Now for entry they can use QR code scans to provide their verifiable presentation to ensure their entry and also secure their private data. In the secondary market, the original ticket VC is revoked and the data is kept in a public revoking registry while a new ticket VC is generated for the buyer. A series of interviews with experts showcasing the proof of concept artifact validated and reviewed the system. It mostly met the targets of secondary market control and ticket validation. For secondary market reselling, some concerns rose over the data revelation of buyers, but such can be solved with using Zero Knowledge Proof(ZKP) scheme. Although unambiguous identification reduced ticket scalping, a complete elimination of scalping. The framework demonstrated reliable and efficient user identification, enhancing security and privacy while providing valuable insights for digital identity management solutions in various contexts required measures to prevent underpricing and potential ticket fraud activities.

The Research conducted by Satybaldy et al. deals with verification of digital documents using Self-Sovereign Identity technology(SSI) [24]. Over the years many techniques have been developed to verify a physical document, but for a digital document there are not many methods available, thus digital documents are prone to forgery. The solutions currently available are expensive and not universally compatible. SSI is built upon the core motivations of distributed Ledger, Decentralization, Cryptography and aims to give total control of peoples private information to themselves. The Models used in this research are Decentralized Identifiers(DID) and VC(Verifiable credentials). DID is resolvable, globally unique, decentralized and cryptographically secure, It can exist without an involvement from third parties. The VC model ensures trust among the parties involved in the ecosystem. It includes issuer, verifier, holder and data repository that can be verified. The paper used a DSR approach and put forward a framework for digital document verification

by targeting decentralized technology. By using SSI and with the help of experts this paper demonstrates how digital documents can be verified.

The procedure presented by Bahrami et al. in [12] consists of three parts. The academic issuers first join the network. An organization creates its address, public and private keys, and blockchain network membership using a wallet. Using smart contracts, the applicant then contacts one of the networks members. Only a select few peers are authorized to perform transactions and write to the ledger. Different access levels and licenses provide the following advantages: greater security, improved work distribution among peers, increased fewer executions, synchronization, and parallelism bottlenecks. Self-executing agreements may be automatically activated using smart contracts. When a registration request is sent to a university, a token is delivered to the institutions address. The institution then makes a request for authorization, including the address of the blockchain through a private channel. Through other network members, the current institution confirms the existence of the new universitys transaction. When a user enrolls at a university or other institution that is part of the blockchain network, for the student, a new multi-signature address is created. This multi-signature address is recorded in the database with the user ID. Multiple private keys are needed to sign a transaction under this protocol. The inability of users to transfer and assign scores from one course to another indicates that data security will be maintained even if one of the keys is compromised. By ensuring that generation of the keys and kept using different devices, there is a chance of data loss and infection with malware, as well as a private key being lost and access issues, may be mitigated [12].

In [32], it describes a full-fledged framework for an academic certificate verification system using blockchain. The main contribution of this paper is to analyze the needs of different stakeholders in the system. Three main stakeholders are associated with such a system; namely students, educational institutions and employer institutions. These parties need to get a simple manageable way to verify certificates. But as different institutions need so many different types of verified certificates, students create long queues of requests, which create long waiting time for them and a hassle for the verifying institutions. An automated system can solve this problem, but both the problem of forging certificates and securing the data of such records are at high risk in such a system. That is where blockchain technology comes into play. Blockchain creates the distributed ledger system of academic records, which keeps the records from being corrupted if they were on a centralized server. So, no piece of academic record can be forged as data is kept in multiple different servers with unique identifying hashes. If any piece of data is changed in a specific server, the hash will be changed. When the other servers will match, the hash will not match and forgery will be caught. Thus, certificate-provider institutes can create certificates and keep them in such a decentralized system to protect certificates from being forged. Also, multiple requesting institutions can access such records at different times, which creates the risk of sensitive academic data being breached. RSA cryptography is suggested by the author as a solution to that. Public key cryptography will keep the data encrypted and the private keys will only be supplied to institutions requiring the verification. This whole framework is a complete novel solution to address the problem at hand. The author urges to deeply analyze the

needs of different stakeholders even more.

The work done by Bahrami et al. as previously mentioned consists of three parts. The academic issuers first join the network. An organization creates its address, public and private keys, and blockchain network membership using a wallet. Using smart contracts, the applicant then contacts one of the networks members. Only a few peers are permitted to conduct transactions and write to the ledger. Different access levels and licenses provide the following advantages: greater security, improved work distribution among peers, increased fewer executions, synchronization, and parallelism bottlenecks. Self-executing agreements may be automatically activated using smart contracts. When a registration request is sent to a university, a token is delivered to the institutions address. The institution then makes a request for authorization including the address of the blockchain through a private channel. Through other network members, the current institution confirms the existence of the new universitys transaction. When a user enrolls at a university or other institution that is part of the blockchain network, for the student, a new multi-signature address is created. This multi-signature address is recorded in the database with the user ID. Multiple private keys are needed to sign a transaction under this protocol. The inability of users to transfer and assign scores from one course to another indicates that data security will be maintained even if one of the keys is compromised. By assuring that generation of the keys and kept Using different devices, there is a chance of data loss and infection with malware, as well as a private key being lost and access issues, may be mitigated [12].

Chapter 4

Threat Modelling & Requirements

For this part, we will introduce a threat model which is a core measure of security when integrating such a blockchain system. This model will help us to comprehend, recognize the threats and mitigate the threats in order to protect the authenticity of the certificates mentioned in this paper. We will also highlight the functional, security and the privacy necessities for our SSI integrated system. We have opted for a model called STRIDE, which is a well established threat model. Now, we will present a variety of the possible security risks which we might face.

- **T1 - Spoofing Identity:** This process refers to the attacker taking on the identity of the authorized user to gain access and attain a malicious goal.
- **T2- Tampering with Data:** The attacker might try to change the information of the academic certificates such as the result or gpa according to his convenience.
- **T3 - Repudiation:** The attacker might repudiate unlawful actions such as certificate forgery.
- **T4 - Information Disclosure:** Crucial and highly sensitive information such as certificate results is disclosed to the attacker either intentionally or unintentionally.
- **T5 - Denial of service (DoS):** The online servers providing SSI functionalities can become unavailable if they become the target of a DoS attack.
- **T6 - Elevation of Privilege:** The attacker can elevate his privilege by means of attack vectors such as a bug in the login system, session-hijacking etc., which can exploit the authentication system.

Apart from STRIDE, there are other security threats which are defined as the following:

- **T7 - Replay:** An attacker might capture an request/response and submit it afterwards, thus launching a replay attack.
- **T8 - Lack of consent:**The processes might be carried out without the permission of the users
- **T9 - Lack of control and Transparency:** The users will have minimal control over the verification process.

4.1 Functional Requirements

In this part, we highlight the functionalities that our system can carry out. The functional requirements represent the key features that this SSI system will contain.

F-1: In this system, each student can have their unique ID in a secure network, which can serve as an independent username and/or as supplementary student information.

F-2: The systems unique ID's with Decentralized Identifiers (DID) should exchange data between the student and organisation.

F-3: The DID should be secure, temper- evident and machine verified using cryptography calculations which should be issued by the authority.

F-4: The system should ensure that only certified organisations will be allowed to issue the credentials.

4.2 Security Requirements

These are the necessary features required once the threat is identified. These requirements should be fulfilled in order to minimize the security risk which we can face.

S-1: The system will be implemented such that only the securely authorized users can gain access to the SSI system and carry out the certificate verification process. This will help us to address the threat T1.

S-2: The system needs to prevent any unauthorized alteration to the VC related data to limit the threat, T2.

S-3: The SSI system should deploy the usage of digital signature to mitigate the T3 threat.

S-4: Information must be conveyed via the encrypted paths maintaining integrity of the VC associated data. This can help mitigate the threats T2, T3 and T4.

S-5: This system should be such that all preventive steps must be taken against DoS attack which will counteract T5.

S-6: By requiring authentication and authorization and maintaining an access control mechanism, the system should ensure that attackers cannot elevate their access privileges.

S-7: To reduce the threat that T7 poses, the system needs to take steps against replay attacks.

4.3 Privacy Requirements

Here, we state the required steps that need to be taken to counteract the threats we stated and identified.

P-1: All the verification and authentication process must take place with the permission of the user. This reduces the T7 threat.

P-2: The user must have full authority and control over the authentication process and this corresponds to the threat T8.

Chapter 5

System Model & Architecture

5.1 SSI Framework:

Current identification schemes include vulnerabilities, which strengthens the argument for digital identity. Self-sovereign identification may support a digital identity paradigm that is more user-centric, trustworthy, and discreet (SSI) to combat this. The model consists of three major participants in the system: Issuers, holders and verifiers also known as the trust triangle (Figure 5.2).

A decentralized identifier (DID) is a unique string of letters used to refer to a particular person, location, or item on the internet. These identifiers are often used as trusted identifiers, encoded inside verifiable credentials, and connected to subjects in such a manner that verifiable credentials do not need to be reissued after being transferred from one repository to another. Digital Identifiers are references to DID documents that offer information about DID controllers and DID subjects (DIDs). Decentralized IDs are kept in a document known as a decentralized identifier document (DID Doc), which is accessible through a trusted data registry. This document describes the decentralized identification, including its related repository, supplementary metadata, public key information, etc. [29].

Since issuers sign the VCs, any third party (Verifier) having access to their DID may validate their validity by matching the signature to the issuers public key. A holder of VCs may store them in a digital wallet until they are required, at which time they may be provided to a verifier as individual credentials or as a combination of numerous VCs. Since blockchain is a distributed, immutable database, it is appropriate for use as a trustworthy registration for DIDs and DID Documents, since no entity may edit its contents.

5.2 ACA-Py Framework

Aries Cloud Agent - Python (ACA-Py) is an open-source software agent developed by the team at Hyperledger Aries. It's designed to provide a foundation for developing decentralized identity applications and services that can run in various

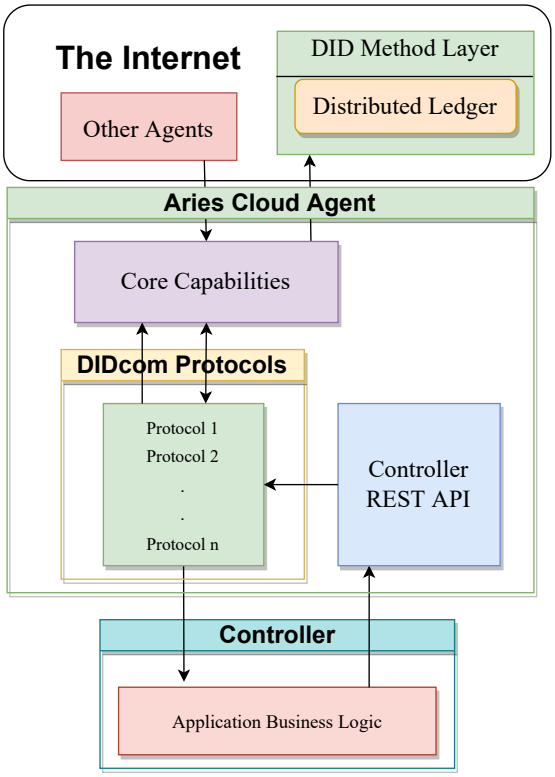


Figure 5.1: ACA-Py Framework Illustration [41]

non-mobile environments. ACA-Py is built to interact with other agents in the Aries ecosystem, allowing developers to create, manage, and use independent digital identities that are rooted in verifiable data registries such as distributed ledgers (blockchains) [32]. In Figure 5.1, we illustrate the internal architecture of ACA-Py.

ACA-Py plays a significant role in the SSI ecosystem as it supports the implementation of SSI by providing a robust and flexible tool set for building applications that can issue, prove, and verify digital credentials in a manner that is compliant with the standards established by the Aries community and W3C [41]. It allows various entities to interact in a peer-to-peer manner, enabling the exchange of verifiable credentials in a decentralized identity system [24].

ACA-Py employs a modular and pluggable design in which distinct SSI functionalities are handled by different components. The controller defines a collection of *REST API* endpoints for various SSI activities and functionalities. These endpoints are usually classified according to the functionality they provide, such as credential issuance, proof presentation, connection initiation, and so on. External applications or services, such as mobile apps or web apps, can interact with ACA-Py by sending HTTP queries to the proper API endpoints. These queries adhere to *RESTful* principles, employing standard HTTP methods such as *GET, POST, PUT, DELETE*, and so on. ACA-Py returns an appropriate HTTP response to the requesting application after processing the API request and performing the relevant SSI functionality. The response typically provides the result of the operation or any other relevant information. For complicated interactions involving several participants, distributed ledgers, and external systems, ACA-Py offers an asynchronous

communication technique. ACA-Py handles this through the use of callbacks or *webhooks*, which allow external applications to get notifications or responses when the operation is finished or when specified circumstances occur [41].

5.3 Architecture and Implementation

5.3.1 System Overview

We have previously outlined the three roles involved in academic certificate authentication using Self-Sovereign Identity (SSI) and blockchain technology, along with their interactions which can be also be seen in Figure 5.2. As we have discussed before, Academic certificates have been subjected to fraudulent activities in the past and the individuals have no control over the amount of information shared or how it is shared in the conventional process. Furthermore, the verification process of certificates in the traditional way is a long and resource consuming process where too many parties are involved. In realization of this problem, we have implemented a proof of concept system using an SSI ecosystem. The system employs the three roles of an SSI ecosystem by enabling academic institutions to issue certificates to their students as issuers, students as holders who can securely store the certificates in their wallet and can present for proof to the parties interested if they wish; and lastly, other organizations as verifiers who can verify any academic certificates the holder possesses.

