TU WIEN Informatics

Privacy-Preserving eHealth: A Self-Sovereign Identity Based Infrastructure for Medical Records

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering und Internet Computing

eingereicht von

Sven Bombera, BSc.
Matrikelnummer 11808593

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Dipl.-Ing. Dr.techn. Bakk.techn. René Baranyi
Mitwirkung: Univ.Prof. Dipl.-Ing. Dr.techn. Thomas Grechenig

Wien, 5. April 2024

Sven Bombera	René Baranyi

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

Privacy-Preserving eHealth: A Self-Sovereign Identity Based Infrastructure for Medical Records

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering und Internet Computing

by

Sven Bombera, BSc.
Registration Number 11808593

to the Faculty of Informatics

at the TU Wien

Advisor: Dipl.-Ing. Dr.techn. Bakk.techn. René Baranyi
Assistance: Univ.Prof. Dipl.-Ing. Dr.techn. Thomas Grechenig

Vienna, 5th April, 2024

Sven Bombera	René Baranyi

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Sven Bombera, BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 5. April 2024

Sven Bombera

Danksagung

Einen besonderen Dank möchte ich den Wissenschaftlichen Mitarbeitern René Baranyi und Thomas Grechenig für die Betreuung und Unterstützung während dieser Diplomarbeit aussprechen. Mit ihrer Erfahrung konnte ich ein passendes Thema wählen und die Diplomarbeit erfolgreich abschließen. Zudem bedanke ich mich für ihre Einsatzbereitschaft mir einige passende Interviewpartner zu organisieren.

Ebenfalls möchte ich mich bei den Interviewpartnern bedanken, die mir sowohl bei der Erstellung der Anforderung als auch bei der Evaluierung der vorgestellten Architektur wertvolle Inhalte geliefert haben.

Des Weiteren bedanke ich mich bei meinen Kommilitonen, mit denen ich gemeinsam die Hindernisse des Studiums überwunden habe. Abschließend möchte ich mich noch meiner Freundin bedanken, die mir eine wichtige mentale Stütze war, und bei meinen Eltern, die mir dieses Studium erst ermöglicht haben.

Acknowledgements

I would like to express my special thanks to the research assistants René Baranyi and Thomas Grechenig for their support and supervision during this thesis. With their experience I was able to choose a suitable topic and successfully complete the thesis. I would also like to thank them for their willingness to organise suitable interview partners for me.

I would also like to thank the interviewees who provided me with valuable information, both during the creation of the requirements and during the evaluation of the architecture presented.

I would also like to thank my fellow students, with whom I overcame the obstacles of my studies. Finally, I would like to thank my girlfriend, who has been an important mental support for me, and my parents, who have made my studies possible in the first place.

Kurzfassung

In der heutigen Zeit sind digitale Daten eine wertvolle Ressource und sind meist im Besitz von Service Providern. Speziell im Gesundheitsbereich sind Daten sensibel und können für Patienten bei nicht gerechter Handhabung einen Schaden verursachen. In aktuellen Systemen müssen die Patienten den Betreibern der Gesundheitssysteme vertrauen, da diese auf zentralisierten oder föderierten Identity und Access Mechanismen basieren, und haben nicht die volle Kontrolle über ihre eigene Daten.

Ein neuwertiges Konzept im Bereich von Identity und Access Management, das den Benutzer in das Zentrum der Applikation stellt, hat die Bezeichnung self-sovereign identity (SSI). Das Ziel dieser Diplomarbeit ist es eine eHealth Architektur für medizinische Daten mit dem Ansatz von SSI zu entwerfen. Dabei hat der Nutzer die volle Kontrolle über sein Daten und muss nicht auf andere Entitäten vertrauen. Die Thesis zeigt den ganzen Prozess der Erstellung der Architektur, von der Erhebung der Anforderung bis zur Evaluierung der fertigen Architektur. Des Weiteren erklären sie welche Möglichkeiten es gibt SSI in Gesundheitsarchitekturen zu verwenden und spezifizieren die einzelnen Komponenten, deren Interaktionen, und die Abläufe innerhalb des Systems. Außerdem berücksichtigen diese Thesis die Datenschutzgrundverordnung und analysieren welche Komponenten kompatibel sind und wo es zu Konflikten kommt.

Der finale Prototyp basiert auf Anforderungen, die aus Interviews mit Experten erhoben wurden und basiert neben SSI auf dem Konzept von Decentralized Web Nodes. Diese einzigartige Kombination von Decentralized Web Nodes und SSI ist in der wissenschaftlichen Literatur, insbesondere im Gesundheitswesen, bisher nicht erforscht worden. Die eHealth-Architektur wird anhand eines Proof-of-Concept, eines Threat Models und einer zweiten Runde von Experteninterviews evaluiert. Die Ergebnisse zeigen einen neuwertigen Ansatz in dem Benutzer ihre medizinische Daten sicher und datenschutzfreundliche Weise verwalten können.

Abstract

In today's world, digital data is a valuable resource and is usually owned by service providers. Especially in the healthcare sector, data is sensitive and can cause harm to patients if not handled fairly. Current healthcare systems rely on centralized or federated identity and access mechanisms, which require patients to trust the operators. However, patients do not have full control over their own data.

An emerging concept in the field of identity and access management that places the user at the center of the application is called self-sovereign identity (SSI). The aim of this thesis is to create an eHealth architecture for medical data using the SSI approach. The architecture allows the user to have full control over their data without relying on other entities. The thesis contains the process of creating the architecture, from identifying the requirements to evaluating the final architecture. It explains the potential use of SSI in healthcare architectures, detailing the individual components, their interactions, and the system's processes. Additionally, this thesis analyze the compatibility of the components with the General Data Protection Regulation and identify any conflicts.

The final prototype is based on requirements gathered from expert interviews and utilizes the concept of Decentralized Web Nodes in addition to SSI. This unique combination of Decentralized Web Nodes and SSI has not been previously explored in scientific literature, particularly in the healthcare sector. The proposed eHealth architecture is evaluated through a proof-of-concept, a threat model, and a second round of expert interviews. The results demonstrate a novel approach in which users can securely and privately manage their medical data.

Contents

CHAPTER 1

Introduction

Digital data, especially personal data, is one of the most valuable resources and will be even more valuable in the future. Indirectly, this also highlights the relevance of privacy and security in systems that work with personal data. Identity management systems (IMS) administer the collection, authentication, or use of identity and information linked to identity [51]. Pöhn and Hommel [85] proposed that identity management systems could be categorized into three categories, which are (i) centralized/network-centric, (ii) federated/application-centric, and (iii) decentralized/user-centric. In centralized identity management systems, the users have little direct control over their identity, data, and privacy protection. This may result in users not knowing what is happening with their data, and not being able to manage their identities and data across multiple different service providers. Personal data in healthcare must be handled with care, because unauthorized access to health-related data could lead to personal disadvantages. For example, an insurance company could deny coverage to a particular patient based on leaked medical history [103].

Decentralized identity management is often based on blockchain technology. Using blockchains to store personalized data conflicts with privacy policies, which is why they are often used for other purposes in decentralized approaches. Partly due to lack of knowledge and experience in decentralized identity management, federated identity management systems are seen more and more frequently. One example for federated identity management is the Telematikinfrastruktur, which is the platform for networking the eHealth applications in Germany. It uses a federal identity management system to make the system more flexible and allow the users secure use of the infrastructure [44]. However, this approach could lead to difficulties when it comes to sharing the data with other applications and controlling the data. Using a decentralized approach for identity and data management could increase privacy and interconnection between healthcare systems and other applications, benefiting all participants in the ecosystem.

1.1 Aim of the Work

The overall aim of this thesis is to design a prototype for an eHealth infrastructure using decentralized identity management systems. This infrastructure should interconnect hospitals, doctors' practices, laboratories, other facilities in the field of diagnostics such as radiological institutions, and health insurance companies in a user centric and privacy preserving way. During medical examinations and treatments healthcare providers generated personal data, among others in the field of documentation and diagnostics. This architecture will allow the users to own their personal health records (PHR) and electronic health records (EHR) issued by these healthcare providers. Handling and processing continuously created data in the context of PHRs such as heart rate, step count, or glucose levels will not be part of this theses. To conform to European standards, this architecture will comply with the European legal standards of data protection.

One novel approach to decentralized identity management is self-sovereign identity (SSI). This allows users to fully own and manage their digital identity in a decentralized manner and without depending on third-party providers [77]. The focus is on designing an eHealth infrastructure using the approach of self-sovereign identity. The system will enable secure access and storage of user data while users have full control over their data. Furthermore, the design will ensure data authenticity and reduce the risk of privacy leaks. An efficient and SSI-based approach to managing various healthcare data in the healthcare sector has not yet been researched and will therefore be investigated in this thesis. We will identify the most important use cases for such an architecture, for example a healthcare provider issues a medical record to a patient and granting and revoking access to healthcare providers when sharing user data. Some parts of the architecture will be implemented to create functional prototypes. This leads to the following research questions:

RQ1 How might self-sovereign identity be used in the context of healthcare?

RQ2 What laws and regulations need to be considered when storing and providing eHealth data related to self-sovereign identity in order to create an eHealth infrastructure that complies to the European laws of data protection?

RQ3 How to construct a privacy preserving eHealth architecture for interconnecting healthcare facilities based on self-sovereign identity, considering theoretical concepts and implementations?

1.2 Methodology

For achieving the goal of this thesis and answer the research questions, we employ the following methodologies. To research the state of the art and accumulate background knowledge for all three research questions, we utilized, among other things, the systematic mapping process, explained by Petersen, et al. [84]. We conduct a search based on key

words. The used data sources are the search engine Google Scholar and digital libraries for journals, standards, and conferences, for example IEEE Xplore and ACM Digital Library. For saving the literature, we pre-select the papers based on the title and filter and reduce the number of papers based on the abstract. Afterwards, we use the snowballing technique to identify similar literature to our research topic.

In addition, we use legal texts and literature in the field of eHealth and data handling to cover the legal situation regarding data protection law and answer **RQ2**. We focus on the corresponding legal guidelines and laws of Europe. One of the most important sources is the General Data Protection Regulation (GDPR [38]). This part of the thesis clarifies how to handle medical data in the context of self-sovereign identity.

For answering **RQ3** we identify the most important use cases for such an architecture and create functional and technical requirements. We define these requirements by interviewing specialists in the individual fields of research and using relevant literature in the field of security and regulations. To identify suitable experts for the interview, we searched the university environment and our professional environment for individuals with relevant experience and knowledge. The content of the requirements refers to properties that are essential for such an eHealth architecture and the use cases, for example, how performant, secure, or practical the system is.

The next step is to define a concept for the actual eHealth architecture. This includes the questions of how decentralization technologies and cryptographic concepts are used for the eHealth system. Additionally, we define possibilities to securely share the data between a user and other entities with user consent. This part is used to answer **RQ1**. To make the processes clear and concise we define the procedures and the actual workflow by using UML diagrams.

Another method for answering **RQ3** is proof of concept. During the design process, we implement prototypes for parts of the architectures. The implementations are used to measure and evaluate the defined requirements. To properly implement parts of the architecture, we follow the standards of software-engineering. One of the last parts of the thesis is an evaluation of the results. We check if the defined architecture and the implemented prototypes match the functional and non-functional requirements and evaluate the security using a threat model. Additionally, a second round of interviews with specialists in the concerned fields of research is conducted, to incorporate the expertise of external parties in the evaluation. This clarifies and argues why the architecture is secure and functional, and in which areas are better than the current standards. In Figure 1.1 we visualized the methodology process.