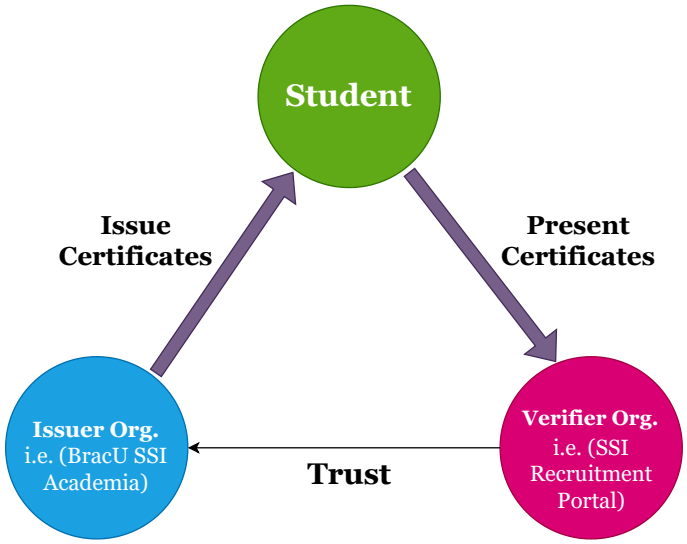


Figure 5.2: Trust Triangle of SSI

In our system, an organization can take up the role of an issuer or a verifier. SSI Agents of such organizations are capable of issuing a certificate or ask a holder to present proof of their certificates and these organizations must hold public DIDs to avail these capabilities. Academic institutes, training centers etc. who need to provide certifications or validate them before hiring people are examples of such organizations. To depict such a scenario, we have created an SSI agent for Brac University which can issue its students with certificates upon graduation or when a student requests for it. Organizations may require to verify a persons academic

certificate before recruiting or for authentication purposes including academic institutions, corporate companies, jobs or freelancing portals i.e. anywhere (both offline or online) where you need to show and prove a claim about your academic qualifications. To portray how organizations can incorporate SSI to verify a students certificate, we created a job portal where companies can post job vacancies and ask applicants to present proof of their certificates before applying for a job. The verification of whether or not an individual meets the academic requirements for a job is verified through SSI protocols even before a person applies for a job hence adding a layer of trust and quality assurance among the eligible applicants. This cuts short time and resources needed on a massive scale rather than going through the documents and certificates of applicants manually contacting the authorities. With our system, there is no central authority holding the certificates or no need for contacting the issuer organization for verification. And because of the cryptographic base and immutable blockchain technology, our system will provide the means to present claims about peoples certificates and anyone (*i.e.*, organizations) can verify those claims with cryptographic guarantee. In the whole process, the user decides what information they want to keep as well as with whom they want to share the information to what extent thus giving the user full control of their certificate data.

We now delve into our system's architecture and the tangible interactions of the roles of SSI using the aforementioned technologies. Figure 5.3 depicts the architecture of our system. Firstly, it's crucial to acknowledge that when an organization first adopts our solution, certain preparatory steps are necessary. Specifically, an Aries agent must interact with the Indy blockchain to:

(a) Store the public DID of the institution that will issue academic certificates in the ledger; and

(b) Establish the format of the Verifiable Credentials (VCs) that will be issued, i.e., define the schema and credential definition and publish it to the distributed ledger.

Following the above preparations, each organization authorized to issue or verify certificates must control an Aries agent via software, mobile app, or website. The Aries agent can operate on the same device that controls it or remotely from an organization's or third-party cloud data center. The Aries agent, managed by the academic institution, generates a public DID, which corresponds to a DIDDoc containing a public-private key pair of the institution for a specific SSI connection. The issuer agent securely stores the private key and uses it to sign and issue a VC.

In the BracU SSI Academia, students are added by the authority by creating accounts for them along with their grade sheet and student information. Added students can then login to the website and view their academic information such as grade sheet. Before carrying out any credential exchange, students must establish an SSI connection between the BracU′s agent and students mobile wallet following the DIDComm protocol by scanning a QR code. Students then will have access to Issue certificate option where they can select what information they want BracU

to include in the certificate from the grade sheet. Once the student requests for a VC, selected information from the grade sheet along with some mandatory student information is fetched from the BracU database and a request for the issuance of a VC from the student is made to the BracUs SSI agent along with date of issuance via an HTTP call through the *REST APIs*. The information sent to the agent can be personalized. Multiple issuing organizations can grant a credential or certificate to the same holder. The Aries agent, managed by the academic institution, fetches the credential definition from the *Indy ledger*, signs the information received in the request and generates a certificate by using the private key corresponding to the public DIDDoc BracU owns. It is noteworthy that the blockchain only stores the DIDs of the issuers, verifiers, or holders, the schema of the VC, and optionally, the value of the cryptographic accumulator to validate the unrevoked status of the VC.

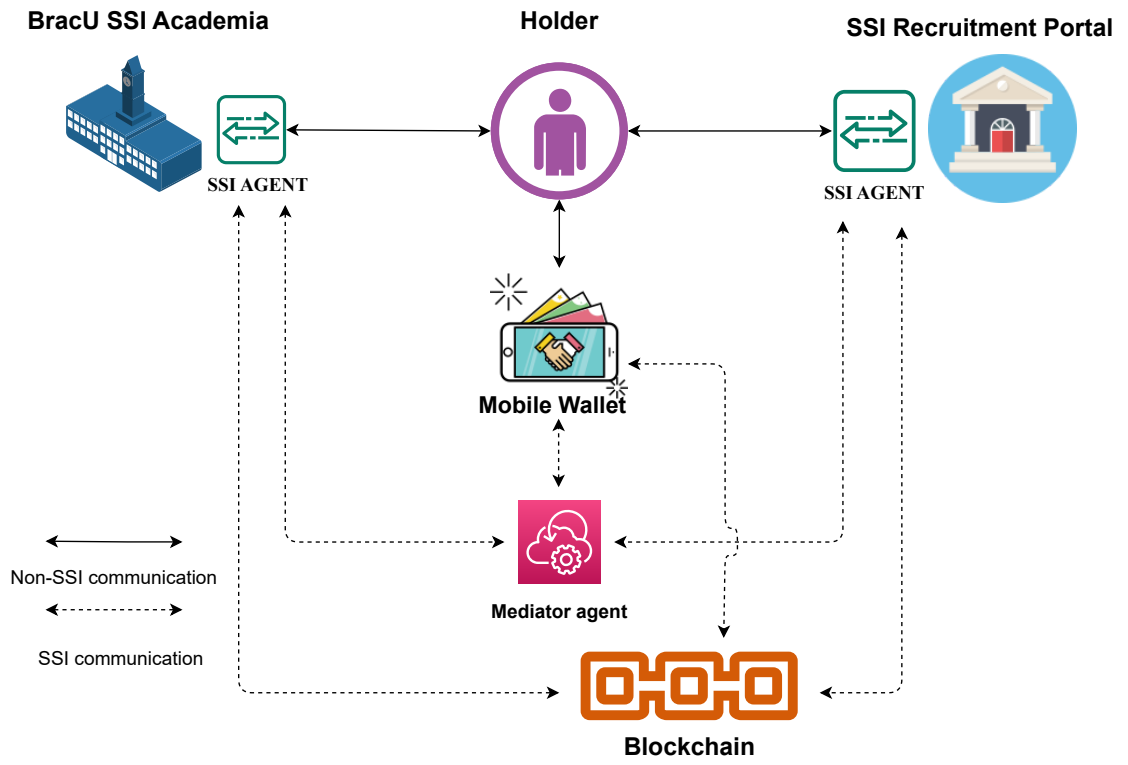


Figure 5.3: System Architecture of Proposed System.

Organizations wishing to verify can use their own agents or trusted third party like our SSI Portal to verify academic certificates. Organizational representatives can create accounts in SSI Portal to post jobs with their specific requirements set. Students can login to SSI portal and apply for jobs after presenting proof of their education. Just like with the Issuer agent, students need to create a secure SSI connection with the verifier agent to avail any credential exchange via DIDComm by scanning a QR code using the mobile wallet. During the job post process, organizations can specify which attributes of the certificate they want validated rather than the complete credential. Students interested in the particular job must generate a verifiable presentation to validate those attributes required by the organization before continuing with the application process. As students proceed to initiate the *proof-request*, they receive a prompt in their mobile wallet whether or not to accept

the incoming presentation request from SSI Portal. The secure environment of the digital wallet creates the VP and sends it via DIDComm to the verifier's Aries agent. The verifiers agent connects to Indy to verify whether the VC that originated the VP was actually issued by BracU by matching the signature on the VP using the public DIDDoc of BracU. All data exchanges are private and peer-to-peer, between the issuer and holder and between the holder and verifier. If the same organization issues certificates and wants to verify the credentials, then the same entity can serve as both the issuer and verifier.

5.3.2 Implementation

The tools mentioned previously are utilized to devise a solution that ensures privacy while verifying online educational certificates. When it comes to the blockchain acting as a verifiable data registry for storing our DIDs, schema, credential definitions, and so on, our Proof of Concept employs an Indy testbed named **BCovrin Dev *(http://dev.greenlight.bcovrin.vonx.io/)*** which is a part of **VON** development network. Developed by Hyperledger, Indy is the inaugural identity-centric blockchain framework, functioning as a network of compatible SSI networks. The Sovrin Foundation contributed the code for Indy, which is primarily a public, permissioned distributed ledger designed explicitly for decentralized identity applications. Only Validator nodes can write on the Indy ledger and such nodes with endorser role are utilized by the issuer organizations since they need to store their public DID in the ledger for issuing and verification purposes. Holders or other organizations without issuing capabilities do not have the permission to write transactions on the ledger as they have no use for a public DIDDoc for issuing. It includes components for developing agents that support the publication of decentralized identifiers (DIDs) and associated identity objects on its ledger. Initially conceived as an all-encompassing SSI identity solution, Indy has been strategically divided into distinct components to cater to a variety of identity setups. At present, Indy comprises the ledger, tools for its administration, and client libraries for ledger interaction. This structure enables various SSI systems to share verifiable credential metadata and interoperate, thereby promoting SSI adoption.

Two responsive websites were developed for the proof-of-concept implementation, compatible with both desktop and mobile devices, using **NodeJS, ReactJS, Bootstrap, HTML, Python** etc. NodeJs and React was used to develop the SSI agent controllers as well as the web services which depend on the ACA-Py framework to provide different SSI services.. The issuer utilizes one website (BracU SSI Academia) for issuing Verifiable Credentials (VCs), while the verifier employs the second website (SSI Recruitment Portal) to define Verifiable Presentation (VP) formats and present this request to students. Any connections between the servers and clients are secured with *HTTPS Secured Socket Layer* encryption. The backend of each website was integrated with an ACA-Py Aries agent. Our choice of Aries agent implementation was the proprietary *Indy Cloud Agent - Python*, selected for its support of Decentralized Identifiers (DIDs) and Verifiable Credentials [30]. This implementation empowers individuals to authenticate their identities or certificates securely. It promotes interoperability through its shared toolkit, secure messaging protocols, and agent-to-agent communication for credential exchange. Additionally,

it offers a flexible framework for the secure storage of identity information, courtesy of various wallet plugins and storage methods.

When a student visits the issuer's website to request a certificate, an underlying process is initiated where the system prompts its Aries agent to generate a Verifiable Credential (VC) and subsequently transfer it to the student's digital wallet. Prior to the transfer, the student's agent must establish a Self-Sovereign Identity (SSI) connection with the issuer's agent, adhering to the **DIDComm protocol**. This connection is established by accepting a connection invitation from the issuer website, facilitated through the scanning of a QR code provided by the issuer agent, as depicted in Figure 5.10 in section 5.4.4. In our implementation, we utilized the ***Aries Mobile Agent React Native (Bifold)*** digital wallet to simulate student interactions. There is a range of digital wallets available, both free and paid, that have the capability to receive Verifiable Credentials (VCs) and produce Verifiable Presentations (VPs), such as connect.me. The *Aries Mobile Agent React Native (Bifold)* serves as an intuitive mobile agent, leveraging React Native and Aries Framework JavaScript (AFJ) to facilitate secure exchanges of verifiable credentials among agents. AFJ incorporates Rust libraries, particularly Indy-SDK, which are compiled into native code. Indy-SDK utilizes the ZMQ protocol to establish communication with the Indy ledgers [40]. AFJ communicates with Aries agents using the HTTP protocol and employs WebSockets for messaging through a mediator. Bifold relies on a mediator due to the dynamic nature of mobile devices, which lack fixed IP addresses and often restrict inbound network connections.

The mediator agent that we used in our system is The ***Indicio Mediator Agent*** which is hosted on a server with a fixed IP address, acts as an intermediary, facilitating message relay between the agent and Bifold. The configuration of the mediator is conducted within the Bifold app. In scenarios where direct communication is not possible, the Indicio Mediator Agent plays a crucial role in message mediation. It stores messages until the recipient becomes available, ensuring reliable and secure communication. Understanding the crucial role of mediation in realizing decentralized identity, Indicio capitalized on the contributions from the open-source community to achieve mediation that is both interoperable and vendor-agnostic. The Mediator Agent was seamlessly integrated into the Aries Cloud Agent Python (ACA-Py) and the Aries Toolbox. As a result, the inbuilt Admin REST API of ACA-Py now enables the handling and debugging of mediation [42].

5.4 Use-case & Protocol Flow

In this section, we are going to explore different uses cases with their protocol flows to portray how our proposed architecture can be used in solving the challenges of certificate verification in the traditional system. At first, Table 5.4.1 presents mathematical notations that will be used frequently throughout the following sections.

5.4.1 Data Model

A comprehensive Data Model designed for the proposed model is illustrated in Table 5.4.2. Our developed proof of concept is designed as a responsive system that

Notations	Description
I	Issuer Organization
V	Verifier Organization
S	Student (Playing the role of a holder)
M	Mediator
O	Organization i.e. I(Issuer) or V(Verifier)
BA	BracU SSI Academia (Playing the role of an issuer)
RP	SSI Recruitment Portal (Playing the role of a verifier)
R	Recruiter who post jobs and set proof-request attributes
A	Admin of BracU SSI Academia
K_O^S	Public key of O for S
$K_O^{-1\|S}$	Private key of O to be used with S
K_S^O	Public key of S for O
$K_S^{-1\|O}$	Private key of S to be used with O
DID_O^S	Decentralized identifier of O for S
DID_S^O	Decentralized identifier of S for O
VC_O^S	A verifiable credential issued by O to S
N_i	A fresh nonce
$\{\}_K$	Encryption operation using a public key K
$\{\}_{K^{-1}}$	Signature using a private key K^{-1}
H(M)	SHA-512 Hashing operation of meesage M
$[...]_K$	Communication over an channel encrypted with key K
[...]	Communication over an unencrypted channel
[[...]]	Optional data

Table 5.4.1: Cryptographic Notations

generates some responses (denoted with *res* in the table) based on the requests made (denoted with *req* in the Table 5.4.2). A *req* can be of 7 types here : *serviceReq, studentInfoReq, attributeReq, jobDetailsReq, inviteReq, credentialReq* and *presentationReq*. Among them, the first four correspond to services that the system provides to enable SSI functionalities whereas the last three involve carrying out those SSI functionalities. Similarly, there are 9 types of responses and among them *serviceRes, studentInfoRes, attributeRes* and *jobDetailsRes* correspond to the first four types of requests previously mentioned. Then the last three requests trigger responses named *connectionRes1, connectionRes2, credentialRes* and *presentationRes* respectively. The *credentialAck* is utilized as a notification response after completing an SSI functionality *i.e.*, issuing a credential.

The *serviceReq* includes all the url_s of the services the system can provide and optional **JSON Web Tokens** as Auth-Tokens. On the other hand, the *serviceRes* correspond to the webpages or functionalities provided when these url_s are resolved or an API is triggered. Then, the *studentInfoReq* contains a list of attributes of student information and *studentInfoRes* contains the attribute values of a particular student. The *inviteReq* contains the *url* of the BracU SSI agent along with a nonce

which will be used in the next two consecutive messages. The *inviteReq* is followed by *connectionRes1* and *connectionRes2*; where the first one contains the private DID of the student (i.e. $\mathbf{DID}_S^O$) that would be used with this particular organization and the nonce that was sent in *inviteReq*. Likewise, the *connectionRes2* contains the same nonce and a DID of the organization (*i.e.*, $\mathbf{DID}_O^S$) for the student.

req $\triangleq$ ⟨ *serviceReq, studentInfoReq, inviteReq, connectionReq, credentialReq, attributeReq, jobDetailsReq, presentationReq*⟩ *res* $\triangleq$ ⟨ *serviceRes, studentInfoRes, connectionRes, attributeRes, credentialRes, credentialAck, jobDetailsRes, presentationRes*⟩
serviceReq $\triangleq$ ⟨ $urls_i$, [[*jwTokens*]]⟩
studentInfoReq $\triangleq$ ⟨$attr_1$, $attr_2$, $attr_3$, ... , $attr_n$⟩
studentInfoRes ⟨($attr_1$, val_1),($attr_2$, val_2),($attr_3$, val_3), ... , ($attr_n$, val_n)⟩
inviteReq $\triangleq$ ⟨url_i, N_i⟩
connectionReq $\triangleq$ ⟨ DID_S^O, N_i ⟩
connectionRes $\triangleq$ ⟨DID_I^O, N_i ⟩
attributeReq $\triangleq$ ⟨$attr_1$, $attr_2$, $attr_3$, ... , $attr_n$⟩
attributeRes $\triangleq$ ⟨($attr_1$, val_1),($attr_2$, val_2), ($attr_3$, val_3), ... , ($attr_n$, val_n)⟩
credentialReq $\triangleq$ ⟨Requesting Certificate, *attributeRes*⟩
credentialRes $\triangleq$ ⟨ VC_I^S ⟩
$\text{VC}_I^S \triangleq \langle\{(attr_1, val_1), (attr_2, val_2), (attr_3, val_3), ... , (attr_n, val_n)\}_{K_i^{-1}} \rangle$
credentialAck $\triangleq$ ⟨Credential stored in holder wallet successfully⟩
jobDetailsReq $\triangleq$ ⟨$attr_1$, $attr_2$, $attr_3$, ... , $attr_n$⟩
jobDetailsRes $\triangleq$ ⟨($attr_1$, val_1),($attr_2$, val_2), ($attr_3$, val_3), ... , ($attr_n$, val_n)⟩
presentationReq $\triangleq$ ⟨$attr_1$, $attr_2$, $attr_3$, ... , $attr_n$⟩
presentationRes $\triangleq$ ⟨VP_I^S⟩
$\text{VP}_I^S \triangleq \langle\{(attr_1, val_1),(attr_2, val_2), (attr_3, val_3), ... , (attr_n, val_n)\}_{K_i^{-1}}\rangle$

Table 5.4.2: Data Model

An *attributeReq* is sent for requesting credential attribute value and it is composed of a list of attribute names whereas the *ttributeRes* contains the value with their corresponding attribute names. The *credentialReq* includes a string as well as utilizes the *attributeRes* for making the credential request. Next, the *credentialRes* contains a verifiable credential denoted by $\mathbf{VP}_I^S$ which was issued to a student by some organization. Moreover, the $\mathbf{VP}_I^S$ is a set of attribute name-value derived from the attributeRes and is digitally signed using the private key of an issuer that corresponds to its public DID. A *credentialAck* contains only a string like shown in

the Table 5.4.2. Similar to *studentInfoReq* and *studentInfoRes*, *jobDetailsReq* and *jobDetailsRes* consists of a list of job information attribute names and a set of attribute name-value pairs respectively. Lastly, the *presentationReq* consists of the list of attribute names that must be included with verifiable presentation generated by the holder wallet and the corresponding *presentationRes* includes the released *VP (verifiable presentation)* derived from the $\mathbf{VC}_I^S$. The $\mathbf{VP}_I^S$ is formed of a set of attribute name-value pairs that was specified by the verifier organization.

5.4.2 Algorithms

Now let us go through 3 major algorithms utilizing which the SSI Agent Controllers facilitates SSI functionalities like handling a connection invitation response (*prepareConnectionRes2* function in Algorithm 1), generating a credential (*generateCredential* function in Algorithom 2) and handling proof responses (*proofStatus* function in Algorithm 3).

Algorithm 1: Snippet for *prepareConnectionRes2* function

```
function prepareConnectionRes2(connectionRes1):
    DID_S^I := connectionRes1.DID_S^I ;
    DIDDoc_S^I :=connectionRes1.DIDDoc_S^I ;
    Retrieve public key K_S^I from the DIDDoc_S^I ;
    Store DID_S^I and DIDDoc_S^I in the agent wallet ;
    Generate K_I^S and K_I^{-1|S}, DID_I^S and DIDDoc_I^S using K_I^S ;
    Prepare connectionRes2 using DID_I^S and DIDDoc_I^S ;
    return connectionRes2;
```

The ***prepareConnectionRes2()*** function in Algorithm 1 handles *connectionRes1*. The function retrieves the $\mathbf{DID}_S^I$ and corresponding $\mathbf{DIDDoc}_S^I$ from the *connectionRes1* message and stores them in the agent wallet. Then the public key $\mathbf{K}_S^I$ and the service endpoint associated with the $\mathbf{DIDDoc}_S^I$ are stored in the wallet for this specific connection. The agent wallet then generates its own public-private key $(\mathbf{K}_I^S, \mathbf{K}_I^{-1|S})$ for creating its own $\mathbf{DID}_I^S$. and corresponding $\mathbf{DIDDoc}_I^S$. Finally, the *connectionRes2* is created using the issuer $\mathbf{DID}_I^S$ and returned back to the student wallet.

Algorithm 2: Snippet for *generateCredential* function

```
function generateCredential(attributeRes):
    Retrieves existing credential definition of the issuing agent;
    requested_attributes := retrieve attribute data from attributeRes;
    VC := generateVC(credential definition, requested_attributes);
    VC_I^S := sign VC using private key K_I^{-1} corresponding to issuer public
     DID_I^pub;
    return VC_I^S;
```

The ***generateCredential()*** function (Algorithm 2) is responsible for generating the signed credential using the certificate data. The function takes *attributeRes* as input and from it, retrieves the data for the attributes to be included in the issued credential. A $\mathbf{VC}_I^S$ is generated based on the *credential definition* that the issuer

owns and signed using $\mathbf{K}_I^{-1}$ corresponding to the public $\mathbf{DID}_I^{pub}$ before sending it to the holder.

Algorithm 3: Snippet for *proofStatus* function

```
1 function proofStatus(presentation):
2     VC_I^S := presentation.VC;
3     DID_I^S := presentation.DID_I^S;
4     Retrieve public DIDDoc_I^S and corresponding public K_I^S from the
        ledger;
5     proofStatus := checkSignature(VC_I^S, K_I^S);
6     if proofStatus then
7         return TRUE;
8     else
9         return FALSE;
```

Finally, the function **proofStatus** in Algorithm 3 is employed to manage the presentation aspect. Initially, the issuer **DID** of the presented $\mathbf{VC}_I^S$ is retreived from the *presentation* message. The acquisition of the public $\mathbf{DIDDoc}_I^{pub}$ of the issuer is accomplished by accessing the ledger. The public key of the issuer is obtained from the $\mathbf{DIDDoc}_I^{pub}$, and utilized after to authenticate the signature on the $\mathbf{VC}_I^S$. If the verification of the sign is successful, the function will return the value TRUE, and if it is unsuccessful, it will return the value FALSE.

5.4.3 BracU SSI Academia Admin Login & Student Registration

Now we will go through the use case where the Admins of BracU SSI Academia can perform student registration after logging in and authenticating themselves. Table 5.4.3 presents the protocol flow of this use case.

M1	A →BA	: $[N_1, \textit{serviceReq}(\textit{url}_{s_1})]_{HTTPS}$
M2	BA →A	: $[N_2, \text{SSI Academia Login Page}]_{HTTPS}$
M3	A →BA	: $[N_2, \text{Admin Login data }]_{HTTPS}$
M4	BA →A	: $[N_3, \text{Admin Dashboard}]$
M5	A →BA	: $[N_3, \text{Add Student}]$
M6	BA →A	: $[N_4, \textit{studentInfoReq}]$
M7	A →BA	: $[N_4, \textit{studentInfoRes}]$
M8	BA →A	: $[N_5, \text{All Students Page}]_{HTTPS}$

Table 5.4.3: BracU SSI Academia Admin Login and Student Registration

i. Authorized administrators can log into the system by accessing the service url for SSI Academia and entering their login credential through step M1, M2, M3 in Table 5.4.3.

ii. Once the login is successful, SSI Academia returns a Dashboard in step M4 with available options that the BracU admin can perform along with a **JW Token**. Figure 5.4 shows the Dashboard.

iii. Admin chooses "Add New Student" option in step M5.

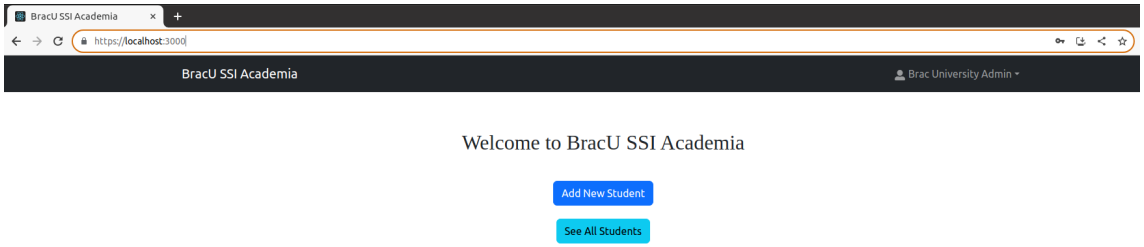


Figure 5.4: BracU SSI Academia Admin Dashboard

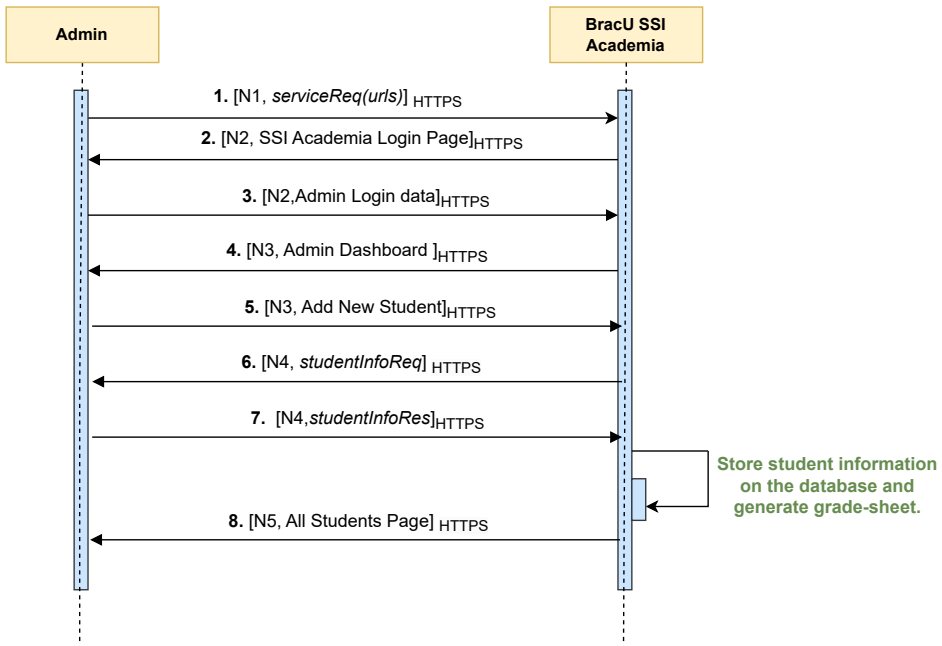


Figure 5.5: BracU SSI Academia Admin Interactions Sequence Diagram

iv. The issuer website returns a form to the admin requesting him/her to enter student information, account credentials and grade sheet for a particular student through *studentInfoReq* in step M6.

v. Admin responds with *studentInfoRes* by entering all the necessary data in the form and clicking the submit button in step M7.

Figure 5.5 represents the procedure using sequence diagram. Moreover, Figure 5.4 and Figure 5.6 shows BracU SSI Academia Admin Dashboard and Student list screenshots respectively. Lastly, Figure 5.7 shows the "Add a New Student Page".