1.3 Overview

Chapter 2 explains the concepts and provides background information necessary for a better understanding of the thesis. Section 2.1 includes the three types of identity management systems, providing the reader with a better understanding of the motivation

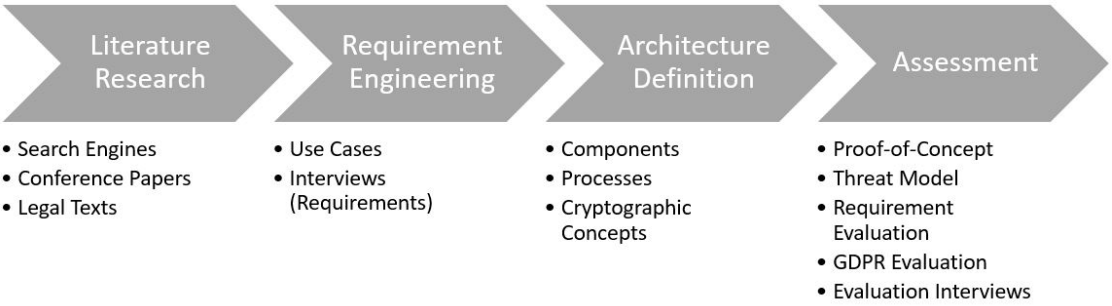

Figure 1.1: Overview of the methodology

behind this work. Additionally, this section presents relevant information about eHealth systems. Section 2.2 focuses on SSI as the underlying technology and explains its functionality and components. Additionally, we propose three methods for using self-sovereign identity in digital health systems, outlining their advantages and disadvantages. This serves to answer **RQ1**. Furthermore, this section discusses standards and vulnerabilities related to components of SSI, as found in the literature. Section 2.3 includes the legal aspects of this thesis and clarifies the regulations based on European data protection laws regarding SSI and health data. The GDPR is analyzed in the context of SSI and health data, contributing to answering **RQ2**. This section serves as the basis for evaluating the compatibility of our architecture with the GDPR.

Chapter 3 discusses the relevant literature on decentralized eHealth systems, including research on blockchain-based solutions, user-centric approaches, concepts using advanced cryptography and decentralized storage, and self-sovereign identity approaches. We make a comparison between the state-of-the-art literature and our proposed architecture, demonstrating its superiority to current standards.

Chapter 4 contains the evaluations and results of this thesis. The first Section (4.1) defines the guidelines for the architecture, including use cases and requirements, that will be considered during the creation process. To create a specific eHealth architecture, we have identified the most important use cases that will be considered during the creation process. Additionally, we have established objective requirements as the basis for the architecture. These requirements were developed by interviewing three experts using a pre-established guideline. Section 4.2 defines the proposed architecture. In Subsection 4.2.1, we explain all necessary components and their functionality, as well as argue how and why they work. Subsection 4.2.2 explains the cryptographic concepts used in the architecture, as security and privacy rely on cryptography. The final subsection of this section, Subsection 4.2.3, illustrates the interactions and processes among the entities in the proposed architecture. These three sections aim to address research questions **RQ1** and **RQ3**. Section 4.3 evaluates the defined architecture and contributes to answering all three research questions. First, we describe the proof-of-concept, including the implementation process of the proposed architecture. We implemented the basic functionalities of the architecture and measured the performance criteria. The following

subsection describes the threat model we have created. We argue the security of the presented architecture and identify potential areas for improvement. The following subsections evaluate the requirements and GDPR compatibility of this architecture based on defined concepts, literature, legal texts, and measurements. The evaluation concludes with an assessment from a group of experts. We conducted a second round of interviews where they evaluated our architecture and compared it to current standards.

The final chapter provides an overview of this thesis and summarizes its key points.

CHAPTER 2

Background

This chapter serves as the foundation for a better understanding of the thesis. It contains relevant concepts and background information. The first section provides theoretical background information to help the reader understand the motivation behind this work. The following section explains the concept of SSI and its relevant components. Additionally, we propose three approaches for using SSI for medical records in healthcare, which address **RQ1** and form the basis for the architecture. The final section presents an overview of the General Data Protection Regulation (GDPR) and its compatibility with SSI, forming the basis for addressing **RQ2**.

2.1 Theoretical Background

This section includes the topics of identity management systems and eHealth. The first part describes the three different kinds of identity management and highlights the drawbacks of the centralized and federated approach. The second part provides information about eHealth, the different types of health records and health data itself.

2.1.1 Identity Management Models

Having a digital identity is essential for using applications, services, etc. on the internet. Such a digital identity is the digital representation of credentials and identifier of entities in order to access applications and services [16]. To administer the collection, authentication, or use of identity and information linked to identity, there exists different management systems [51]. Pöhn and Hommel [85] proposed that identity management systems could be categorized into three categories, which are (i) centralized/network-centric, (ii) federated/application-centric, and (iii) decentralized/user-centric. All three identity models differ in their characteristics and use cases. We illustrated all identity models in the corresponding section. The illustrations in Glauser's [47] masterthesis form the basis for these images. Figure 2.1 shows the legend for the respective illustrations.

Figure 2.1: Legend for the identity model illustrations

2.1.1.1 Centralized Identity Model

According to Pöhn and Hommel [85], and Preukschat and Reed [86] in the centralized identity model, the identity of a user is bound to an account, which allows to use just a specific website, service, or application. This tight binding between an identity and the application causes, for example, a deletion of all identities when the application is shut down. The owner of the application can access, edit, and delete data from the identities. Therefore, users have little control over their identity, their data, and the protection of their privacy. Moreover, the users are not able to manage their identities and data across multiple different service providers, as shown in Figure 2.2. Some disadvantages of this identity model are the following [86]:

- For each application or service is a new account/identity necessary. The identity data is not portable or reusable.
- The users do not have full control over the data. The actual owner is the application/service.
- Centralized databases of personal data are giant honeypots, which could lead to financial damage in case of a security issue.

2.1.1.2 Federated Identity Model

Pöhn and Hommel [85] mention that in the federated identity model, an account is not bound to a single website or application. Instead, the identity is available to multiple entities within certain trust boundaries. This is accomplished by using, for example, an identity provider (IDP), which is used to log in a user and share the data associated with the identity, with compatible applications or services, accroding to Preukschat and Reed [86]. In this model, as illustrated in Figure 2.3, an application is not able to access, edit or delete identity data without approval of the user. The IDP stores all the identity data and therefore is the actual owner. There are several federated identity protocols,

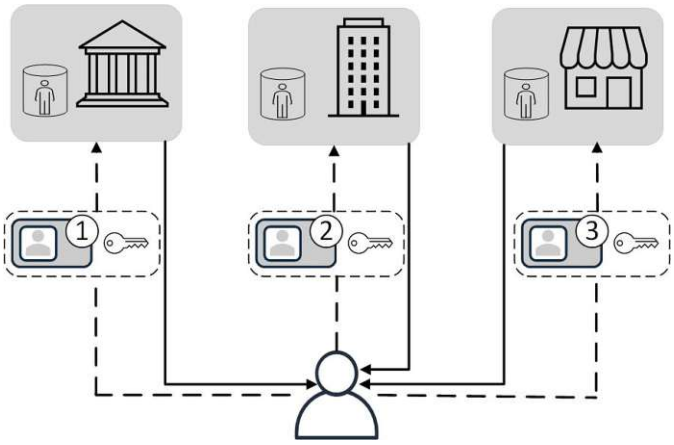

Figure 2.2: Illustration of the centralized identity model

for example Security Assertion Markup Language (SAML), OAuth, and OpenID. The disadvantages of such a model are [86]:

- IDPs are single point of failures and some of the biggest honeypots for cybercrime.
- Not every applications and services support every IDP.
- An IDP acts as a "man in the middle", which allows the IDP to log user activity made with the linked identity.
- As in centralized identity models, the data is not portable, if the account gets deleted at the IDP.

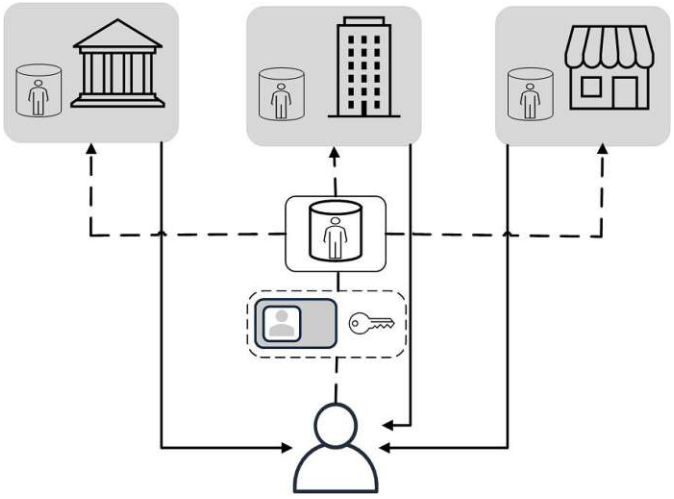

Figure 2.3: Illustration of the federated identity model

2.1.1.3 Decentralized Identity Model

The decentralized identity model is not dependent on centralized or federated identity provider but works in a decentralized manner. Compared to the other two approaches, the entity itself is the center of the model, as illustrated in Figure 2.4. In this case, an entity can refer to either a user or a company. This is ensured by keeping as much information as possible within the entity itself [27]. In this identity model exist direct relationships between two entities, e.g., between users and applications. Such a relationship is a shared connection between the two parties that is not owned or controlled by either party. Both sides can delete the connection at any time. The base for decentralized identity models is public key cryptography. Self-Sovereign identity is a decentralized identity approach and will be described in section 2.2. [86]

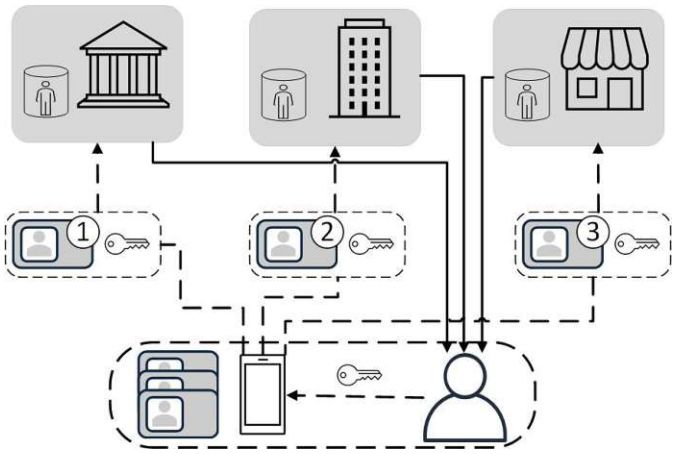

Figure 2.4: Illustration of the decentralized identity model

2.1.2 eHealth

The European Commission defines eHealth as tools and services that use information and communication technologies (ICTs) to improve prevention, diagnosis, and processes in the health sector [23]. This digitization helps to improve various areas in the healthcare sector, among others the interconnection in and between healthcare facilities. Concrete applications are the electronic health insurance card, health records, medication overviews, and telemedicine [17].

2.1.2.1 Health Data

The diverse areas of healthcare require different medical data. As a result, data take on different formats, files, sizes, etc. Houtan, et al. [59] classified data into two categories based on their research, which are (i) self-reported data and (ii) health and medical data placed in healthcare information systems (HIS). A healthcare information system is used for the collection, storage, management, and exchange of healthcare data. Self-reported data describes data which is generated by third party lifestyle applications and IoT

devices, such as mobile pedometers or wearables. The other category refers to data, which is more essential to the healthcare providers such as medical records, lab results and medical history. This includes also static data, for example a name or the birth date, and data created by the user such as questionnaires.

A healthcare information system can be classified into three types: electronic medical records (EMR), electronic health records (EHR) and personal health records (PHR)[37]. An EMR is an electronical record in healthcare containing information from a patient. It is created, gathered, managed, and consulted within one specific healthcare provider and cannot be linked to other sources of data from other facilities [63].

Electronic Health Records (EHR)
According to Jabob [63] EHRs are digital medical records, which support the interoperability between healthcare providers. They are created, managed, and consulted by authorized staff in different healthcare facilities. An EHR interconnects EMRs across multiple healthcare facilities into one comprehensive longitudinal health record for a specific patient. Houtan et al. [59] mentions the existences of several organizations which define standards for the eHealth sector to ensure interoperability, security, and reliability. In the field of interoperability, the available standards are for example OpenEHR([83]), ISO EN13606([62]), DICOM([31]) and Health Level 7([53]).

Jacob [63] points out the variety of data an EHR manages such as texts, images, and sounds, that are needed in different areas of healthcare. This data is used in the following use cases, among others [63]:

- **Documentations of clinical notes**, including history and exam done by a doctor or healthcare provider, progress notes and nursing notes (pulse rate, temperature)
- **Chart review and results review**, allowing a healthcare facility to access results of lab tests, medications of a patient and medical images (e.g., MRI, CT, x-ray)
- **Orders for laboratory, medications, radiology, procedures**, needed for areas where billing and the services are carried out
- **Care plan** for planning patient care and document changes in the patient's condition

Electronic Patient Records (PHR)
The concept of PHRs [55] is similar to EHRs, except they are maintained by the patients themselves. PHRs are defined as electronic records containing medical data and information about a patient that are fully owned by the patient. This allows the patients to access their data, such as test results and prescriptions on-demand. Additionally, the patients can manage their medical records and add information to them. Data exchange between the healthcare facilities and the patients PHR require consent from the patients, which make this approach more privacy-preserving [59]. Typical examples of PHRs

are immunization status, emergency contact, insurance information, allergies and other relevant medical conditions that can impact the delivery of emergency care, patient's medical history, and past medical and surgical interventions [82].