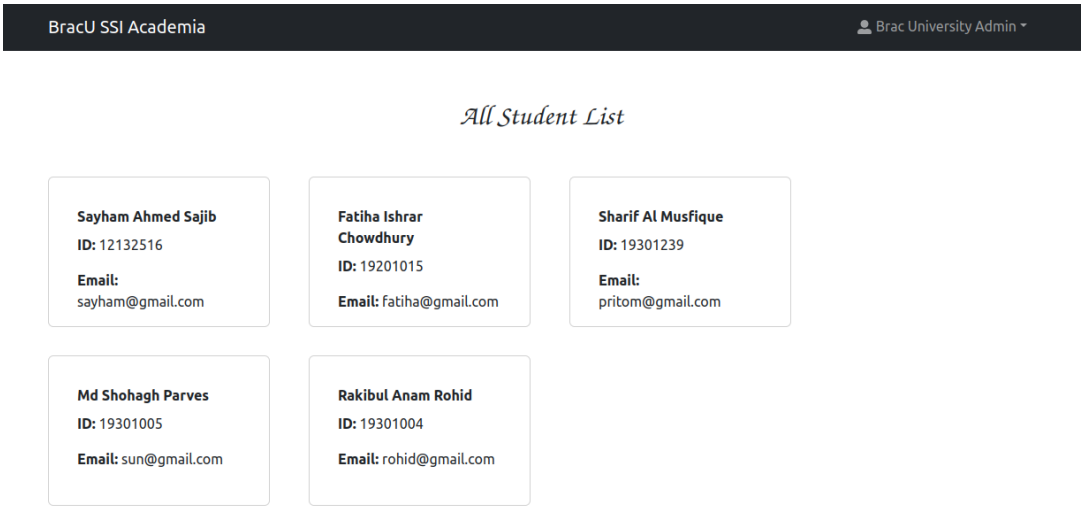


Figure 5.6: BracU SSI Academia Student List

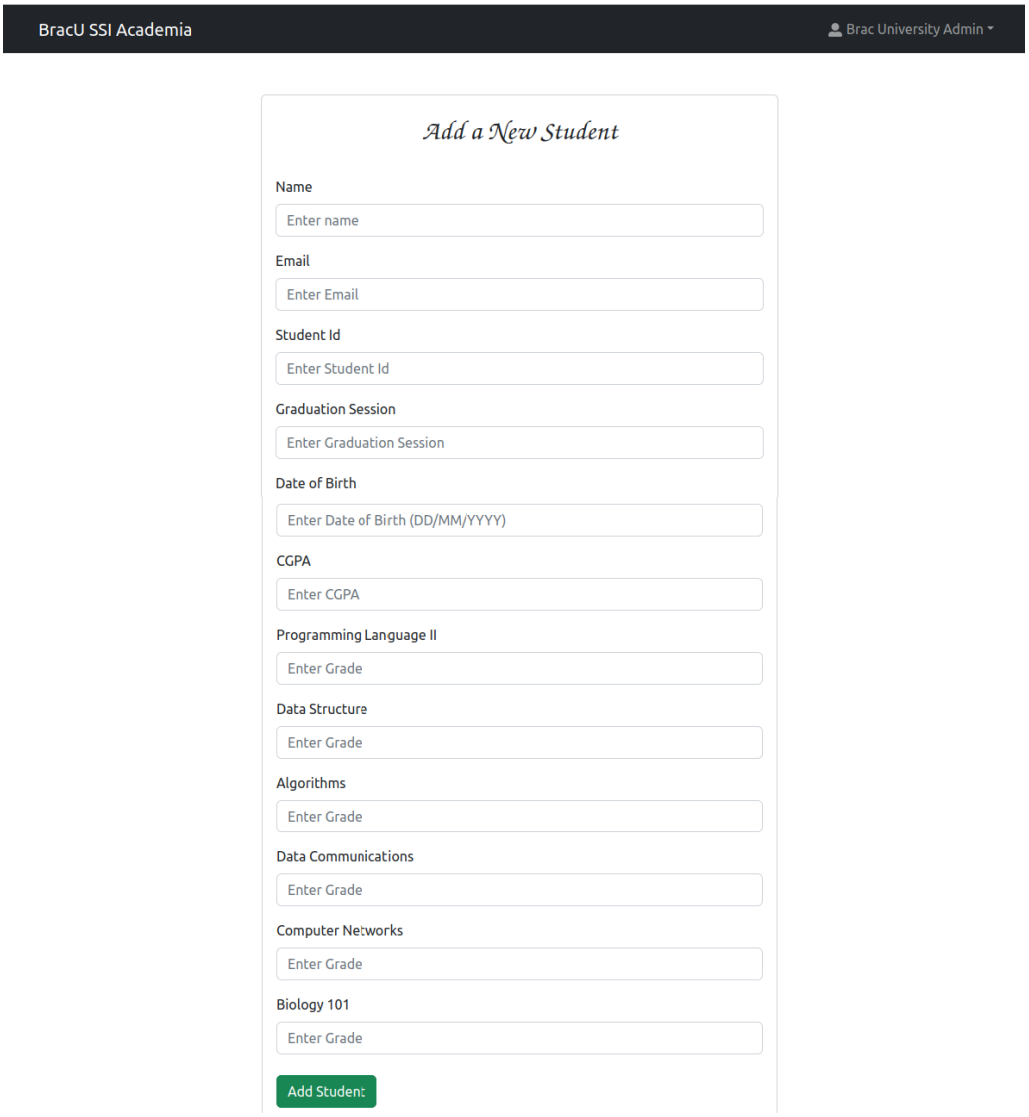


Figure 5.7: BracU SSI Academia Student Adding Form

5.4.4 BracU SSI Academia Student Interactions

Student login and SSI connection:

In this use case, we discuss the various functionalities the students can avail from the BracU SSI Academia such as creating a SSI connection, viewing their gradesheet, acquiring a credential or certificate etc. The protocol flow for this use case is depicted in Table 5.4.4. Figure 5.9 represents the protocol flow for this use case.

M1	S →I	: $[N_1, \textit{serviceRequest}(\textit{url}_{s_1})]_{HTTPS}$
M2	I →S	: $[N_2, \text{Login Page}]_{HTTPS}$
M3	S →I	: $[N_2, \text{Login Data}]_{HTTPS}$
M4	I →S	: $[N_3, \text{Student Dashboard}]_{HTTPS}$
M5	S →I	: $[N_3, \text{Create SSI Connection}]_{HTTPS}$
M6	I →S	: $[N_4, \textit{inviteReq}]_{HTTPS}$
M7	S →M	: $[\text{I}, N_4, \textit{connectionRes1}]$
M8	M →I	: $[N_4, \textit{connectionRes1}]$
M9	I →M	: $[\text{S}, N_4, \textit{connectionRes2}]$
M10	M →S	: $[N_4, \textit{connectionRes2}]$
M11	I →S	: $[N_5, \text{Student Dashboard}]_{HTTPS}$
M12	S →I	: $[N_5, \text{View Grade-sheet}]_{HTTPS}$
M13	I →S	: $[N_6, \text{Grade-sheet Page}]_{HTTPS}$

Table 5.4.4: BracU SSI Academia Student Login and SSI Connection

i. The student utilizes one of the url_s of the issuer's website to access the login page. Student responds with login data to BracU SSI Academia and after successfully validating the login data, student is forwarded to the Dashboard with a **JW Token** for authorization purposes where the student is shown the available options he can avail in BracU SSI Academia though step M1 to M4 in Table 5.4.4.

ii. Depicted in Figure 5.8, Student chooses "Create SSI Connection" in step M5.

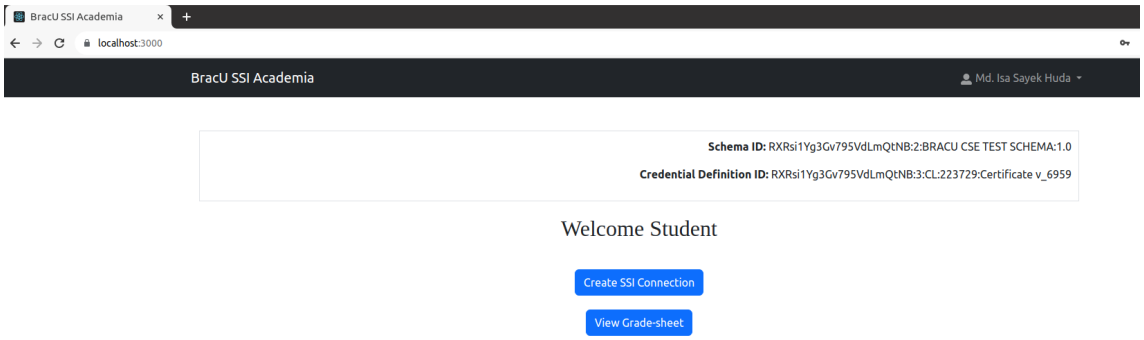


Figure 5.8: BracU SSI Academia Student Dashboard

iii. The issuer agent sends an out-of-band invitation following the *Aries RFC 0160 - Connection Protocol* where the sender gives information to the receiver about

the kind of **DIDComm Protocol** response messages it can handle, a *recipientKeys* that the sender securely stores and how to deliver the response (i.e. a service endpoint). The *invitation url* is parsed into a QR code and display it to the screen for scanning in step M6 as shown in Figure 5.10.

iv. After scanning the QR code i.e. receiving the out-of-band message, the student's wallet utilizes *prepareConnectionRes1()* to initiate the DID exchange. The holders wallet generates a public-private key pair ($\mathbf{K}_S^I$ **and** $\mathbf{K}_S^{-1|I}$), a new $\mathbf{DID}_S^I$ and $\mathbf{DIDDoc}_S^I$ using the generated public key for the connection being established and stores them in the connection record for this particular student. The Request message is structured and encoded leveraging the *recipientKeys* from the received invitation. After the encoding, the message is dispatched to the *serviceEndpoint* through the Mediator. Subsequent to the successful delivery of the *connectionRes1* message to the issuer/verifier agent, a state transition happens in the student's wallet altering its state to requested, step M7 and M8 in Table 5.4.4.

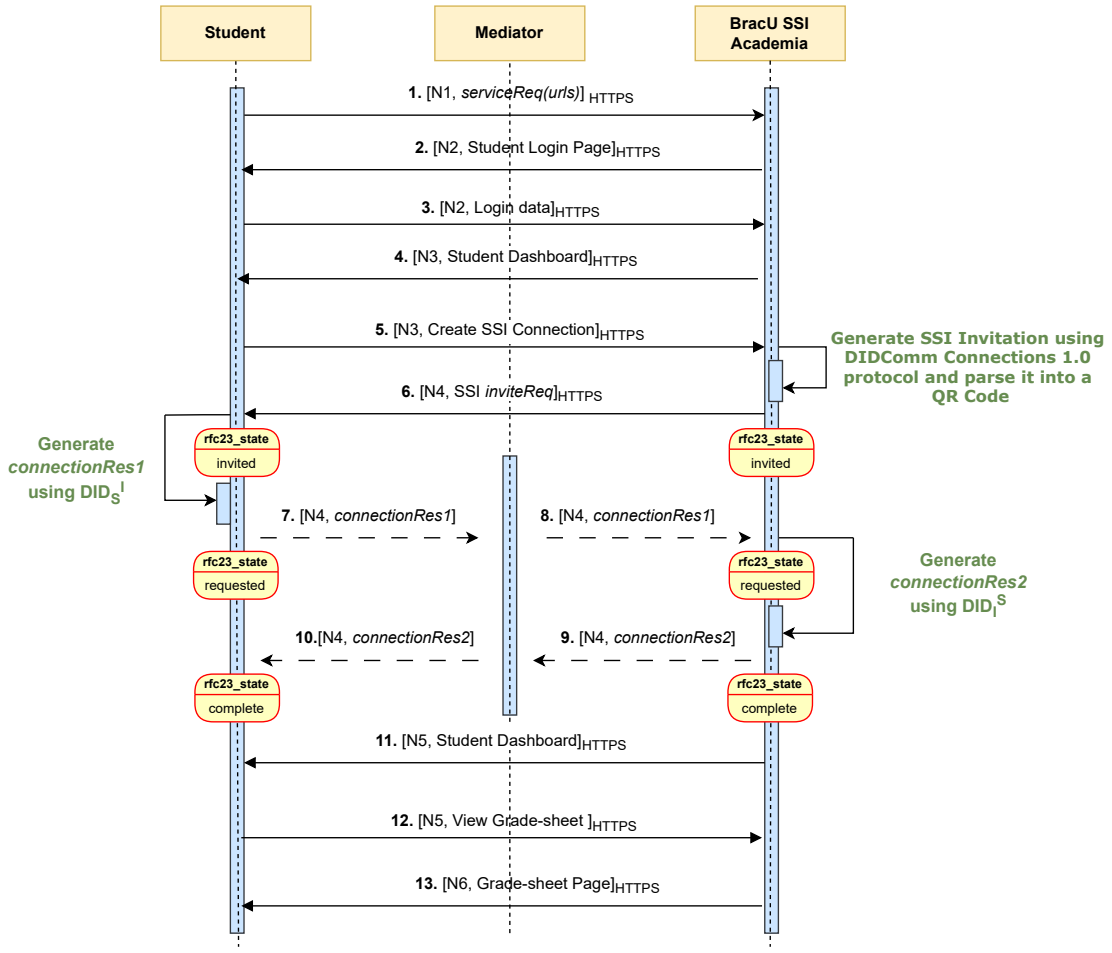


Figure 5.9: BracU SSI Academia Student Login and SSI Connection Sequence Diagram

v. The issuer's webhook is constantly listening for messages from the student and when it gets the *connectionRes1* message, it calls the *prepareConnectionRes2()*

function after storing the $\mathbf{DID}_S^I$ and $\mathbf{DIDDoc}_S^I$ in its connection record. The function operates similar as the *prepareConnectionRes1()* function and generates a *connectionRes2* message using $\mathbf{DID}_I^S$ of BracU SSI Academia and corresponding $\mathbf{DIDDoc}_I^S$ and sends it to the student wallet endpoint through the Mediator using *DIDComm envelop* through step M9 and step M10.

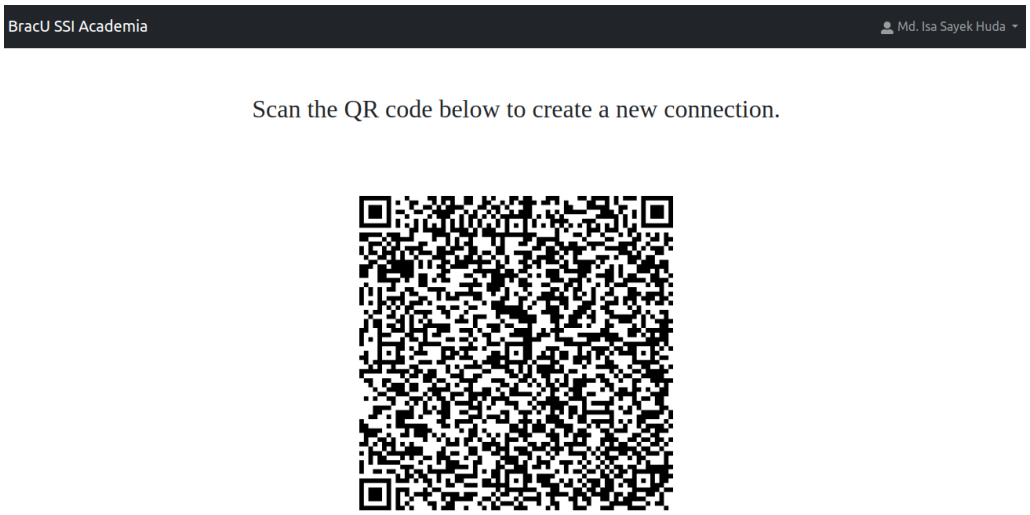


Figure 5.10: BracU SSI Academia QR code generation

vi. Students wallet retrieves the issuer $\mathbf{DID}_I^S$ and $\mathbf{DIDDoc}_I^S$ from the *connectionRes2* message. Along with the data from student $\mathbf{DID}_S^I$, the wallet saves the information of the issuer $\mathbf{DID}_I^S$ in its record as a new connection. Upon completion, the connection is established and both the student wallet and issuer agent transitions into the complete state.

vii. After an SSI connection is established with the students mobile wallet, the student is forwarded to the Student Dashboard page of SSI Academia in step M11 where the student can select from "View grade sheet" or "Issue certificate" option as shown in Figure 5.11.

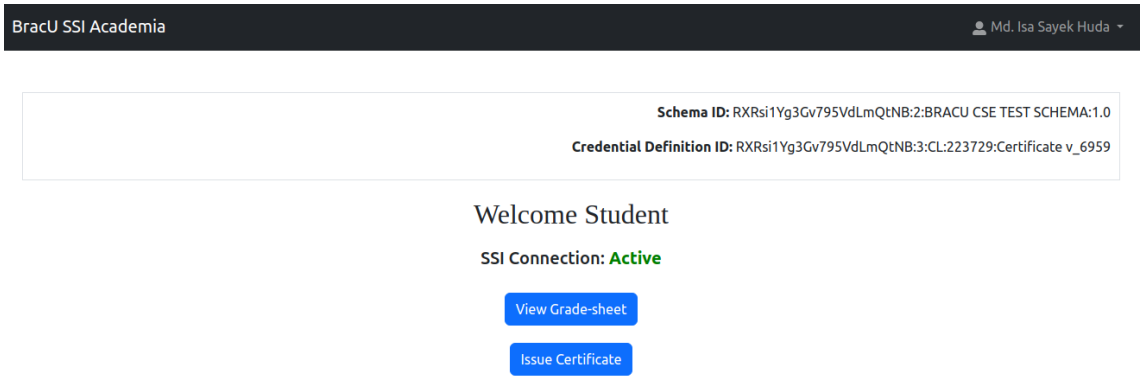


Figure 5.11: BracU SSI Academia Student Dashboard After SSI Connection

viii. On the screen shown in Figure 5.11, the student chooses "View Grade-sheet" option at step M12.

ix. SSI Academia returns a page containing information about the student grade sheet requested in step M13 which is visible in Figure 5.12.

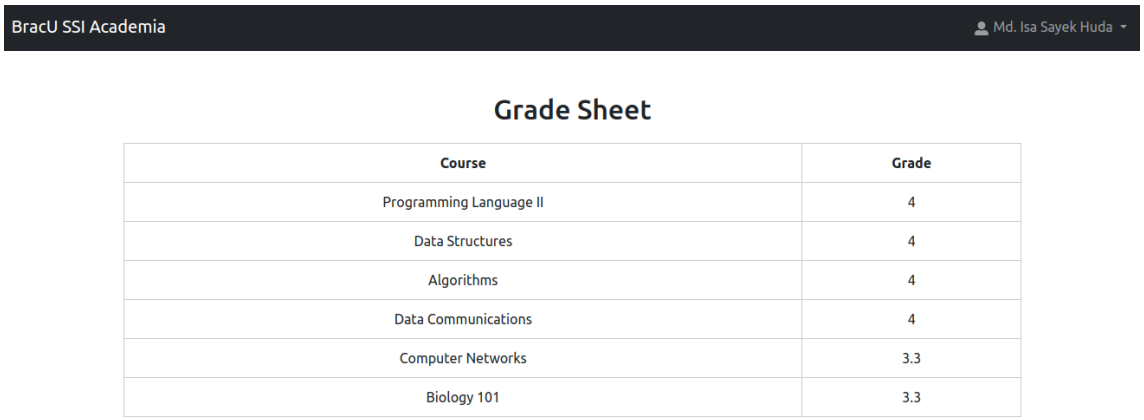


Figure 5.12: BracU SSI Academia Student Dashboard After SSI Connection

SSI Academia Issuing Certificate:

This part focuses on the use case wherein a student obtains a credential from the BracU SSI Academia. The protocol flow is illustrated in Table 5.4.5 for this use case.

M1	S →I	:	$[N_1,$ Issue Certificate$]_{HTTPS}$
M2	I →S	:	$[N_2,$ *attributeReq*$]_{HTTPS}$
M3	S →I	:	$[N_2,$ *credentialReq, attributeRes*$]_{HTTPS}$
M4	I→M	:	$[\mathrm{S}, \{\{N_2,$ *credentialRes*$\}_{K_I^{-1}}\}_{K_S^I}]$
M5	M →S	:	$[\{\{N_2,$ *credentialRes*$\}_{K_I^{-1}}\}_{K_S^I}]$
M6	S →M	:	$[\mathrm{I}, N_2,$ *credentialAck*$]$
M7	M →I	:	$[N_2,$ *credentialAck*$]$
M8	I →S	:	$[N_3,$ VC Generation Successful Page$]_{HTTPS}$

Table 5.4.5: Issuing Certificates Protocol

As depicted in Table 5.4.5 as well as the sequence diagram in Figure 5.13, from the student dashboard:

i. In the first step (M1) of Table 5.4.5, the student selects the option to "Issue Certificate" as Figure 5.11 depicts.

ii. During issuing credentials, all the messages are sent to the student wallet using the end-to-end encrypted channel that was previously established using **DID-Comm Messaging Protocol**. The procedure adheres to the *Aries RFC 0036: Issue Credential Protocol 1.0* [34] and starts with presenting the student with a page showing the information about the grade sheet from the database and requests for attribute data that will be used to issue the certificate (step M2) as shown in Figure 5.14.

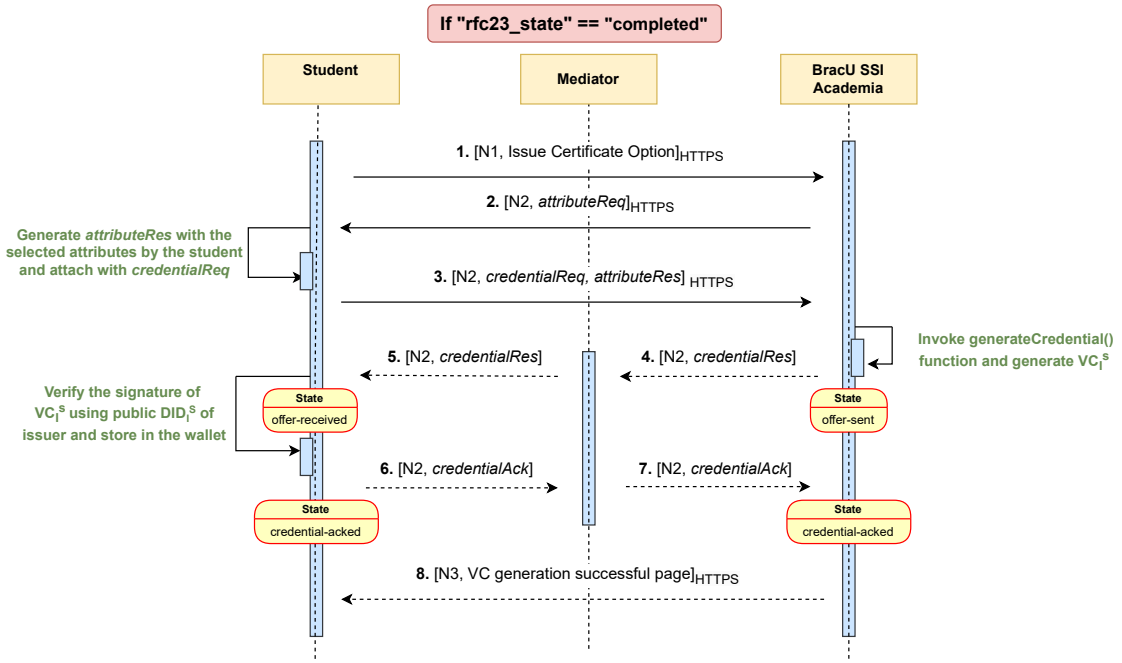


Figure 5.13: SSI Academia Issuing Certificate Sequence Diagram

iii. Students can control which course results from the grade sheet will be included with the credential and generate *attributeRes* which contains the attributes student selected. Then *attributeRes* is then attached with *credentialReq* to make a request for **VC** using the information received from *attributeRes*, step M3 in Table 5.4.5.

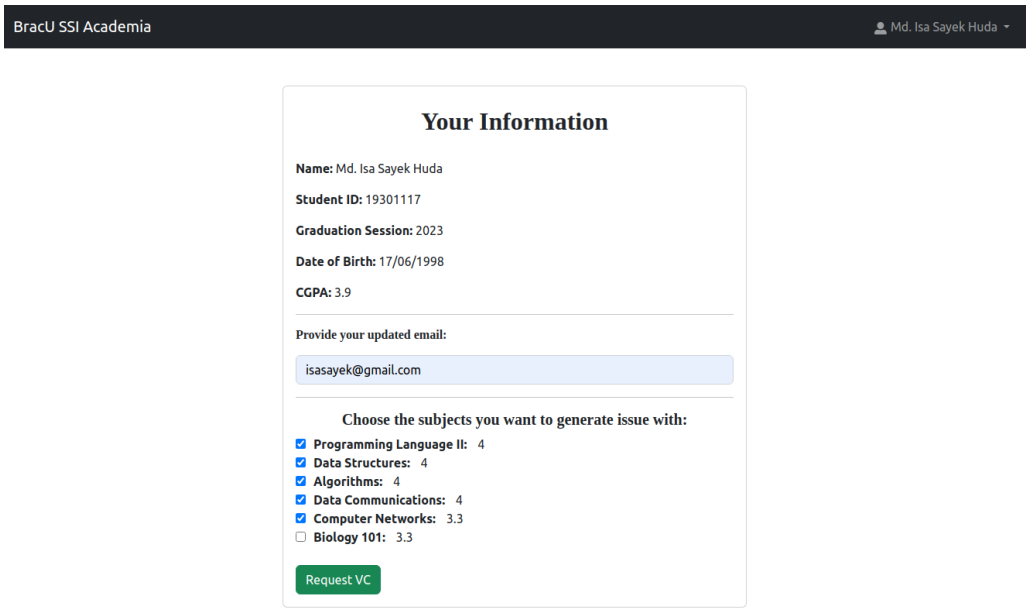


Figure 5.14: Issuing Certificate Screenshots

iv. The issuer website invokes *generateCredential()* and generates the requested **VC** using the data received from *attributeRes*. The BracU SSI agent then utilizes one of its REST API to send an *credentialRes* message to the holder describing

the credential they intend to offer. The message contains the $\mathbf{VC}_I^S$ that was signed using the public $\mathbf{DID}_I^{pub}$ and it is relayed through the mediator agent, represented in the Table 5.4.5 by step M4 and M5.

v. The student gets a prompt in their mobile wallet about the incoming certificate offer from BracU agent and decides whether or not to accept the credential and store it in the wallet as depicted in Figure 5.15. If the student accepts, the mobile wallet retrieves the public $\mathbf{DIDDoc}_I^{pub}$ of the issuer from the ledger and verifies the sign on the credential using the public key $\mathbf{K}_I^{pub}$ retrieved from the **DIDDoc**. If they match, the certificate credential is stored in the holders wallet. Upon completion, the student's wallet generates an acknowledgment by sending an ack message to the issuer agent through the mediator in step M6 and M7.

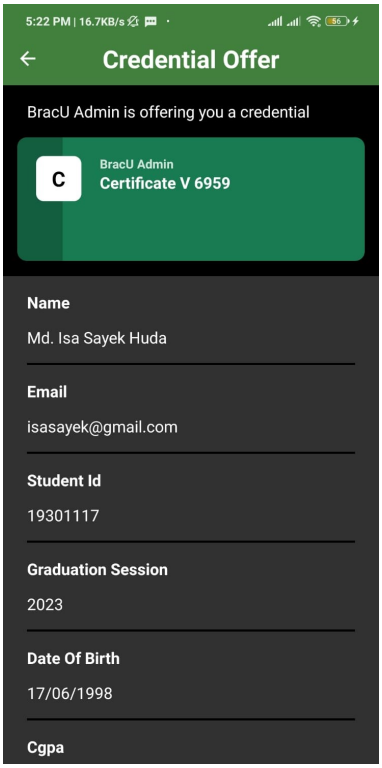


Figure 5.15: Screenshot of Credential Offer in Mobile Wallet

vi. Once the state of both the holder and verifier agent changes to credential-acked, the student is forwarded to a confirmation page in step M8 depicted in Figure 5.16.

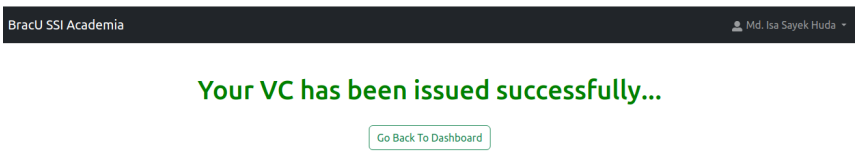


Figure 5.16: Certificate Generation Screenshot

5.4.5 SSI Recruitment Portal Recruiter Login and Job Post

This use case refers to the SSI Recruitment Portal, which involves the process of authorized recruiters or admins logging in to the system in order to post job listings. The protocol flow for this particular scenario is illustrated in Table 5.4.6. The representation of the protocol flow is also depicted by a sequence diagram, as illustrated in Figure 5.17.

M1	R →RP	: $[N_1, \textit{serviceReq}(\textit{url}_{s_1})]_{HTTPS}$
M2	RP →R	: $[N_2, \text{SSI Portal Login Page}]_{HTTPS}$
M3	R →RP	: $[N_2, \text{Recruiter Login data }]_{HTTPS}$
M4	RP →R	: $[N_3, \text{Recruiter Dashboard}]_{HTTPS}$
M5	R →RP	: $[N_3, \text{Add Jobs}]_{HTTPS}$
M6	RP →R	: $[N_4, \textit{jobsDetailsReq}]_{HTTPS}$
M7	R →RP	: $[N_4, \textit{studentInfoRes}]_{HTTPS}$
M8	RP →R	: $[N_5, \text{Added Students List Page}]_{HTTPS}$

Table 5.4.6: SSI Recruitment Portal Recruiter Login and Job Post

Above protocol is depicted by a sequence diagram in Figure 5.17 and starts with:

i. Authorized recruiters or organizational representatives can create accounts and log into the system by accessing the service *url* for SSI Portal and entering their login credential and receiving a **JW Token** through step M1, M2 and M3 in Table 5.4.6.