Data Types

The actual data in EHRs and PHRs is categorized by Raghupathi and Raghupathi [95] into structured, semi-structured, and unstructured data. Structured data describes data, which can be stored, queried, analyzed, and manipulated by a machine. This includes numeric and alphanumeric values such as patient name, height, weight, data of birth, treatment reimbursement codes, hospital name and other information, which could be stored in databases in standardized formats.

In contrast, Raghupathi and Raghupathi [95] specified unstructured data as not automatically processable. Examples are medical images (e.g. MRI, CT, x-ray), office medical records and evaluations, handwritten nurse and doctor notes, hospital admission and discharge records, paper prescriptions, audio recordings (e.g. dictation) and radiograph films.

EHRs and PHRs contain different file types and for this reason also the sizes of these files and information differ. The information can be represented, starting with small text files up to large images such as CT and video images of operations. A file with text based information starts with a size of a few kilo byte, whereas an average cardiac MRI exam had a size of 200 MB and a Viosworks exam, which allows to acquire a whole heart functional exam in a non-gated, free breathing acquisition, had 20 GB in the year 2016 [26, 54].

2.1.3 Terminology

Regarding the topics of security and privacy, a number of considerations flow into the development of the architecture. In the following, we explain some terms that are related to the topic and are essential to understand.

2.1.3.1 Trust

The National Institute of Technology (NIST) [108] defines trust in one of their publications as "A belief that an entity meets certain expectations and therefore, can be relied upon".

2.1.3.2 Privacy

In the NISTIR 4734 publication of the NIST [109], they defined privacy as "The right of a party to maintain control over and confidentiality of information about itself". The main goal of the architecture is to create a privacy-preserving solution, which may not be entirely trustless. The architecture, along with its protocols, processes, and assumptions, guarantees user privacy, but some level of trust is required, such as in the issuer. The more trust in another entity is necessary, the greater the threat to privacy since the confidentiality of the information depends on another entity.

2.2 Self-Sovereign Identity

This section aims to explain the concept and functionality of self-sovereign identity. This includes the individual components of an SSI architecture. The term SSI does not have one explicit definition, it is a concept using different principles and technologies. We will outline what these principles are and when a decentralized identity management system could be considered as self-sovereign. Furthermore, we propose three approaches of how SSI could be used in the context of healthcare and discuss existing criticism on the concept of SSI.

2.2.1 Overview

Self-sovereign identity is a user-centric identity management system, which allows individuals to fully own and manage their digital data. As a consequence, this allows a user and its data to exist independently from applications and services [77].

With the emergence of bitcoin [80], the topic of decentralization became popular and decentralized ledger technology opened new opportunities. Avellandeda, et al. [4] mention that several sessions on "blockchain identity" were held at the 23rd Internet Identity Workshop in Mountain View. This was one of the first events that addressed the topic of blockchain identity. The foundational idea and term "self-sovereign identity" was first mentioned by developer Devon Loffreto on his own blog ([100]), as described by Christopher Allen [2]. The actual concept of "self-sovereign identity" was first described in detail by Christopher Allen [2], who is considered the inventor of SSI. He published a blog entry in 2016, called "The Path to Self-Sovereign Identity". There he explained his idea and vision of a self-sovereign identity, that returns power over information to the user in a decentralized way.
When the article was published, the digital world was becoming more and more important and offered the possibility of redefining modern concepts of identity. This is where Christopher Allen's SSI approach should help.

In 2015, Zyskind, et al. [126] proposed a decentralized identity management system that uses blockchain technology to ensure users own and control their data. The authors defined three privacy issues, where their solution should protect the users. These privacy issues overlap with the description of "self-sovereign identity" given by Christopher Allen [2]. Christopher Allen mentioned that there is no explicit definition for self-sovereign identity. Instead, he defined ten principles, which attempt to ensure user control, as the foundation of SSI. These principles are described in the following:

1. **Existence**: Users must have an independent existence. A self-sovereign identity represents an identity and cannot exist completely in digital form.

2. **Control**: Users have full control over their digital identity. They should always be able to manage and use them.

3. **Access**: Users always have access to their digital identity data without any restrictions.

4. **Transparency**: The systems and algorithms for the identities have to be transparent on how they work and how they are managed.

5. **Persistence**: Identities must be long-lived. The user should decide over the live span of the identity, including its deletion.

6. **Portability**: The information and services about identity must be transportable. The identity must not be held by a single third-party entity, as this will result in the identities disappearing as well if the service is deleted.

7. **Interoperability**: Identities should be as widely usable as possible, without losing user control.

8. **Consent**: Users must agree to the use of their identity. Sharing and providing the identity to another entity is any possible with the consent of the user.

9. **Minimalization**: Disclosure of the identity data must be minimized to enhance privacy. It should just involve the minimum amount of data necessary to accomplish the task.

10. **Protection**: The rights of users must be protected. In a conflict between the needs of the identity network and the rights of users, the users are always in favor.

These principles were the foundation for the emergence of SSI. Mühle, et al. [77] and Preukschat and Reed [87] describe an architecture of SSI, which consists of four building blocks. These four building blocks, shown in Figure 2.5, are: (i) blockchains or other verifiable data registries, (ii) verifiable credentials (VCs), (iii) decentralized identifiers (DIDs) and (iv) a trust triangle consisting of the entities of issuer, holder, verifier. Preukschat and Reed [87] mention three additional building blocks, which are (v) credential repositories such as digital wallets, (vi) digital agents, and (vii) governance frameworks.

The base of the model is a blockchain or any other kind of verifiable registry. It is used to register decentralized identifier (DID), which are tied to a specific user. This maintains the pairing of identification and authentication. By using asymmetric cryptography and pairing the public key with the identifier on the blockchain, anyone can verify the identifier by reading the blockchain. The verifiable credentials contain a set of the actual identity claim about the subject of the credential. The trust triangle consists of the three roles issuer, holder, and verifier. Every verifiable credential needs an issuer to create it. Such issuers are mostly organizations, such as government agencies, financial institutions, or universities. The created VCs were then passed to holders, which hold them in a credential repository. They can present proofs of claims from one or more credentials to the verifiers. The verifiers want to check one or multiple claims about the subjects of credentials. They request a proof from a holder, who responds with a proof, which can

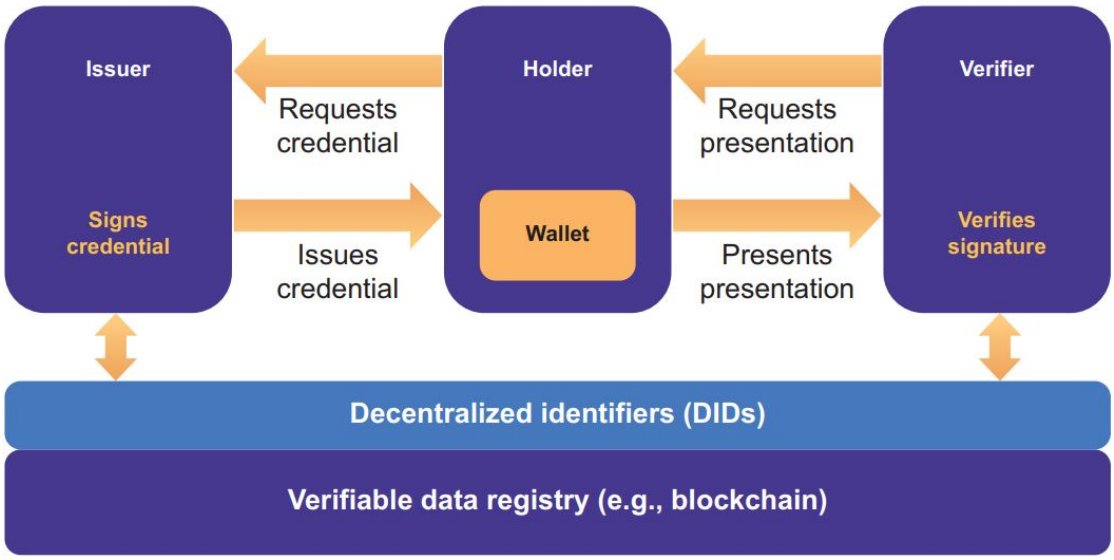

Figure 2.5: Self-sovereign identity model [87]

then be verified by the verifiers. Additionally, the verifier verifies the digital signature of the issuer by reading the DID from the blockchain. [77, 88]

2.2.1.1 Verifiable Data Registries (VDR)

In SSI verifiable data registries act as a decentralized public key infrastructure (DPKI), which allows to register decentralized identifiers. It replaces a central authority and maintains the pairing of identification and authentication. This allows an entity to check the validity of an identifier and the corresponding public key. The majority of VDRs is based on Distributed Ledger Technologies (DLT), in specific blockchains. [77, 89]

In general DLTs allow users who do not trust each other to interact without the need of a trusted third party in a decentralized way. DLTs are data structures, designed in a transparent, traceable, and secure way, to record transactions and functions. In general, DLTs are based on public key cryptography, distributed peer-to-peer networks, and consensus mechanisms. A blockchain is a specific type of DLT. It is an immutable ledger storing transaction histories. Each transaction is digitally signed using public key cryptography. Multiple transactions are comprised into blocks. These blocks are then connected to each other using hash codes (hash pointers), where each block references the previous block. The consensus mechanism decides which block is appended to the chain. All the transactions and the chain itself relies on public key cryptography. The structure and properties of a blockchain make it hard to manipulate the transactions. [36]

Another approach for VDRs is the Key Event Receipt Infrastructure (KERI [105, 106]). Unlike DLTs, where the root of trust relies on algorithms or transactions, KERI has a self-certifying root of trust, established solely on secure random number generation and

cryptography. Today this seems as the most self-sovereign possibility for decentralized key management and represents an alternative to DLTs as VDRs. The principle relies on the self-certifying identifier (SCID). A SCID is derived from a public/private key pair using one or more applications of cryptographic one-way functions. Similar to the concept of a public key-based identifier in Bitcoin. The cryptographic nature of SCIDs enables anyone with access to the SCID and the public key to verify the binding using cryptography alone, thus establishing its self-certified status. The purpose of KERI is to enable the entire DID method to be self-certifying, including the creation of SCID and all subsequent key rotations. The principle of KERI is similar to the original version of Pretty Good Privacy (PGP) with the difference that key rotation has not to be done manually. [90, 106]

2.2.1.2 Decentralized Identifiers

Decentralized identifiers (DIDs) were standardized by the World Wide Web Consortium (W3C) [114] to create a new and global type of identifier that enables verifiable and decentralized digital identity. In contrast to most unique identifiers issued by central authorities, DIDs allow individuals and organizations to generate their own identifiers using systems they trust. Additionally, this concept allows an entity to create and own as many DIDs as necessary. The ownership of identifiers could be proven by using cryptographical proofs such as digital signatures. An overview of the DID architecture with the basic components is shown in Figure 2.6. [114]

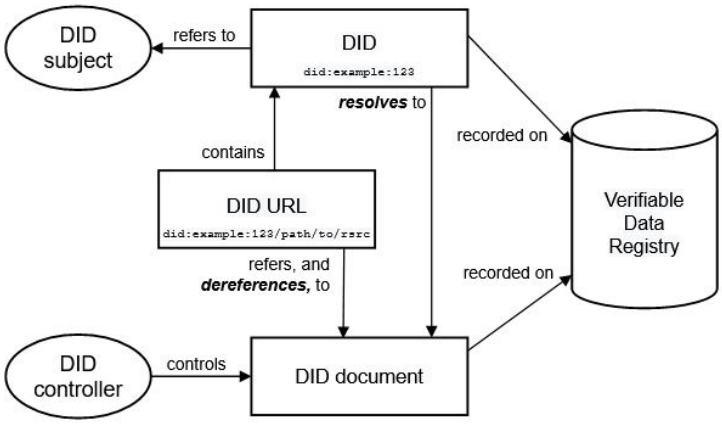

Figure 2.6: Overview of DID architecture [114]

A DID is represented as a text string consisting of three part, shown in Figure 2.7. The first part is the DID URI scheme identifier. The second part describes the DID method, which is a definition of the implementation of a DID method scheme. This includes how to create, update, read and deactivate a DID. The last part of the DID represents the specific identifier for a DID method. [114] A DID is a Uniform Resource Identifier (URI) used to resolve a DID document. This DID document contains information describing a

Figure 2.7: Example of a decentralized identifier [114]

DID subject. A DID subject is any entity in the real world, which could be identified by the DID. DID subjects are for example persons, organizations, physical things, or digital things. Typical data contained in a DID document are mechanism and cryptographic public keys for authentication to prove the association with the DID. A DID refers to exactly one DID document. In addition to DIDs, there exist DID URLs. They extend the basic DID and allow to incorporate other standard URI components, such as path, query, and fragment to locate a particular resource inside a DID document. [114]

2.2.1.3 Verifiable Credentials

As with DIDs, the World Wide Web Consortium (W3C) [119] created a standard for verifiable credentials (VC). They solve the problem of a missing possibility to express credentials on the web such as in the physical world. A credential in the physical world might contain information about identification of the subject of the credential, the issuing authority, the credential itself (e.g., type, expiration date), and specific attributes and properties, as shown in a simplified representation in Figure 2.8. A VC provides a standard way to digitally express all this information of credentials such that they are cryptographically secure, privacy respecting and machine-verifiable. Credentials consist of one or more claims made by the same entity. A claim itself is a statement about a subject and a subject could be any entity, similar to a "DID subject". Furthermore, subject-property-value relationships define claims. These individual claims can be merged to express a graph of information about a subject. A verifiable credential is defined as a set of tamper-evident claims and metadata that cryptographically prove who issued it. The metadata in a VC contains information, describing properties of the credential, such as the issuer, the expiry date and time, a representative image, a public key to use for verification purposes and the revocation mechanism. A verifiable presentation (VP), shown in a simplified version in Figure 2.9, is another component of the verifiable credential standard. It ensures privacy and allows the holder of a credential to reveal just a subset of the verifiable data, stored in one or multiple VCs. This property is called selective disclosure. [119]

The content of verifiable credentials and verifiable presentations could be represented by various data representation, such as XML or YAML. However, the most common way, which is also described in the documentation of the VCs, is using JSON-LD and plain JSON formats. [119]

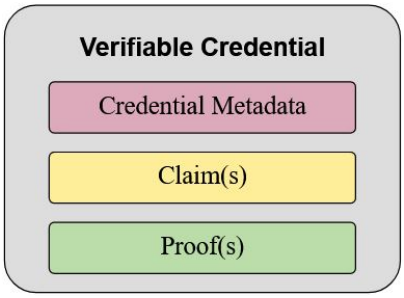

Figure 2.8: Overview of a verifiable credential [119]

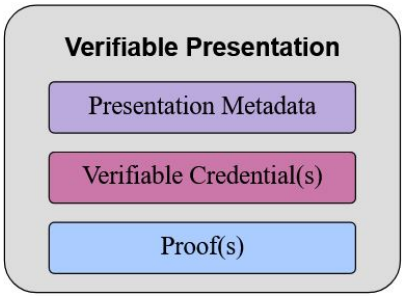

Figure 2.9: Overview of a verifiable presentation [119]

The use of verifiable credential is based on the three roles issuer, holder, and verifier. An issuer creates the VC and passes it to the holder. For validity and verifiability, the issuer includes its own identifier and the identifier of the holder. The identifiers of the VCs are not necessarily based on DIDs. However, DIDs are often used and are a practical solution. The holder receives the VC and is now able to provide verifiable presentations to a verifier. When the verifier receives a VP, it can verify the signatures and the validity of the credential/presentation. In case of using DIDs, the verifier checks the DID document from a verifiable registry and the validity of the signatures based on the information of the DID document. [119]

2.2.1.4 Digital Wallets & Digital Agents

A digital wallet serves the same purpose as a physical wallet in the real world. It stores credentials, DIDs, cryptographic keys, receipts, and other sensitive data. In addition, it protects them, and makes them accessible. The two core functionalities are key management and encrypted storage. The key management manages the generation, rotation, revocation, storage, signing, and protection of cryptographic keys. There exist multiple kinds of wallets. The first one is called server-side wallet, custodial wallet or cloud wallet, which are hosted by a provider. Another type is the client-side wallet, non-custodial wallets or edge wallet, which is self-managed and could be, for example a hardware wallet or an app running on a user's smartphone. [87, 91]

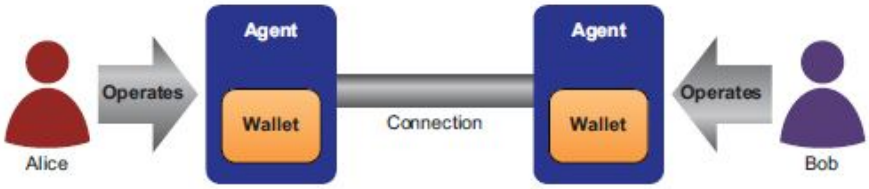

Figure 2.10: Overview of relations between user, wallet and agent [87]

The software operating a wallet in the context of SSI is called digital agent. The

relations between users, wallets and agents is shown in Figure 2.10. There are four core functionalities of an agent (i) messaging, (ii) routing, (iii) backup and recovery, and (iv) secure storage, where secure storage functionality is ensured by the wallet. Some concrete tasks are the initiation and negotiation of DID-to-DID connections, requesting issuance of a verifiable credential and sending digitally signed messages. These agents communicate via secure messaging protocols such as DidComm [28]. There exist two categories of agents, edge agents and cloud agents. Edge agents operate at the edge of the network, on an identity holder's local devices, such as mobile phones. Whereas cloud agents operate in the cloud and different providers are hosting them. Furthermore, they can be used to store and synchronize additional data, such as files or medical records. In contrast to traditional cloud storages, the data is all encrypted by the identity holder. Such storages are also called "secure data storage". [87, 91]

2.2.2 eHealth Applications

Three methods for using self-sovereign identity in digital health systems are described below. These approaches aim to address the first research question of this master's thesis (**RQ1**) regarding the usage of SSI in the context of healthcare. The first method utilizes verifiable credentials directly for storing health records. The second option is to use SSI to represent information that maintains integrity and relates to health data, such as hash values. This approach is used for example by Tcholakian, et al. [111]. In the last approach a Decentralized Web Node is the main component of the architecture, which utilizes DIDs and VCs. The functionality of this component is described in the corresponding section below. In all three approaches we visualized and described a basic workflow of receiving and sharing data between a patient and a healthcare provider. These descriptions are simplified versions of the processes used to understand the differences between approaches and are based on our own definition of the process.

There are many other approaches based on SSI, such as the approach by Belchior, et al. [10], which uses SSI as an authentication method and an additional access control engine to provide access to the requested resource. A concrete approach, using such a concept, was proposed by Saidi, et al. [97]. There are some cases where this concept is helpful, but in our scenario the data would again reside with an external entity.

2.2.2.1 Medical Records as Verifiable Credentials

As described by Kurshid, et al. [69] and Harell, et al. [52] it is possible to store information relevant for healthcare directly in VCs. In their solution the credentials themselves, such as the health ID credential or the health insurance credential, contain the relevant information stored in a digital wallet on a mobile device. This approach allows a patient to share all, or a subset of the information stored in a VC with healthcare providers. A signature at the end of the credential provides authenticity, as the credential was issued and signed by a healthcare provider.

Basic Workflow of Creating and Sharing of Credentials

The sequence of such a process, as visualized in Figure 2.11, is similar to the one described by Harell, et al. [52]. First, the patient and healthcare provider establish a secure connection. Using a DID-based protocol for communication, such as DidComm [28], allows the patient and healthcare provider to establish a secure connection based on their DIDs and communicate over that connection. To establish the connection, the patient scans a QR code provided by the healthcare provider. Once the connection is established successfully, the receptionist issues the patient's wallet with the credentials. These credentials are signed by the healthcare provider and bound to the DID provided by the patient. Additionally, the wallet software verifies the credentials before saving them.

The process of sharing credentials with another healthcare provider begins in the same way as the previous one. This process is visualized in Figure 2.12. The patient establishes a secure connection with the healthcare provider. Next, the patient can select the credentials/claims they want to share on their mobile wallet. Using verifiable presentations of the credentials enables the patient to share the information, which can then be verified by the healthcare provider. The workflow described is simplified and involves multiple parties, including the patient's wallet and digital agents communicating in the background to establish connections and exchange credentials.

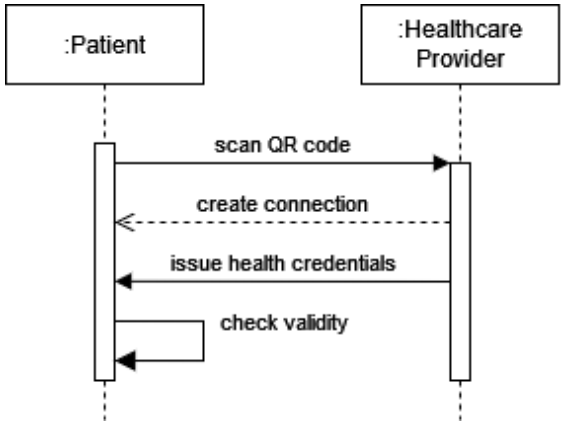

Figure 2.11: Medical records as verifiable credendtials – Basic workflow of patient receiving credentials

Advantages of Medical Records as Verifiable Credentials

- **Offline secured in own wallet:** From a security perspective, this concept offers an attacker few opportunities to access patients' health data. Due to the lack of a centralized storage location, the data can only be obtained by hacking the wallet of each patient individually, or the underlaying device So.

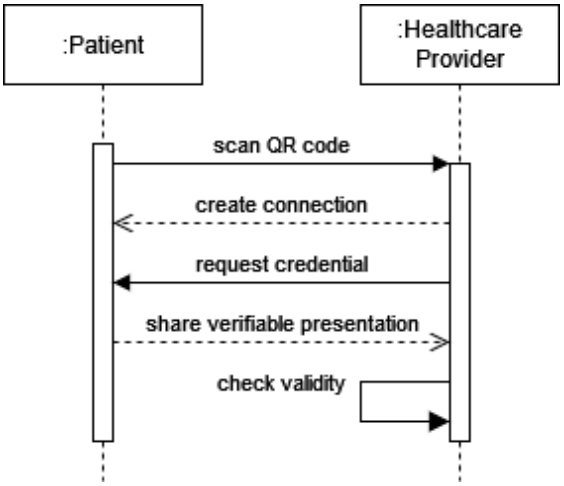

Figure 2.12: Medical records as verifiable credendtials – Basic workflow of patient sharing credentials

- **Fully self-sovereign:** There is no mandatory dependence on external service. All the data is controlled and stored by the patient.

- **Signed and trustable:** The credentials containing the health data are signed by the healthcare provider and bound to a specific DID. This allows an entity to easily check the authenticity of the provided information.

- **Selective disclosure:** The property of selective disclosure allows a patient to decide not only which credential to share, but also which individual information is included.

Disadvantages of Medical Records as Verifiable Credentials

- **Backup:** A medical record should be designed to last a long time. There is no backup mechanism in the proposed concept. The data is stored in an offline storage, which leads to data loss as soon as the device is defective.
  Therefore, it is necessary to consider the issue of backup when developing the architecture or encourage to help the users to use backup mechanism, which might very likely to be useful for other applications too.

- **Unstructured data:** Verifiable credentials are well suited to store structured data such as texts. They are not suitable for saving unstructured data such as images or videos. Therefore, we will need an additional approach for this type of data.

2.2.2.2 Medical Records Using SSI and Distributed File Storage

The concept described by Tcholakian, et al. [111] combines SSI and a distributed file storage system, more precisely IPFS ([11, 61]). In their approach, IPFS stores the encrypted health data and SSI is used to establish a mutually authenticated channel by using decentralized identifiers. Additionally, a blockchain is used to maintain integrity and store the hash of the data. In an improved version, as described in the next paragraph, VCs maintain integrity. This credential is issued by the healthcare provider and contains the hash of the file or data. This solution provides authenticity without the use of a blockchain by sharing the decrypted file together with a VP of the credential.