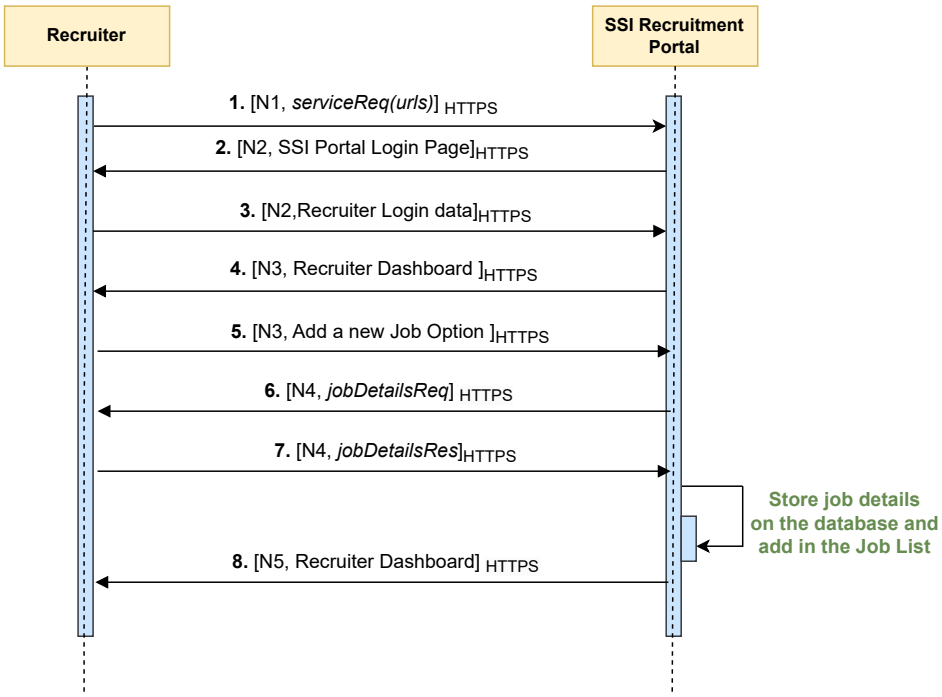


Figure 5.17: SSI Recruitment Portal Recruiter Sequence Diagram

ii. Once the login is successful, SSI Portal returns a Dashboard where the recruiter selects "Add Job" option through step M4 and M5. The Dashboard is shown in Figure 5.18.

iii. The issuer website returns a page to the recruiter requesting for details about the job vacancy and requirements and sends *jobDetailsReq* in step M6.

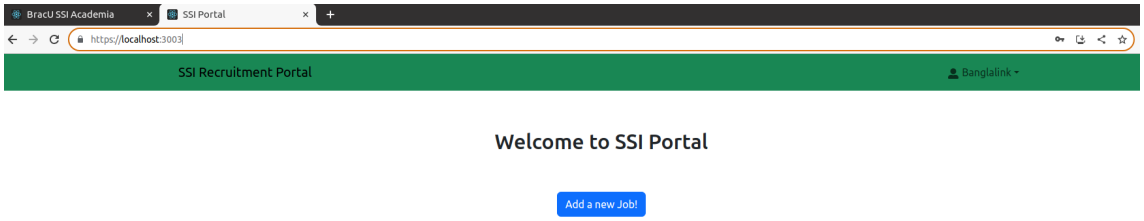


Figure 5.18: SSI Recruitment Portal Admin Dashboard

iv. The recruiter defines the verifiable presentation format which the applicants will need to provide. Organizations may require applicants to present proof about particular course results instead of the full grade sheet. In that case, recruiters can select which course results the applicants need to include as credentials in the verifiable presentation along with the job details as represented in Figure 5.19; thus enabling *Selective Disclosure* for the *proof-request*. The recruiter responds with *jobDetailsRes* with all the necessary data about the *proof-request* and job details by clicking "Submit" button in step M7 as illustrated in Figure 5.19.

v. After successfully posting the job, the recruiter is forwarded to the Dashboard in step M8.

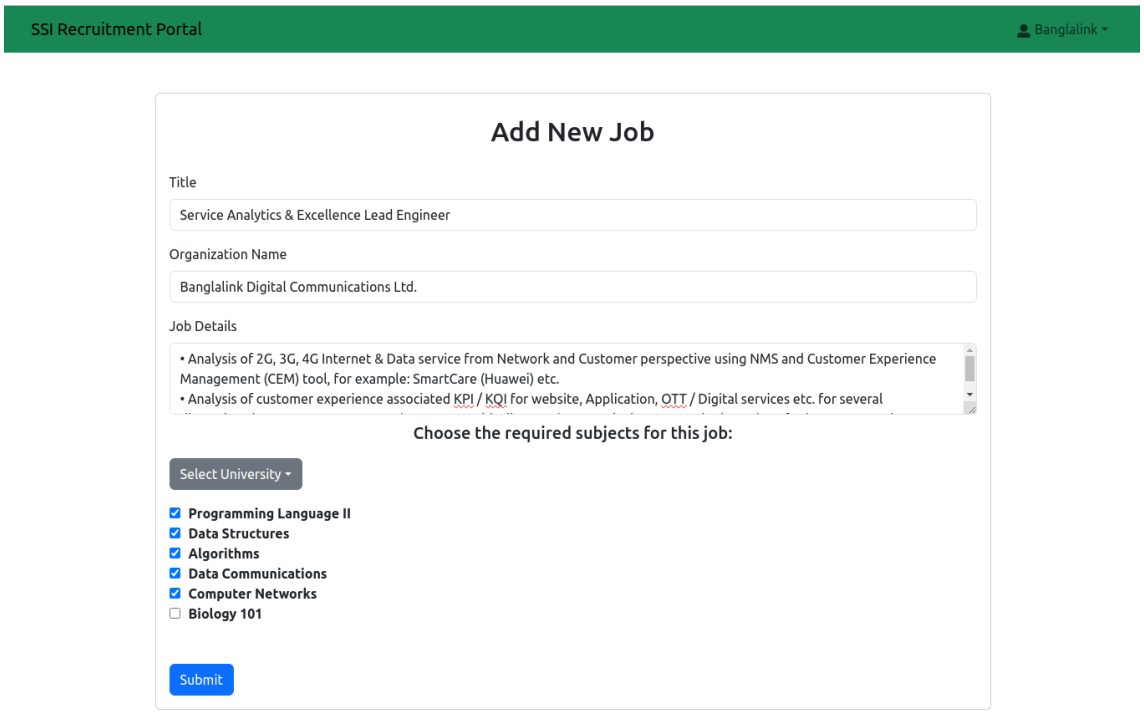


Figure 5.19: SSI Recruitment Portal Admin Dashboard

5.4.6 SSI Recruitment Portal Applicant Interactions

Applicant login and SSI connection:

This use case examines the features available to students or job applicants on the SSI Recruitment Portal, including the option of using the student dashboard. The protocol flow for this specific case is illustrated in Table 5.4.7. The representation of is also depicted in Figure 5.21 by a sequence diagram.

M1	S →V	: $[N_1, \textit{serviceReq}(\textit{url}_{s_1})]_{HTTPS}$
M2	V→S	: $[N_2,$ SSI Recruitment Portal Login Page$]_{HTTPS}$
M3	S →V	: $[N_2,$ Student Dashboard$]_{HTTPS}$
M4	V →S	: $[N_3,$ Student Dashboard$]_{HTTPS}$
M5	S →V	: $[N_3,$ Create SSI Connection$]_{HTTPS}$
M6	V →S	: $[N_4, \textit{inviteReq}]_{HTTPS}$
M7	S →M	: [V, N_4, *connectionRes1*]
M8	M →V	: [N_4, *connectionRes1*]
M9	V →M	: [S, N_4, *connectionRes2*]
M10	M →S	: [N_4, *connectionRes2*]
M11	V →S	: $[N_5,$ Student Dashboard$]_{HTTPS}$

Table 5.4.7: Applicant login and SSI Connection

i. The applicant utilizes the *urls* of the SSI Portal to access the login page. The applicant selects "Student Dashboard" option and the applicant is forwarded to the Dashboard where the her or she is shown with available options in SSI Portal through step M1 to M4 in Table 5.4.7 and in Figure 5.22.

ii. Applicant chooses "Create SSI connection" option in step M5 as depicted in Figure 5.20.

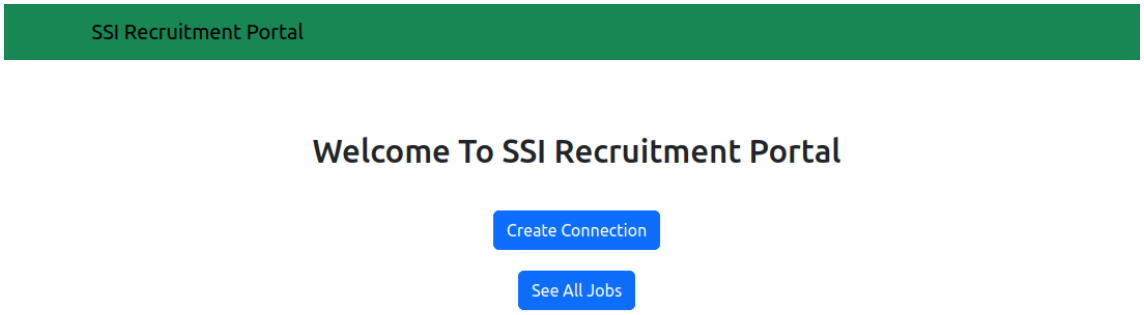


Figure 5.20: SSI Recruitment Portal Student Dashboard

iii. The connection protocol flow is similar as before with the issuer agent. SSI Portal displays the invitation url using a **QR** code. The applicant scans it using their mobile wallet and initiates a DID exchange between the mobile agent and the SSI Portal agent following the **DIDComm Connection Protocol**; steps M6 to M10 in Table 5.4.7.

Here in Figure 5.21, the protocol flow is illustrated through a sequence diagram:

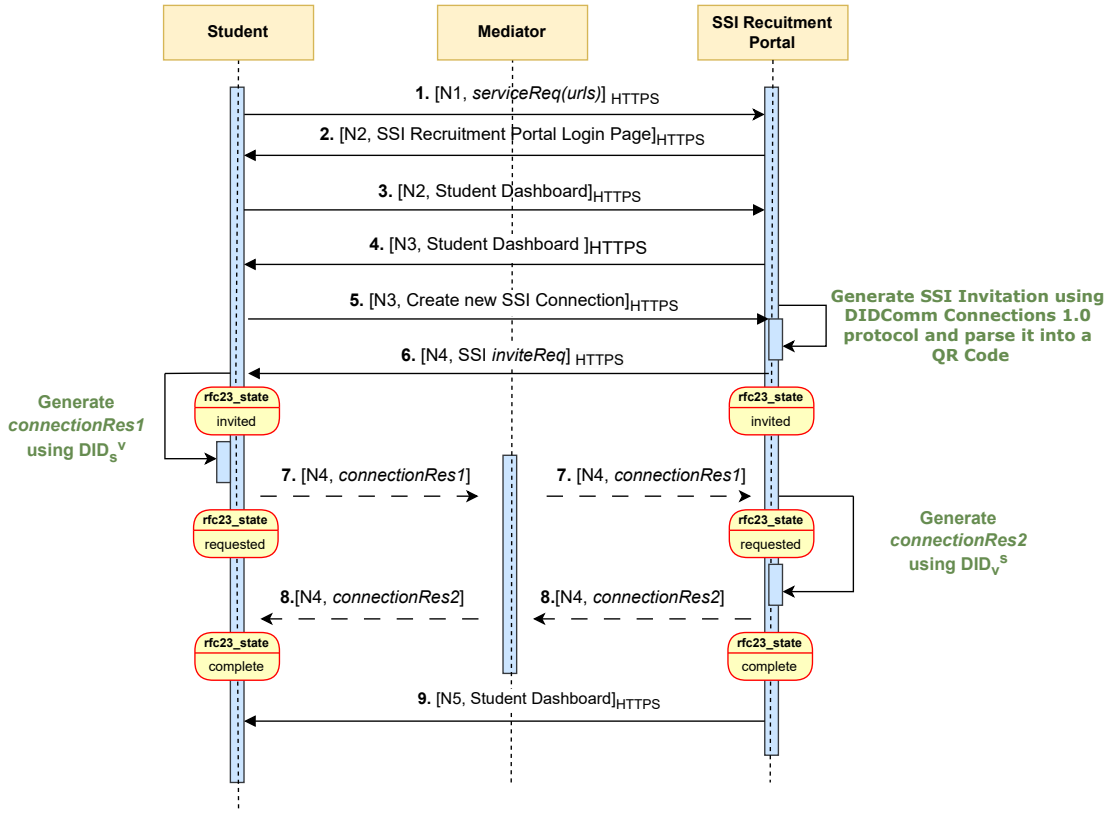


Figure 5.21: SSI Recruitment Portal SSI Connection Sequence Diagram

iv. Once the connection is successful, the applicant is forwarded to the Dashboard in step M11 and now he/she is eligible to present-proof and apply for a job. Without making a SSI connection, the applicant could only see the Job Posts but cannot apply for them as shown in Figure 5.22.

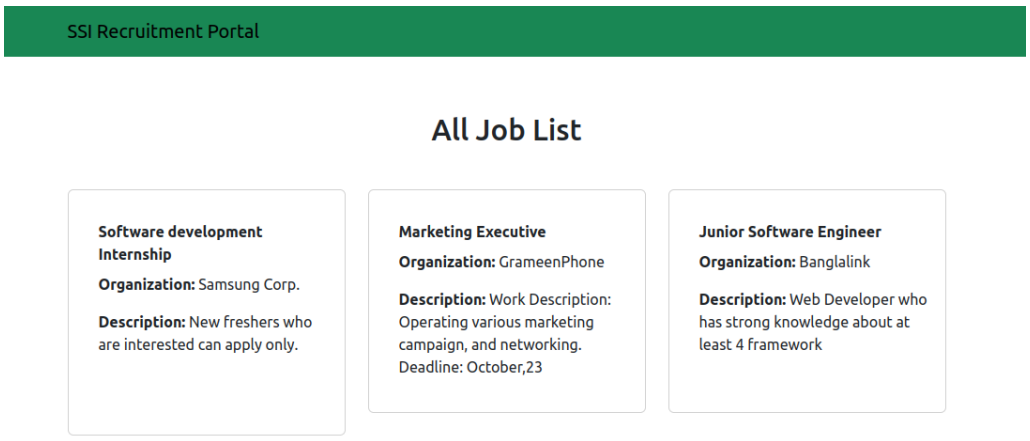


Figure 5.22: View Job List without SSI Connection

Present Proof of Certificates:

Lastly, we define the protocol flow in Figure 5.4.8 for applicants presenting the proof of certificate to SSI Recruitment Portal. A sequence diagram is also generated in Figure 5.24 for a better understanding.