Basic Workflow of Creating and Sharing of Credentials

The sequence of creating (shown in Figure 2.13) and sharing (shown in Figure 2.14) the health records will be related to the one described by Tcholakian, et al. [111]. The patient and the healthcare provider start to establish a connection based on their DIDs by scanning a QR code at the healthcare provider. For this process, the healthcare provider and the patient utilize a DID based protocol such as DidComm [28]. Optionally, the healthcare provider and the patient create new DIDs for this specific communication channel and exchange them. Compared to the approach by Tcholakian, et al. [111], we will not use a blockchain for maintaining integrity of a file. Instead, we will directly use verifiable credentials for integrity. We can accomplish that as described in chapter 8.2 of the W3C VC specification [119]. There, the W3C provides two possibilities, either using a hashlink [107] or IPFS link. Both ways provide content integrity protection. The hashlink allows verifying the integrity of a file stored on an online location. The IPFS variant allows to verify the integrity of a file stored in IPFS. When the patient wants to store a newly issued medical record, it receives a VC with either a signed hashlink or an IPFS link property. In the IPFS variant, the healthcare provider stores the medical record as an encrypted file on IPFS, using the patient's public key from the DID document. With the hashlink variant, the healthcare provider could store the encrypted file in any location, for example a centralised database.

To share a medical record with a healthcare provider, the patient utilizes a cloud agent, as this approach using an edge-agent solution has multiple shortcomings. Suppose a patient wants to share MR images with a hospital. In total, these images have a size of multiple gigabytes. To share them with an edge client on a mobile phone, the user must retrieve the images, decrypt them, and send them to the agent of the hospital. This procedure loads the data quota in the patient's mobile network with twice the amount of data of the MR images. Moreover, during the entire process, the smartphone must be continuously online to share the data. Often the patient's mobile connectivity is limited in hospitals, making this approach inefficient. Besides, decrypting several gigabytes of data with a public key decryption algorithm could be computationally intensive for an older generation smartphone. With an edge agent, the patient can authorize the cloud client to send the images. There are alternative solutions for sharing the records, such as Sharma, et al. [101], with the help of proxy re-encryption ([12]), but these solutions will

not provide any benefit to our application. When the patient wants to share a medical record, the patient initiates the creation of a connection between the cloud agent and the agent of the healthcare provider. The patient authorizes the edge client to send the file. The edge client loads the encrypted file from the storage location, decrypts it, and sends the file together with a VP of the signed credential over a secured connection to the healthcare provider. The healthcare provider verifies the validity of the VP and the health record and is then able to use the provided data.

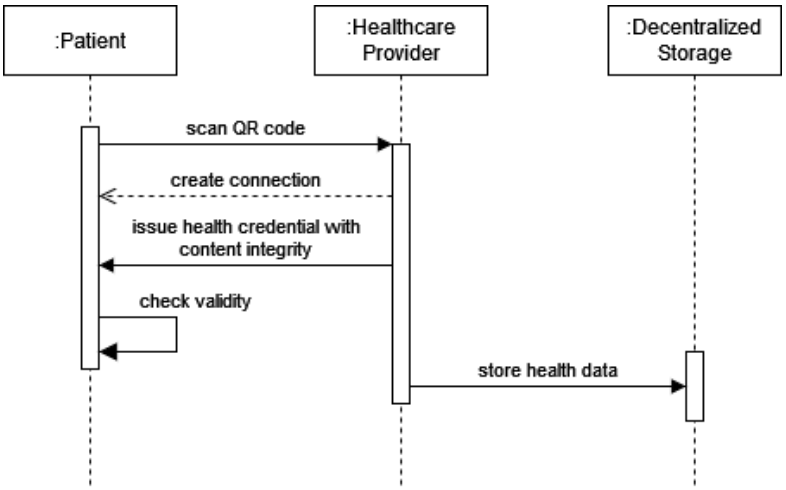

Figure 2.13: Medical records using SSI and distributed file storage – Basic workflow of patient receiving credentials

Advantages of Medical Records Using SSI and Distributed File Storage

- **Backup & Redundancy:** The patient's health data is stored in an online location, thus the medical data is not affected by a loss of the smartphone, compared to the first approach. Additionally, redundancy mechanisms could be used for backups.
- **Signed and trustable:** A signed verifiable credential containing an IPFS link or hashlink with the location of the file provides integrity of the medical record. The VC is signed by the healthcare provider and bound to a specific DID. This allows an entity to easily check the authenticity of the provided information.
- **No datatype restriction:** This approach allows a patient to store all types of files (structured and unstructured) as health records.
- **Selective disclosure:** The property of selective disclosure allows a patient to decide not only which credential to share, but also which individual information is included. However, the property only applies to verifiable credentials and files which support this property.

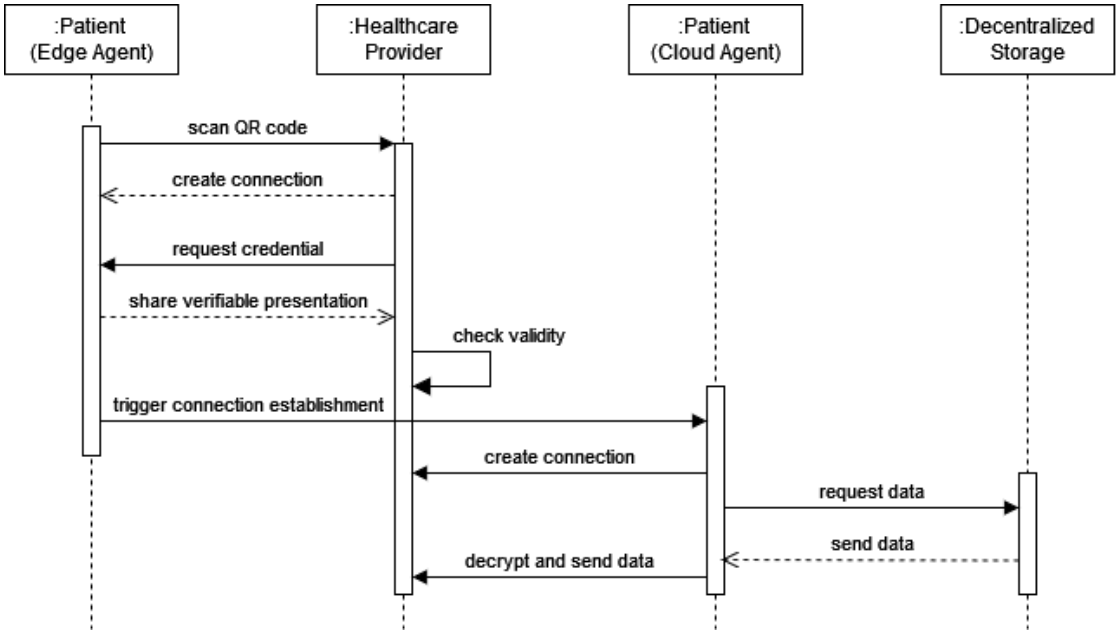

Figure 2.14: Medical records using SSI and distributed file storage – Basic workflow of patient sharing credentials

Disadvantages of Medical Records Using SSI and Distributed File Storage

- **Attack surface:** The medical records are stored in an online accessible location. An attacker can now access the publicly available and encrypted files. Despite the encryption, the attack surface is increased, and the security of the system is decreased.

- **Data availability:** The IPFS approach is decentralized and is more independent from a provider than a centralized server, but a redundancy mechanism must be installed to always be able to retrieve the data.

- **Not completely self-sovereign:** In this approach the patient is dependent on a service, e.g. IPFS, hosting the data. This could violate the principle of "Access" in the context of SSI, which denotes that a user always has access to their digital identity data without any restrictions.

- **Additional encryption layer:** The data stored on a public IPFS system could be accessed by any entity. To secure data access, the data itself has to be encrypted, which results in additional complexity.

2.2.2.3 Medical Records using Decentralized Web Nodes

As described by Preukschat and Reed [91] a digital wallet in the context of SSI is not designed to store an unlimited amount of data, including a lifetime record of financial records, educational records or health records. However, to make this possible, there exists the concept of secure data store (SDS) or "encrypted data vault". The basic idea of SDS describes an "electronic file storage (typically in the cloud) accessible only to the controller because the contents are encrypted with private keys from the controller's digital wallet".

The Decentralized Identity Foundation (DIF) created a draft of a potential specification for encrypted data vaults [115]. In this draft, the encrypted data vaults fulfill the low-level encrypted storage role in a bigger architecture, as shown in Figure 2.15. Due to the concept being unfinished, we will not use the encrypted data vault and focus on the superordinate component, the Decentralized Web Node (DWN), also called Identity hub in previous specifications.

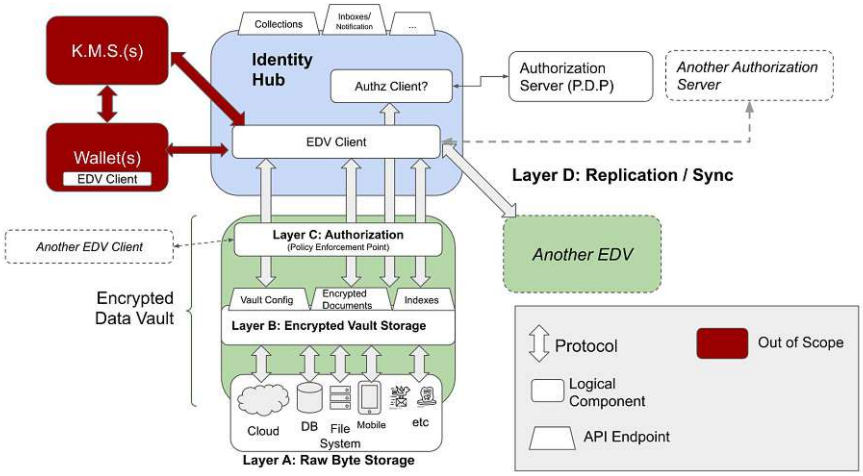

Figure 2.15: Architectural overview of encrypted data vaults [115]

A DWN is "a decentralized personal and application data storage and message relay node" according to the specification draft of the Decentralized Identity Foundation [29]. Preukschat and Reed [92] mention that DWNs do not directly belong to one specific identity controller, but are independent services that manage data on behalf of any number of identity controllers. DWNs are mostly represented as passive and permanently available responders, similar to specialized web servers, with the functionality of data storing and sharing.

The endpoint for accessing a user's Decentralized Web Node (DWN) is determined through the DID document. A patient's DID document is required to contain the "service" property, which includes essential details as the "serviceEndpoint". To specifically receive a person's DWN endpoint, the DID-Relative URL must incorporate the "service" parameter set to "DecentralizedWebNode". The communication between the entities

and the DWNs is accomplished by messages. These messages can cover additional requirements such as encryption or signatures and have Response Objects to respond to incoming messages. [29]

The DWN specification [29] supports customizable features via the Feature Detection Interface, which includes (i) Records, (ii) Protocols, (iii) Permissions, (iv) Hooks, and (v) Sync interfaces. The Records interface (i) enables structured data storage according to shared schemes, ensuring uniformity in schema usage for users in specific areas. While Records allow owners to write isolated records, it is not sufficient for supporting decentralized apps. Protocols (ii) introduce a method for defining rules in apps or services, covering record organization, relationships, data requirements, and participant interactions. This promotes interoperability across diverse app implementations. Permissions (iii) allow external entities to request access to a Decentralized Web Node's data and functionality. It uses a capabilities-based system with DID-based authorization and delegation, contingent on owner approval. With Hooks (iv) DWNs enable apps and services to subscribe to specific data in a user's DWN and act in response. Unlike Web Hooks, they also allow for responding to the entity that initiated the data write or modification. The Sync interface (v) and its methods allow replication between multiple DWNs. [29]

Basic Workflow of Creating and Sharing of Credentials

The main component of this approach is the DWN, as shown in Figures 2.16 and 2.17, which represent the simplified workflows of this concept. Initially, the patient and healthcare provider exchange keys. This process is initialized by the healthcare, which generates a QR code containing the DID and provides it to the patient. The edge agent of the patient queries the corresponding DID document that contains the "serviceEndpoint" property from the verifiable data registry and contacts the DWN of the healthcare provider. The healthcare provider's DWN forwards the message to the healthcare provider's edge agent. The edge agent then queries the patient's DID document from the verifiable data registry. Following this key exchange, the patient utilizes the edge agent to create a permission rule for the healthcare provider on its DWN. The healthcare provider's edge agent sends the data to the patient's DWN, which then verifies the data's validity.

The process of sharing in this approach is similar to that of storing. The patient and healthcare provider exchange keys, as described in the creating process in the previous paragraph. Then, the patient creates a permission rule for the healthcare provider for a specific health record on their DWN. The healthcare provider's edge agent queries the patient's DWN for the record, and the DWN responds with the corresponding health data. The edge agent verifies the validity of the data and provides it to the healthcare provider for further processing.

Advantages of Medical Records using Decentralized Web Nodes

- **Backup & Redundancy:** The patient's health data is stored in an online location,

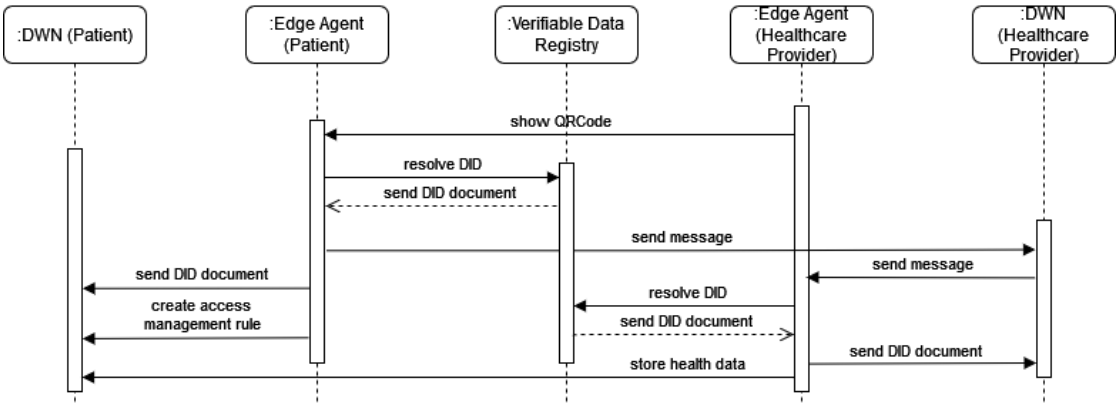

Figure 2.16: Medical records using Decentralized Web Nodes – Basic workflow of patient receiving credentials

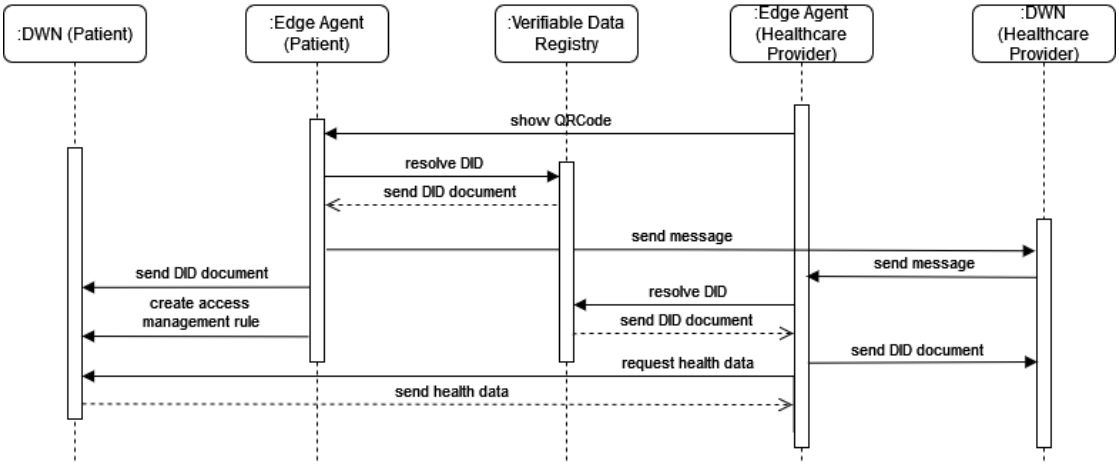

Figure 2.17: Medical records using Decentralized Web Nodes – Basic workflow of patient sharing credentials

thus the medical data is not affected by a loss of the smartphone, compared to the first approach. Some solutions of DWNs provide a build in backup feature.

- **Signed and trustable:** The credentials containing the health data are signed by the healthcare provider and bound to a specific DID. This allows an entity to easily check the authenticity of the provided information. Additionally, a verifier could validate the integrity of a file by using verifiable credentials containing a hash of the file.
- **No mandatory dependence on external service:** There exist solutions of service provide, hosting DWNs. However, the concept of DWNs allows users to host their own DWNs, which makes this approach fully self-sovereign, and a user is not dependent on external services.

- **No datatype restriction:** DWNs, are designed as long term storage for all kind of files. This does not limit the storage to verifiable credentials only.
- **Selective disclosure:** The property of selective disclosure allows a patient to decide not only which credential to share, but also which individual information is included. Additionally, this feature could be extended to any kind of file, if they support it.

Disadvantages of Medical Records using Decentralized Web Nodes

- **Attack surface:** A DWN is most of the time available, which gives an attacker more possibilities to gain unauthorized access than in the "Medical Records as Verifiable Credentials" approach and increases the attack surface.
- **Data availability:** Based on the architecture, a DWN is mostly running on a server. As a consequence, the data is available as long as the server is reachable.
- **Additional component:** Compared to the "Medical Records as Verifiable Credentials" approach, an additional component is needed for the architecture.

2.2.3 Standards and Vulnerabilities

The concept of SSI is based on the components of decentralized identifiers and verifiable credentials, standardized by the W3C Credentials Community Group [113] and W3C DID Working Group [114]. In 2020 Halpin [50] published a paper criticizing a concept, which utilizes the technology of SSI for the implementation of immunity passports. Among other things he criticizes the implementation of the DID and VC standards published by W3C. In this section, we will analyze the technical details of DID and VC based on Halpin's criticism. This will help to look for vulnerabilities in the design process of the architecture. The potential attacks on SSI systems and their security risks, as described for example by Naik, et al. [79] and Gruener, et al. [48], will be discussed in detail in Section 4.3.2 in the context of the architecture.

2.2.3.1 W3C Verifiable Credentials

The criticism of verifiable credentials by Halpin [50] refers to the W3C Verified Credential Data Model 1.0 standard. On the third of March 2022 W3C published version 1.1 of the verifiable credentials data model, but this does not have an impact on the criticisms. Halpin highlights the risk of possible signature exclusion and signature replacement attacks when using verifiable credentials. The problem arises when using the Semantic Web RDF format for normalizing the claims of the VC. RDF uses a semantic graph based data model for making claims about the world allowing for multiple serializations of the same graph [50, 117]. A prerequisite for a signature, however, is a uniform serialization [19]. Additionally, Halpin notes the absence of a standardized approach to handle "blank nodes"

across serialization formats. When Halpin published the paper, he initially referenced the "RDF Dataset Normalization" specification, which is no longer available. The reference now points to a new W3C working draft called "RDF Dataset Canonicalization" [117]. This updated specification claims to transform input datasets into a serialized canonical form and introduces stable identifiers for blank nodes. Although this canonicalization approach appears to address the concerns raised by Halpin, including the identification of blank nodes and the serialization of graphs, the document is still marked as a working draft and therefore not an official specification. Another problem Halpin raised in relation to canonicalization is the NP-completeness of the graph isomorphism [6] problem. This signifies that even if a graph is canonicalized and then serialized, there is no efficient way to determine if two graphs match efficiently.

The upcoming version of the verifiable credential (version 2 [120]) specification is considering adopting the "RDF Dataset Canonicalization" approach. The W3C has released several interconnected specifications. Specifically, the verifiable credential specification relies on the JSON-LD specification, which is a concrete RDF syntax [116]. Furthermore, the "RDF Dataset Canonicalization" outlines the process of achieving canonicalization for an RDF dataset. To address the concerns raised by Halpin, each specification needs to be updated and cross-referenced with the latest versions. In doing so, we must also evaluate the remaining risk associated with the NP-complete graph isomorphism problem. This involves determining whether efficient algorithms exist to handle the problem for a specific credential size, ensuring the application of JSON-LD and RDF dataset canonicalization is secure.

In the same paper, Halpin [50] proposed an alternative solution to this problem. He suggests that verifiable credentials can be serialized using IETF JSON Web Tokens (JWT) instead of RDF, avoiding the mentioned issues. However, Halpin cites Jager, et al. [64] mentioning, that multiple JWT implementations allow misuse cryptography and create the opportunity for cryptographic downgrade attacks. Jager, et al. [64] also mentions practical and effective countermeasures for this concern. The most straightforward countermeasure is to disallow legacy algorithms, but this could potentially disrupt interoperability for parties unable to automatically update and adopt the latest encryption/signing methods. Another solution involves implementing the principle of key separation, using different keys for different purposes. Specifically, the system using verifiable credentials should use a different key for each algorithm. This can be achieved by creating one key, k, and having the user generate a derived key from k for each algorithm using a pseudorandom function and the algorithm identifier.

2.2.3.2 W3C Decentralized Identifiers

Halpin's [50] first technical vulnerability concerns privacy, as in most DID implementations, it is feasible for anyone to retrieve the identifiers without an access control mechanism in place. The design of the blockchains may allow correlation attacks that can be conducted by anyone, primarily due to the public accessibility of DID documents. In specific, anyone can search for times of changes to DIDs and DID documents. Correlation attacks are

attacks, which analyze the relationship between inputs and outputs of the system to gain information [75]. For example, a malicious actor might measure variations in the execution time of a system and try to correlate this information with the used inputs. Halpin refers to a document from Microsoft's ION identity system, which customized the DID implementation, such that this kind of problem is solved by using the sidetree protocol [30, 32].

Additionally, Halpin [50] mentions that privacy is at risk when a DID document contains personally identifiable information. This arises because the data is written on a blockchain and is readily visible to the public. However, DIDs should not be used for this purpose, but for storing public keys. Halpin also suggests this as a countermeasure. But he gives the example that the addition of service endpoints for COVID-19 testing center would leak the information that the person owning the public key has been tested for COVID-19. Although, this example is correct, in general, it is not necessary to update the DID document on the blockchain for each connection, which decreases the chances of success regarding such attacks.

Besides these issues, Halpin [50] faults the lack of support of zero-knowledge proofs for DIDs in the actual standard. Furthermore, he criticizes the usage of a blockchain for DIDs. We will not go into these criticisms here, as they are not essential to this thesis and do not serve to solve security risks.

2.3 Legal Framework

This section serves to clarify the legal regulations based on the European data protection laws in relation to SSI and health data. Furthermore, this section serves as the basis for evaluating our architecture for Section 4.3.4. We analyze how to administer medical data to create a privacy compliant solution and address the second research question (**RQ2**). In the first section, we mention and analyze all important articles and paragraphs of the General Data Protection Regulation (GDPR), which are relevant to this thesis on a technological level. In the following section we analyze how the GDPR can be applied to SSI and Decentralized Web Nodes, how compatible it is with the technologies and what conflicts exist. The last part summarizes the findings and describes what is important for the architecture.

2.3.1 General Data Protection Regulation

In May 2018, the GDPR (General Data Protection Regulation) [38] was introduced by the European Union to protect natural persons in relation to the processing of personal data in Europe, according to Article 1 GDPR. The Regulation introduced multiple principles that define how personal data must be managed when processed wholly or partly by automated means and what rights a natural person has to its data. Article 4(1) of the GDPR defines an "identifiable person" as a person who can be identified directly or indirectly. This includes a reference to an identifier such as a name, an

identification number, or an online identifier. “Personal data” is any information related to an identified or identifiable natural person. In the context of this this, this paragraph identifies any health data as personal data, if it could be associated with a natural person. The term “processing” in the context of GDPR Article 4(2) describes any operation, which is performed on personal data, including collection, storage, and retrieval.

A “controller” (Article 4(7) GDPR) is an entity, which determines the purposes and means of the processing of personal data alone or jointly with others. Furthermore, a “processor” (Article 4(8) GDPR) is any entity processing the data on behalf of the controller. According to Article 26(1) GDPR, if two or more controllers jointly determine the purpose and means of processing, they are joint controllers, and thus it must also be determined who has what obligations under this Regulation. This becomes relevant in a decentralized system, where multiple parties are involved for data processing.

Article 5 and 6 of the GDPR set out the principles and lawfulness of the processing of personal data. The processor and controller must ensure that the users’ personal data is used in a lawful manner and only for specified, appropriate and legitimate purposes. Furthermore, the data must be stored only in a form that allows identification, as long as it is necessary for a purpose. In total Article 5 of the GDPR defines seven key principles when processing personal data, which were described and refined in the further Articles. These seven principles are (i) lawfulness, fairness, and transparency, (ii) purpose limitation, (iii) data minimisation, (iv) accuracy, (v) storage limitation, (vi) integrity and confidentiality, and (vii) accountability. Article 6 defines the conditions necessary to accomplish lawfulness of the processing. In the context of this thesis, we assume that the individuals have consented to the data processing, to achieve lawfulness. In Article 6(1)(d) is defined, that the processing of data is allowed to protect the vital interests of the person. This would allow medical staff to access data in case of emergency and will be considered in the design of the architecture. Article 9 defines special categories of personal data, including health data, which prohibits the processing, with exceptions, such as when the data subject has consented to processing.