M1	S →V	: $[N_1$, See Job List$]_{HTTPS}$
M2	V →S	: $[N_2$, Job List Page$]_{HTTPS}$
M3	S →V	: $[N_2$, Apply For This Job$]_{HTTPS}$
M4	V →M	: $[$S, $\{N_3, presentationReq\}_{K_S^V}]$
M5	M →S	: $[\{N_3, presentationReq\}_{K_S^V}]$
M6	S →M	: $[$V, $\{N_3, \{presentationRes\}_{K_I^{-1}}\}_{K_V^S}]$
M7	M →V	: $[\{N_3, \{presentationRes\}_{K_I^{-1}}\}_{K_V^S}]$
M8	V →S	: $[N_4$, Submit CV and Job Application$]_{HTTPS}$

Table 5.4.8: Present Proof of Certificates Protocol

In the applicant dashboard:

i. The applicant selects "See Job List" option in step M1 in Table 5.4.8.

ii. SSI Portal returns a Job List page where a list of all available jobs are shown. Applicants may submit an application for any Job Post as long as they meet the academic qualifications specified by the recruiter and produce a verifiable presentation of their certificates. To apply for a job, the applicant selects the "Apply For This Job" button from one of the job cards, step M2 and M3 in the Table 5.4.8 and illustrated in Figure 5.23.

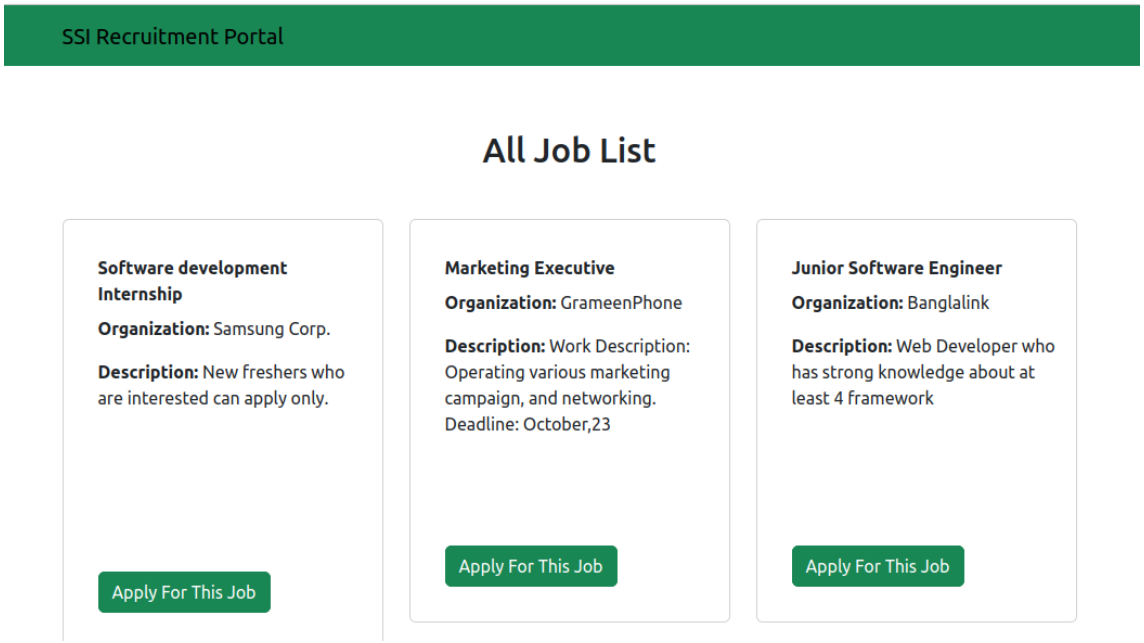


Figure 5.23: SSI Recruitment Portal Job List after SSI Connection

iii. *Aries RFC 0037: Present Proof Protocol 1.0* is then initiated by the verifier agent in sending a *presentationReq* message to the holder (step M4 and M5 in

Table 5.4.8) through the mediator agent requesting a presentation to the student wallet via the SSI Portal's REST API, specifying the attributes and predicates the Verifier wants the student to present [31]. The applicant is forwarded to the waiting screen as shown in 5.25 while the mobile wallet gets a notification about an incoming presentation request.

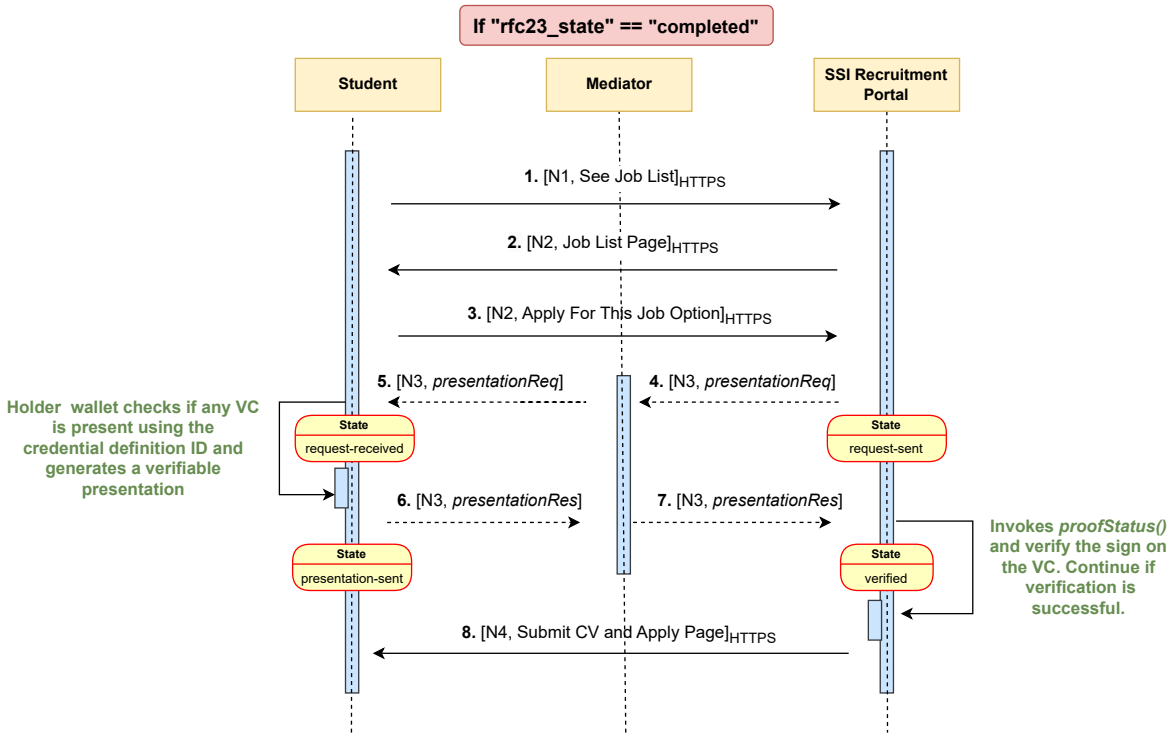


Figure 5.24: Present Proof of Certificates Protocol Sequence Diagram

iv. The student wallet processes the *presentationReq* message to get the requested credential definition ID *(cred_def_id)* and searches for any **VC** matching the *cred_def_id* stored in the mobile wallet. If found then the wallet shows the **VC** matching the *cred_def_id* to the student and ask for approval to generate the verifiable presentation, illustrated in Figure 5.26. If the student approves, the wallet prepares a presentation message using the verifiable presentation created and returns it to the verifier agent, represented in step M6 and M7 in Table 5.4.8.

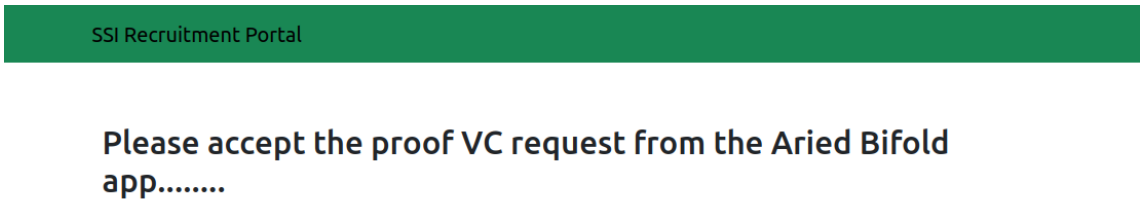


Figure 5.25: Proof Request Waiting Screen

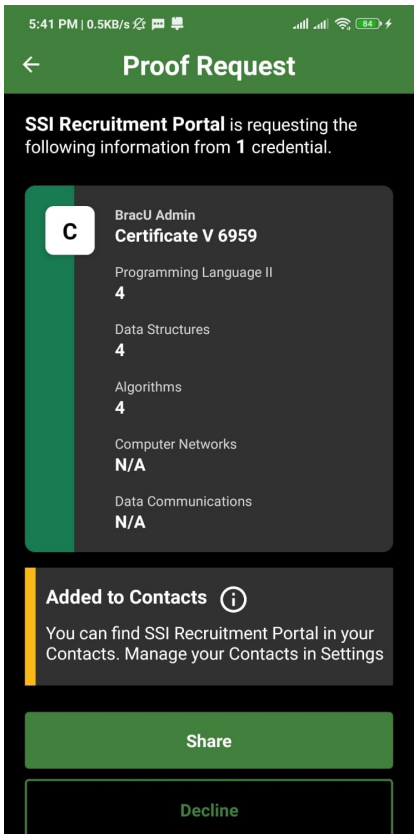


Figure 5.26: Present Proof Prompt in Mobile Wallet

v. All communication in step M6 and M7 are relayed through the mediator agent. The verifier agent receives the VC and invokes *proofStatus()* function to validate the signature of the presentation. If validated successfully, the *proofStatus()* function returns **TRUE** and the verifier site forwards the student to the "Submit CV and Job Application Page" through step M8 as shown in Figure 5.27.

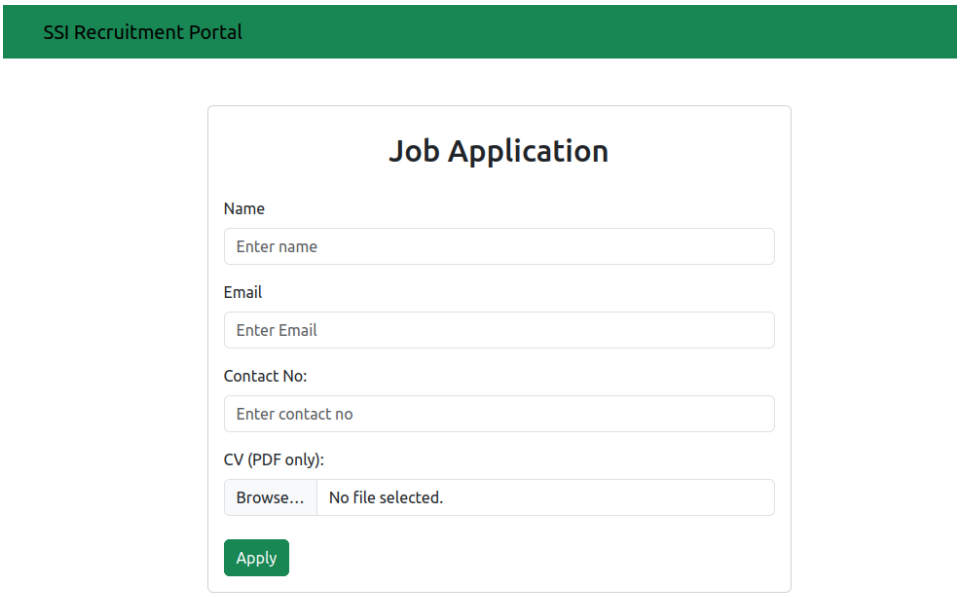


Figure 5.27: Job Application Page upon successfully presenting proof

Chapter 6

Discussion

6.1 Requirements Analysis

6.1.1 Functional Requirements:

This system meets the functional requirements mentioned. The DID is a unique string created from 32 bits seed, which facilitates it as an Identifier thereby achieving **F-1**. **F-2** is attained through the usage of mobile wallets; where data is stored without the need of a middleman. Through the implementation of Ursa hyperledger and the DIDcomm protocol, the concern of **F-3** is addressed. **F-4** is achieved because the portal of our system allows login of the certified admin and only they can add the students grade sheet as an input to create schema; and only they can create schema and credential definition in the ledger because they have the endorser role.

6.1.2 Security Requirements:

S-1 is satisfied because only BracU Academia Admin and the Recruiters of SSI portal are authorized to carry out the verification process. Before the VC is sent, it is encrypted with the recipient's public key, thereby tackling **S-2**. The security requirement **S-3** is met as the digital signature is implemented by signing the credential with the private key of the issuer that corresponds to the issuer public DIDDoc. Through the DIDcomm protocol, end-to-end encrypted channels are created before data is exchanged, hence **S-4** is addressed. In this research, we are not focused on counteracting **S-5** as it is out of our research scope. Admins, recruiters and students are authenticated and authorized using hash passwords, user roles and JW Tokens, thus satisfying **S-6**. To satisfy **S-7**, we have made heavy use of nonces in all interactions.

6.1.3 Privacy Requirements:

In this work, the user has complete control over their wallet. A user gets to select which attributes of a VC he wants to keep in his wallet through the process of selective disclosure. A user has the right to decline any certificate issued by the issuer because the user's approval needs to be taken while releasing the VC hence accomplishing **P-1** and **P-2**.

6.2 Research Objectives Analysis:

The system thoroughly fulfills the research objectives in the following ways:

RO1: This objective is attained because all the parties involved communicate as per the protocol flow illustrated; and follow the same protocol. The issuers public DID is stored in ledger which ensures data integrity so the verifiers can easily identify the issuer thus upholding the trust triangle.

RO2: Students have complete control over their data and nothing will be done with their data against their consent because the students data is stored in their wallet. Only the student has access to his/her wallet, and gets to decide with whom he wants to share data with ensuring a smooth verification process. The student also gets to decide whether he wants a certain VC from a certain issuer or not.

RO3: This implementation has assured portability because the verifiable credentials (VC) are stored in a mobile wallet; thus free to move anywhere. As the whole ideology revolves around a decentralized system, people are not tied down to a single platform. Moreover, different agents are free to communicate with the same or different ledgers as they are agnostic.

RO4: The system will allow different schemas to be generated by different issuers which can be moved between platforms, allowing multiple verification to be done on a worldwide scale. Multiple VC can be issued by the issuer and one VC can also be verified by multiple verifiers.

RO5: Through the method of selective disclosure, our implementation ensures that only the information required and relevant, is shared thus ensuring the issue of oversharing.

RO6: This objective is achieved through the usage of Hyperledger Ursa; which provides cryptographic notations like personalized signatures and public/private keys. The communication through DIDcomm protocol addresses end to end communication. The usage of Indie ledger also plays a crucial role in information confidentiality.

6.3 Comparative Analysis:

In this section, We have compared some existing works with our current proposal. Table 6.3.1 presents an analysis of the current work with the previous research works against a number of criteria, where the "✓" symbol signifies whether the respective work fulfilled a criterion and the "✗" symbol shows whether the specific criterion was not satisfied. SSI, VC, DID, Threat Model, Authenticated Interaction, Encrypted Communication, Time Efficiency, Selective Disclosure, Distributed Ledger Technology, Interoperability and Mobile Wallet are the criteria considered.

Our objective is to meet all of the requirements (having a ✓symbol). As shown in our table, the current effort meets all of the defined requirements without the use

of a third party and so outperforms the present solutions.

Criteria	**Our Work**	Ghazali (2018) [39]	Han et al. (2018) [5]	Huang (2020) [14]	Litoussi et al. (2022) [22]	Ferdous et al. (2022) [25]	Raman (2013) [2]
SSI	✓	✗	✗	✗	✗	✓	✗
VC	✓	✗	✗	✗	✗	✓	✗
DID	✓	✗	✗	✗	✗	✓	✗
Threat Model	✓	✗	✗	✗	✗	✓	✗
Authenticated Interaction	✓	✓	✓	✓	✓	✓	✓
Encrypted Communication	✓	✓	✓	✓	✓	✓	✓
Time Efficient	✓	✗	✓	✗	✓	✗	✓
Selective Disclosure	✓	✗	✗	✗	✗	✗	✗
Distributed Ledger Technology	✓	✓	✓	✓	✓	✓	✗
Portability & Interoperability	✓	✗	✗	✗	✗	✓	✗
Mobile wallet	✓	✗	✗	✗	✗	✓	✗

Table 6.3.1: Comparative Analysis of the System against some existing works.

6.4 Advantages & Limitations

The academic certificate verification process using SSI provides a number of advantages which are discussed next:

1. **User Autonomy:** SSI empowers individuals with full authority over their digital identities. Individuals are capable of establishing, managing, and exchanging their identification data without contacting or relying on central administrations or issuer organizations. Individuals currently possess the ability to exercise authority over their privacy and personal data.

2. **Privacy:** The architectural design of SSI inherently puts an emphasis on privacy. Individuals have the option to selectively disclose aspects of their data that are deemed essential for a particular discourse, while intentionally excluding any more personal details.

3. **Security:** SSI leverages cryptographic techniques to ensure the security and integrity of identity data. Data is stored and verified using public-private key pairs, making it extremely difficult for malicious actors to tamper with or falsify data. This enhances the overall security of digital interactions.

4. **Reduced Friction and Correlation:** The adoption of self-sovereign identification allows individuals to employ unique identifiers (i.e. DIDs) for various online engagements, hence preventing the linkage of their activities across various contexts. This phenomenon leads to a reduction in the correlation and friction between data points.

5. **Automated Verification Process:** The automated verification process is facilitated by using the technique of self-sovereign identification (SSI), which employs distributed identifying systems. By eliminating the need for human inspections and reliance on third-party intermediaries or even the issuer organizations, this automated verification method enhances security, privacy, efficiency and user experience.

Unfortunately, the current implementation of academic certification verification system has certain drawbacks, which are listed below:

1. In our situation, the wallet is crucial to the user. Because all of the VCs and connections are kept in the wallet, therefore, a safe backup method is essential.

2. SSI systems are designed to provide users more control and experience with their data. However, the usability of most solutions, particularly for non-technical users, may be difficult or unclear, resulting in adoption obstacles.

3. This system has to rely on external services of Aries, such as distributed ledgers or cloud agents, this can provide new sources of failure or cause dependency issues.

6.5 Future Work

In future, we intend to work on the following:

1. Creating a safe backup and synchronization method for the user wallet in order to ensure a consistent user experience across many devices.

2. Scalability considerations are taken into account as the system grows. The burden on the blockchain and other components may expand dramatically as more students and educational institutions utilize the platform. In the future, we can emphasize on improving the system's performance so that it can manage a bigger number of users and transactions more effectively.

3. For mobile interactions, the research project specifies the use of the Aries Mobile Agent React Native (Bifold) and the Indicio Mediator Agent. Future enhancements might look towards increasing connectivity with other mobile wallets and mediator agents to provide users more options.

4. Implementing VC revocation will allow an issuer to withdraw a credential that was previously issued to a holder, ensuring more control over data and reducing fraudulent acts.

Chapter 7

Conclusion

Verification and authorization are done mostly using centralised and federated system. While federated identification systems can allow businesses to exploit people's personal information to keep and follow their internet activities without their permission, centralised identity systems frequently leave organisations open to large-scale hacks and data branches. Centralised identity management systems have led to frequent data breaches, identity theft, loss of personal data control, and dissemination of private information.

Credential fraud and a lack of product traceability are major issues in many industries, especially academia, because ID and credential verification procedures are antiquated, costly, sluggish, and ineffective. Our proposed system has mitigated all challenges that the traditional approach faces. If this proposed system can be integrated on a large scale then, the issue of sensitive information falling in the risk of fraudulent actions will be almost minimized.

As we are moving towards a digital world, we have to make sure there is a way to validate and authenticate everything we interact with, especially the artifacts that play a very important role such as certificates and credentials. On the internet anything can be forged, SSI along with the blockchain technology adds a trust component to the internet, by which exchange of information and an Identity on the internet can be trusted. Our system contributes to the vision of a trusted digital world by providing security to the hard earned certificates and making the verification process easier and cost effective for the verifiers.

Bibliography

[1] A. A. Schreier, K. Wilson, and D. Resnik, *Academic research record-keeping: Best practices for individuals, group leaders, and institutions*, Jan. 2006. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3943904/.

[2] R. Raman, D. Pramod, and V. Bharathi, *Information and communication technology based model for dematerialization of academic certificates for indian educational system*, Dec. 2013. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2363246.

[3] Raman, "Information and communication technology based model for dematerialization of academic certificates for indian educational system," *American Journal of Applied Sciences*, vol. 11, no. 1, pp. 85–88, 2014. DOI: 10.3844/ajassp.2014.85.88.

[4] N. Dlamini, S. Mthethwa, and G. Barbour, "Mitigating the challenge of hard-copy document forgery," in *2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, IEEE, 2018, pp. 1–6.

[5] M. Han, Z. Li, J. (He, D. Wu, Y. Xie, and A. Baba, "A novel blockchain-based education records verification solution," *Proceedings of the 19th Annual SIG Conference on Information Technology Education*, 2018. DOI: 10.1145/3241815.3241870.

[6] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinel, "A survey on essential components of a self-sovereign identity," *Computer Science Review*, vol. 30, pp. 80–86, 2018. DOI: 10.1016/j.cosrev.2018.10.002.

[7] Universa, *Blockchain in education*, May 2018. [Online]. Available: https://medium.com/universablockchain/blockchain-in-education-49ad413b9e12.

[8] *What is self-sovereign identity?* Dec. 2018. [Online]. Available: https://sovrin.org/faq/what-is-self-sovereign-identity/.

[9] A. Ezell, "Diploma mills and counterfeit operations," *College and University*, vol. 94, no. 3, pp. 39–46, 2019.

[10] M. S. Ferdous, F. Chowdhury, and M. O. Alassafi, "In search of self-sovereign identity leveraging blockchain technology," *IEEE Access*, vol. 7, pp. 103 059–103 079, 2019. DOI: 10.1109/ACCESS.2019.2931173.

[11] N. Kulabukhova, A. Ivashchenko, I. Tipikin, and I. Minin, "Self-sovereign identity for iot devices," in *International Conference on Computational Science and Its Applications*, Springer, 2019, pp. 472–484.

[12] M. Bahrami, A. Movahedian, and A. Deldari, "A comprehensive blockchain-based solution for academic certificates management using smart contracts," in *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*, IEEE, 2020, pp. 573–578.

[13] S. Hong and H. Kim, "Vaultpoint: A blockchain-based ssi model that complies with oauth 2.0," *Electronics*, vol. 9, no. 8, p. 1231, 2020.

[14] S. Huang, "Academic records verification platform based on blockchain technology," in *2020 International Conference on Computer Science and Management Technology (ICCSMT)*, IEEE, 2020, pp. 203–206.

[15] A. PREUKSCHAT, *Self-sovereign identity decentralized digital identity and verifiable credentials*. OREILLY MEDIA, 2021.

[16] A. Preukschat, D. Reed, and D. Searls, *Self-sovereign identity: Decentralized Digital Identity and verifiable credentials*. Manning, 2021.

[17] M. Shuaib, S. Alam, M. S. Alam, and M. S. Nasir, "Self-sovereign identity for healthcare using blockchain," *Materials Today: Proceedings*, 2021.

[18] H. Yildiz, C. Ritter, L. T. Nguyen, B. Frech, M. M. Martinez, and A. Küpper, "Connecting self-sovereign identity with federated and user-centric identities via saml integration," in *2021 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, 2021, pp. 1–7.

[19] Jul. 2022.

[20] A. De Salve, A. Lisi, P. Mori, and L. Ricci, "Selective disclosure in self-sovereign identity based on hashed values," in *2022 IEEE Symposium on Computers and Communications (ISCC)*, 2022, pp. 1–8. DOI: 10.1109/ISCC55528.2022.9913052.

[21] A. Fraser and S. Schneider, "On the role of blockchain for self-sovereign identity," *Competitive Advantage in the Digital Economy (CADE 2022)*, 2022. DOI: 10.1049/icp.2022.2032.

[22] M. Litoussi, M. Fartitchou, K. E. Makkaoui, A. Ezzati, and Z. E. Allali, "Digital certifications in moroccan universities: Concepts, challenges, and solutions," *Procedia Computer Science*, vol. 201, pp. 95–100, 2022. DOI: 10.1016/j.procs.2022.03.015.

[23] A. Mohammad and S. Vargas, *Challenges of using blockchain in the education sector: A literature review*, Jun. 2022. DOI: 10.3390/app12136380. [Online]. Available: https://doi.org/10.3390/app12136380.

[24] A. Satybaldy, A. Subedi, and M. Nowostawski, *A framework for online document verification using self-sovereign identity technology*, Nov. 2022. [Online]. Available: https://doi.org/10.3390/s22218408.

[25] M. S. Ferdous, A. Ionita, and W. Prinz, "Ssi4web: A self-sovereign identity (ssi) framework fornbsp;thenbsp;web," *Lecture Notes in Networks and Systems*, pp. 366–379, 2023. DOI: 10.1007/978-3-031-21229-1_34.

[26] [Online]. Available: https://docs.opencerts.io/v1/.

[27] [Online]. Available: https://github.com/WebOfTrustInfo/self-sovereign-identity/blob/master/self-sovereign-identity-principles.md.

[28] [Online]. Available: https://www.w3.org/TR/did-core.

[29] [Online]. Available: https://www.w3.org/TR/vc-data-model/.

[30] [Online]. Available: https://github.com/hyperledger/aries-rfcs/blob/main/concepts/0004-agents/README.md.

[31] [Online]. Available: https://github.com/hyperledger/aries-rfcs/blob/main/features/0037-present-proof/README.md.

[32] *A blockchain platform for the enterprise*. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/.

[33] *AnonCreds Specification — hyperledger.github.io*, https://hyperledger.github.io/anoncreds-spec/, [Accessed 23-09-2023].

[34] *Aries-rfcs/features/0036-issue-credential/*, https://github.com/hyperledger/aries-rfcs/blob/main/features/0036-issue-credential/README.md, [Accessed 17-09-2023].

[35] Blockcerts. [Online]. Available: https://www.blockcerts.org/guide/.

[36] *DIDComm Messaging Specification v2.0 — identity.foundation*, https://identity.foundation/didcomm-messaging/spec/v2.0/, [Accessed 17-09-2023].

[37] ENISA, *DIGITAL IDENTITY Leveraging the Self-Sovereign Identity (SSI) Conceptnbsp; to Build Trust*,

[38] *Exploring the use of self-sovereign identity for event ticketing systems - Electronic Markets — doi.org*, https://doi.org/10.1007/s12525-022-00573-9, [Accessed 17-09-2023].

[39] O. Ghazali and O. S. Saleh, "A graduation certificate verification model via utilization of the blockchain technology," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10,

[40] Hyperledger, *Hyperledger/aries-mobile-agent-react-native: Aries mobile agent react native - part of the aries bifold effort to provide ssi capabilities in a production ready app*. [Online]. Available: https://github.com/hyperledger/aries-mobile-agent-react-native.

[41] *Hyperledger/aries-cloudagent-python: Hyperledger aries -cloud agent python (aca-py)*. [Online]. Available: https://github.com/hyperledger/aries-cloudagent-python.

[42] *Indicio Public Mediator — indicio-tech.github.io*, https://indicio-tech.github.io/mediator/, [Accessed 17-09-2023].

[43] *Introduction to Hyperledger Self-Sovereign Identity Blockchain Solutions — edx.org*, https://learning.edx.org/course/course-v1:LinuxFoundationX+LFS172x+1T2023/home, [Accessed 17-09-2023].

[44] *Leveraging Self-Sovereign Identity, Blockchain, and Zero-Knowledge Proof to Build a Privacy-Preserving Vaccination Pass — doi.org*, https://doi.org/10.2139/ssrn.4036226, [Accessed 17-09-2023].

[45] *Verifiable Credentials Data Model v1.1 — w3.org*, https://www.w3.org/TR/vc-data-model/, [Accessed 17-09-2023].

[46] *Verifiable Credentials Data Model v2.0 — w3.org*, https://www.w3.org/TR/vc-data-model-2.0/, [Accessed 17-09-2023].