Articles 12 to 23 in the GDPR define the rights of the data subjects. Article 15, for instance, establishes an individual’s right to access their personal data and acquire information regarding its processing. It also grants them the right to confirm whether their personal data is being processed, with subsequent access to this data and relevant details if processing is confirmed. Article 17 introduces the “right to erasure“ empowering individuals to request the deletion of their personal data from controllers under specific circumstances, enhancing their control over data removal from, for example online services. Furthermore, Article 16 addresses the rectification of inaccurate data, while Article 18 specifies the conditions for the restriction of data processing. Under certain circumstances, an individual can request the limitation of processing for their personal data. Additionally, Article 20 of the GDPR outlines the “right to data portability“ enabling individuals to receive their personal data in a commonly used and machine-readable format, facilitating its transfer to another controller.

According to Article 24, the data controller has the responsibility to implement suitable

technical and organizational measures, ensuring that data processing aligns with GDPR requirements and can be demonstrated effectively. Furthermore, Article 25 should encourage the controller to make appropriate data protection design decisions and set appropriate policies by default. The controller's duty includes utilizing suitable technical and organizational measures, such as pseudonymization, to integrate data protection principles and necessary safeguards in compliance with the GDPR.

The GDPR defines the rights and duties of the processor and outlines the requirements for data processing between the data controller and the processor in Article 28. A controller must choose processors, which provide appropriate measures when processing data. Furthermore, the processor is just allowed to process the data as instructed by the controller. The authorization or declaration of consent should be made in writing and contain necessary information such as the duration and purpose of processing. In addition, the processor must follow documented instructions, comply with confidentiality and the laws set out in the GDPR, and implement security measures. The controller is responsible for ensuring the processor complies with technical and security requirements. Furthermore, the processor must delete all personal data after concluding the provision of processing-related services.

Article 32 further states that data controllers and processors must use appropriate protective measures according to various requirements at the technical and organizational levels. This guarantees the confidentiality, integrity, and availability of personal data, among other aspects. In accordance with Article 30, the controller is obligated to maintain a register of all processing activities, containing specific information as defined by the GDPR.

2.3.2 Technologies

This section examines the compatibility between the technologies and the GDPR by assigning the roles defined in the GDPR to the entities of the various technologies. We will use the outcome of this part in Section 4 to evaluate the compatibility of the technologies used in the proposed architecture.

2.3.2.1 SSI

The fundamental idea of the GDPR was "the protection of natural persons with regard to the processing of personal data" [38]. In situations characterized by centralized architectures, where roles are distinctly assigned, and applications are in control of the data, as is frequently encountered, the GDPR indeed offers advantages to individuals. However, as mentioned by Penedo [20] and Kondova and Erbguth [71] the approach of SSI does not fit this model. SSI is a decentralized concept, which establishes equality between all the actors involved. Therefore, it is not clear which actor has which role in an SSI based architecture and it depends on the specific scenario. The explicit definition of roles and classification of data, whether it is personal data or not, is essential to apply the GDPR. To help classify roles and apply the GDPR to SSI, the European Data

Protection Board published a guideline named “Guidelines on the concepts of controller and processor in the GDPR“ [14].

Controller
To define the role of a controller within the scope of GDPR, the Guidelines [14] outline five key considerations, as explained by Penedo [20]:

1. **Nature of the Entity**: Firstly, the type of entity that can assume the role of a controller is examined. Article 4(7) of the GDPR defines this entity as a “Natural or legal person, public authority, agency, or other body“. Therefore, any entity can potentially act as a controller.

2. **Influence Over Processing**: Secondly, the extent of influence of the controller over the data processing activities is assessed. Controllers are typically those entities that exert a significant degree of influence, both in terms of decision-making and execution.

3. **Sole or Joint Determination**: The third consideration is whether the determination of control is conducted solely or jointly with others. This determination can vary depending on whether multiple entities share responsibility for data processing.

4. **Purpose and Means Determination**: Controllers are responsible for determining both the purpose and means of data processing. This includes not only why the data is processed but also the methods employed in the process.

5. **Process Determination**: Lastly, the determination of the entire data processing process itself is examined. Controllers are the entities that have the authority to shape and oversee this process.

As already mentioned, the GDPR is designed for data privacy in interactions between two or multiple parties. In most scenarios this includes a subject, and a controller or processor. This is not clearly defined in the GDPR, but could be concluded from various articles and paragraphs, such as Article 2, Article 4(1), and Article 4(7). However, what remains uncertain is whether an entity can simultaneously function as both a “data subject“ and a controller. Penedo [20] concludes that the GDPR can thus not be applied if only the own data is processed, because of the GDPR’s purpose to protect others interest, for example the data subject. As a result, a person cannot be the controller of their own data and identities in the context of GDPR. Other entities building and operating other peoples’ identities can be data controllers and processors, based on the situation.

Processor
In contrast to the controller, the processor could not process personal data on its own, but on behalf of the controller. As the “Guidelines on the concepts of controller and

processor in the GDPR“ [14] explains, a processor could be, like the controller, “a natural or legal person, public authority, agency or other body”. Penedo [20] mentions that every entity processing data in an SSI based system, e.g., the validation of VPs, could be seen as processor. This is because none of these activities implies the determination of purposes and means. As mentioned in the guidelines [14] “The role of a processor does not stem from the nature of an entity that is processing data but from its concrete activities in a specific context“. In specific, an entity could be a processor, or a controller based on the situation. In general, a processor needs a controller, which delegates instructions regarding the processing to the processor. If the processor is going against the instructions, it becomes a controller.

GDPR Analysis

To analyze and apply the GDPR on SSI we assigned the GDPR roles to the participants of a SSI system, as shown in Table 2.1. A holder in the context of SSI managing its own identity and credentials as mentioned, in the previous sections, could not be a controller or processor in the context of GDPR. Therefore, a holder does not have a defined role in the context of GDPR and the GDPR is not applicable to the holder. Other components that process data for the holder, for example an agent, can be a processor or controller. An issuer and verifier take the role of a controller or processor, based on the scenario. This is because both work with personal data, which identify the holder. The issuer is creating and disclosing data to the holder and the verifier is retrieving and using data. According to Article 4(2) of the GDPR all these activities are defined as “processing”. As Naik and Jenkins [78] mention, there are additional SSI roles, which are necessary to operate a SSI infrastructure, such as “Transaction Authors” or “Node Controllers”. These roles are not necessary to consider within the scope of this thesis. However, we also address the blockchain and violations of the GDPR, as the blockchain is a central component in the SSI system.

SSI Roles	**GDPR Roles**
Holder	No GDPR role
Issuer	Controller/Processor
Verifier	Controller/Processor

Table 2.1: Overview of SSI roles in GDPR

To ensure compliance with the directives, the specific use case or architecture must be evaluated, as well as the specific technology used in the system, e.g., which kind of blockchain. As Naik and Jenkins [78] explain, there exists different types of blockchain, e.g. private or public ones and this may have an impact on the validity of data processing. In the following we analyze the compatibility of issuer and verifier with the GDPR and what these roles have to consider in a healthcare infrastructure. We will use the same structure as Naik and Jenkins [78].

SSI Compatibility With Lawfulness, Fairness and Transparency This GDPR principle defines lawfully, fairly, and transparent personal data processing. Article 6 in the GDPR describes under which circumstances a controller is allowed to process data. From the perspective of an issuer, the main activity is to issue a VC and bind the user's identifier to it. During this process, the holder shares its DID with the issuer so that the issuer can associate its identifier with the data. Only at this point does data become personal data. To achieve this, the holder must already have given his consent. Therefore, lawfulness is given automatically in the issuing process. The same reasoning applies to the verifier. The holder provides a verifiable presentation with its personal data to the verifier, which is only possible with the consent of the holder. Similarly, when it comes to creating a DID on the blockchain, the user provides consent by generating the DID along with the corresponding public key in the DID document. In all three scenarios, the user retains full control over the data, and their consent is a prerequisite for any data processing. This aligns with Article 9 of the GDPR, which mandates explicit consent for processing special categories of data, including health information. Regarding the fairness and transparency, Naik and Jenkins [78] explain, that this has to be evaluated on the specific SSI system.

SSI Compatibility With Purpose Limitation According to Article 4 (b) of the GDPR, data may only be processed for a specific purpose and not used in any other way. As described by Naik and Jenkins [78], a holder can control to whom the personal data is shared, but it cannot control for which purpose this data is used. This must be clarified by the verifier and issuer and depends on the specific implementation of an SSI system. Regarding Article 15 of the GDPR, the "Right of access by the data subject", the verifier must provide information to the holder about how the data is processed, if a holder requests this information. This also applies to the issuer.

SSI Compatibility With Data Minimisation When processing data, the controller or processor in the context of GDPR must only use the information required for the respective purpose. The property of selective disclosure in a verifiable presentation allows a holder of a credential to reveal just a subset of the verifiable data. In an interaction between a holder and a verifier, the holder can select which data to share with an issuer. Hence, a holder can control the data and just provide the information necessary for an issuer. The problem with the data minimization principle arises with the blockchain component of a SSI system, as mentioned by Naik and Jenkins [78]. A blockchain is designed to replicate data to multiple nodes and has the characteristics of append-only. Both properties lead to compatibility issues. However, Giannopoulou [46] explains that the French Data Protection Authority has issued an official opinion on blockchains. This official opinion argues that the structure and functioning of the blockchain is accompanied by a permanent visibility of the keys and "that their retention periods are, by essence, in line with the blockchain's duration of existence". Giannopolou therefore concludes that the principle of minimalization can be observed. However, the General Data Protection

Regulation (GDPR) does not currently address the unique challenges posed by blockchain technology, and therefore is not fully compatible with this principle.

SSI Compatibility With Accuracy This principle ensures the accuracy of personal data. To be more precise, maintaining the accuracy of personal data is essential, and the possibility for both deletion and rectification of any inaccurate data must be made available. The DID specification allows an update functionality for DID documents, which allows to accomplish the accuracy of data and obey Article 16 "Right to rectification". However, as described by Naik and Jenkins [78], this principle includes one of the most important rights in the GDPR, the "Right to erasure" in Article 17. The issuer and verifier can accomplish this by deleting any data related to the holder. In the case of the blockchain used for the DID infrastructure, this article could lead to compatibility issues, because of the append-only property. In an SSI system, the blockchain stores the public keys associated with the DIDs. Based on the definition of "personal data" in Article 4(1) of the GDPR, public keys could be identified as personal data, leading to conflicts with Articles 16, 17 and 18. As the GDPR is currently defined, the blockchain is not compatible with this principle.

SSI Compatibility With Storage Limitation According to the "storage limitation" principle in Article 5(e) of the GDPR, the personal data must not be longer identifying the data subjects, if not necessary. Both the issuer and verifier must consider this requirement, which introduces similar conflicts as those outlined in the preceding principle, especially concerning the blockchain's role in maintaining the SSI infrastructure.

SSI Compatibility With Integrity and Confidentiality The appropriate security measures to protect personal data depend on a concrete SSI system and must be evaluated for the issuer, verifier and the blockchain technology. As described by Naik and Jenkins [78], a SSI system is by design based on cryptography, decentralization and anonymization, which prevent attacks, such as single point of failure.

SSI Compatibility With Accountability The principle of accountability establishes that a controller holds the responsibility and should have the ability to demonstrate compliance with all the other governing principles. The GDPR mentions these responsibilities in various articles of the GDPR, in specific Articles 30 and 32. As in previous paragraphs this principle must be evaluated within a specific architecture. Naik and Jenkins [78] point out that the utilization of blockchain in a SSI based system may introduce difficulties when it comes to ensuring accountability. This complexity arises from the decentralized nature of blockchain and the GDPR's model for defining roles, making it challenging to attribute accountability to a particular entity.

2.3.2.2 Decentralized Web Node

In the context of the GDPR a DWN is easier to classify than the technology of SSI, since a DWN can be seen as an online storage with special access procedures and functionalities. We differ between self-hosted DWNs and DWNs hosted by third-party service providers. With both approaches we have the same uncertainty as mentioned in the previous section of whether an entity can simultaneously function as both a data subject and a controller. This is caused by the household exception mentioned in Article 2 (c) of the GDPR, which states that the GDPR does not apply when "purely personal or household activity" is present. The third-party hosting provider for DWNs is classified as a processor in the context of the GDPR. As mentioned in Article 4 (8) of the GDPR a controller "determines the purposes and means of the processing of personal data". In this case the user has full control of its data and determines the purposes and means of processing. The hosting provider just provides the infrastructure for a user but has no control over the data. A processor "processes personal data on behalf of the controller". Hence, when we assume that the user can be considered as a controller, the hosting provider acts as a processor.

GDPR Analysis

To ensure compliance with the directives, the specific use case or architecture must be evaluated, as well as the specific technology used in the system. In the following we analyze the compatibility of DWNs with GDPR and what to consider in a healthcare infrastructure. This analysis is under the assumption that the user is classified as a controller in the context of the GDPR and the hosting provider as a processor. If the user were not classified as a controller, there would be no GDPR role assignments for DWNs and therefore no restrictions imposed by the GDPR.

The hosting provider's task is running the DWN, ensuring it is available to the user, and securely storing the encrypted user data within it. The users can choose a hosting provider and check the technical and security measures beforehand. In terms of purpose limitation, the provider has no impact on the processing itself as the hosted DWN is fully controlled by a user and not accessible to unauthorized third parties, such as the provider. Only the patient with the appropriate DID can access the personal data stored in the DWN. Every instruction from the user will be performed on the DWN without the processor having a direct impact. The only possibility to violate the instruction of the user is to shut down the DWN and keep the DWN and the data stored inside the DWN. Nevertheless, the provider will not be able to get access to the encrypted data. A contractual agreement with the provider, as recommended by the GDPR, can ensure the declaration of consent. The appropriate security measures to protect personal data depend on the concrete service provider. In general, the processor does not have direct access to the data and is therefore not able to handle data improperly. The provider's responsibility primarily lies in preventing unauthorized access to the entire DWN component. This fulfills the confidentiality and security requirements outlined in Article 28 of the GDPR.

2.3.3 Outcomes

Table 2.2 shows an overview of considerations and implications of the GDPR for SSI technology. Open-source components help with both the technological aspects and compliance with the principles of the GDPR. To strengthen trust and transparency. Open-source components allow the patients to comprehend how the system works and brings fairness and transparency to the users. However, this does not accomplish the first principle of lawfulness, fairness, and transparency by default. Also, for other principles such as purpose limitation, accountability, integrity, and confidentiality, can open-source components help to accomplish the goals. In terms of purpose limitation, the verifier or entity accessing the data must prove how data is processed, for example by using a logging mechanism. This is especially important when the patient allows permanent access to the data. Moreover, the request for data access should include the specific reason for the request, which can be recorded in the access permission. Both the issuer and verifier are obligated to delete the data if it is no longer required for its intended purpose, unless other legal obligations, such as retention requirements, apply. The append-only characteristics of blockchains and the synchronization of multiple nodes could conflict with the principles of data minimization, accuracy, and storage limitation. This technology is an edge case in the GDPR, which must be evaluated by the European Union. Regarding the principle of accuracy, the chosen DID method must include the DID update functionality to comply with Article 16 "Right to rectification". It is important to note that the append-only feature of blockchains could potentially conflict with Article 17, the "Right to erasure", as well as Article 18, the "Right to restriction of processing". Both the issuer and the entity accessing the data must consider these three articles (16, 17, 18). About the principle of storage limitation, both the issuer and the verifier must make sure that the data no longer identifies the data subjects except for obeying other laws such as retention obligation. Each entity within the system must implement appropriate security measures to comply with the principle of integrity and confidentiality. These measures must be evaluated on the specific architecture. The issuer and the verifier must demonstrate compliance with all the other governing principles. As in other principles blockchains could conflict with the accountability principle because it is challenging to attribute accountability to a particular entity. However, as mentioned by Naik and Jenkins [78] the insight of the data transaction could provide some kind of accountability to the users. Furthermore, using a governance framework for blockchains, and issuer and verifier could provide greater accountability.

When patients decide to choose a third-party hosting provider for their DWNs, they must consider the following aspects. Firstly, the provider must obtain explicit consent from the controller to process the data, which can be granted through written or digital confirmation specifying the permitted processing operations. Additionally, the hosting provider should maintain documentation outlining the purposes of processing, which can be produced if requested by the controller. The hosting provider must implement state-of-the-art security measures. However, it falls on the user or data controller to verify their implementation. Additionally, the hosting provider must guarantee confidentiality

by limiting access to the DWN. Users can gain access, for instance, through REST calls. Access to the entire DWN component, such as via a Docker container, should be restricted for all, as there is no need. At the end of the hosting process, the hosting provider must delete all personal data.

GDPR principle	Considerations and impact on the architecture
Lawfulness, fairness and transparency	• Usage of open source components for traceability of the system • Evaluation on concrete implementation
Purpose limitation	• Issuer and verifier need a prove to show how data is processed • Implement mechanisms such as credible logging mechanism • Issuer and verifier must delete the data if no longer used for the respective purpose
Data minimisation	• Append-only characteristics of blockchains and the synchronization of multiple nodes could conflict with GDPR
Accuracy	• DID method has to include DID update functionality • Append-only characteristics of blockchains could conflict with GDPR, especially Article 17 and 18 • Issuer and verifier must implement functionalities to obey Articles 16, 17, and 18
Storage limitation	• Append-only characteristics of blockchains and the synchronization of multiple nodes could conflict with GDPR • Issuer and verifier must make sure that the data no longer identifies patient if not necessary anymore
Integrity and confidentiality	• Every entity must use appropriate security measures • Evaluation on specific architecture
Accountability	• Issuer and verifier must demonstrate the compliance with other governing principles • Blockchains could conflict with GDPR, because it is challenging to attribute accountability to a particular entity • Using governance frameworks for blockchains, and issuer and verifier to provide greater accountability

Table 2.2: Overview of seven GDPR principles to be observed for SSI in the architecture

CHAPTER 3

Related Work

The concept of sharing electronic health data between different healthcare facilities exists since patient data could be digitally documented. Unfortunately, the centralized identity access management systems and architectural system designs make it hard to share this data globally without limitations [102]. With the emergence of bitcoin [80], the topic of decentralization became popular and blockchains opened new opportunities in many different areas, including sharing of data while maintaining privacy. In the following we will describe already existing approaches. The final section of this chapter summarizes the approaches discussed and provides an overview and comparison of both existing approaches and our proposed solution.

3.1 Blockchain Based EHR Approaches

Xia, et al. [122] proposed a solution for blockchain-based data sharing in cloud environments for electronic medical records. The authors use blockchain technology and cryptographic keys to address the issue of managing access for users and their sensitive data, stored in cloud-based data storages. In a paper published later by the same author, Xia, et al. [123] presented MeDShare, which allows sharing of medical data between big data entities in cloud environment using a blockchain and smart contract functionality. The security and access mechanisms are based on public key cryptography and corresponding identities, similar to the first paper. These two solutions provide secure access and sharing mechanisms focused on cloud provider. However, these approaches do not consider smaller local databases and services providers, representing smaller entities as is often the case in the healthcare sector.

In 2016, Azaria, et al. [5] proposed MedRec, a decentralized record management system to handle electronic medical records (EMRs) using blockchain technology. MedRec allows patients to access and share their EMRs, hosted off-chain by the providers, in one decentralized system. The utilization of smart contracts guarantees both access

management and data localization. Two years later, Fan, et al. [39] addressed the same issue of constructing a summarized EMR for one patient from multiple hospital databases with their solution MedBlock. Both approaches (MedRec and MedBlock) use a bread crumb mechanism, which allows to search for medical records on a ledger. The approach proposed by Fan, et al. (MedBlock) achieves security and privacy by using public key cryptography for providing the EMRs stored in the databases of distributed healthcare providers. Data stored on the blockchain is encrypted and protected by a suitable access control protocol. A similar approach with a focus on security and privacy was presented by Zhuang, et al. [124]. They proposed a patient-centric health information exchange framework, which utilizes smart contracts to ensure the security of data and the privacy of patients, while also granting users complete authority over their medical records. The framework's intent is to facilitate appropriate record access throughout various medical institutions, creating an interconnected healthcare system. The actual medical records are encrypted and stored within the healthcare facility's databases. Smart contracts are the central components, which, among other functions, validate the permissions and administer the decryption keys for the data. In 2019 Shen et al. [102] proposed MedChain, an efficient healthcare data sharing approach via blockchain. The architecture of MedChain, as shown in Figure 3.1,is designed as a decentralized network, which connects all healthcare providers. The model includes two types of peer nodes, which represents small and large healthcare providers and their resources. Additionally, the large peer nodes operate three modules, the blockchain service, the directory service, and healthcare database. In MedChain, the blockchain is used to maintain the data integrity, whereas the data itself is stored off-chain in the databases of the healthcare provider.

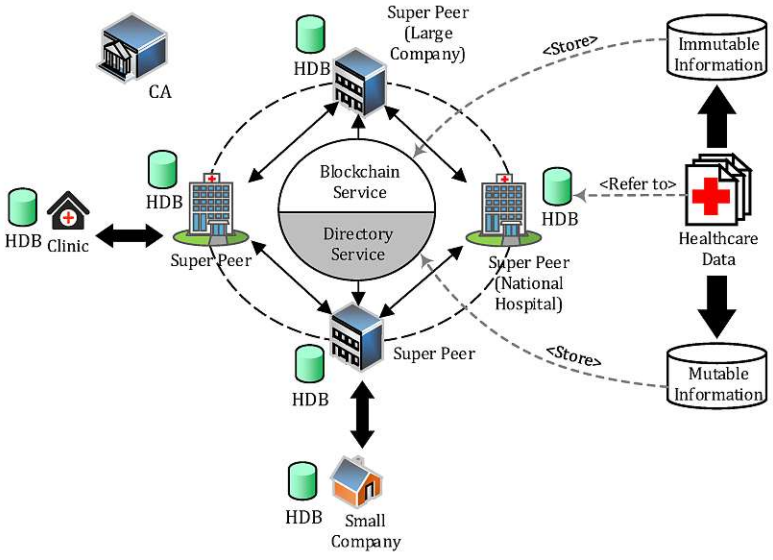

Figure 3.1: Overview of MedChain architecture [102]

The user centric approach proposed by Liang, et al. [73] focuses on wearable technology and the Internet-of-Things in the healthcare domain. In their solution users collect health

data from wearable devices, which are uploaded to a cloud database. The users have ownership of their health data and can grant or revoke access to healthcare providers and insurance companies. The data is stored on a blockchain network for integrity protection and validation. Similar approaches by authors such as Huang et al. [60], Wang et al. [121], and Zou, Lv and Zhao [125], employ blockchain technology to access data located in diverse central databases. All the approaches mentioned utilize blockchain technology to build a summarized EMR and interconnect the healthcare provider in a decentralized way. However, the actual problem of privacy and data ownership is not solved by these approaches. These solutions can be seen as additional layers connecting multiple centralized systems without having full control over the data. The focus is on sharing data across multiple healthcare facilities. Especially the last three approaches (Zhuang et al.[124], Shen, Guo, and Yang[102], Liang, et al.[73]) use the blockchain to store information and maintain data integrity. The use of blockchain to store information, especially sensible data, is a risk to privacy and may lead to conflicts with the GDPR as Gassner [43] proposed.

3.2 InterPlanetary File System (IPFS) based Approaches

As of 2019, some more advanced approaches have been developed that use other technologies besides blockchain, such as InterPlanetary File System (IPFS) [11, 61] or cryptographic solutions such as proxy re-encryption[13]. Nguyen, et al. [81] presented an EHR sharing framework using blockchain and IPFS on a mobile cloud platform. The electronic health records of the users get encrypted and uploaded to IPFS. The keys for encryption and decryption are administered by an entity called “EHRs manager”. This concept do not allow a user to own and manage their own data, but be dependent on the “EHR manager” to gain access to the data, as visualized in Figure 3.2. A more privacy preserving way is described by Sharma, et al. [101]. In their storing and sharing approach, they utilize a proxy re-encryption scheme and Zero-knowledge proofs (ZKP) for accessing the EHRs stored on IPFS. The hospitals get access by sending proofs of their ownership and knowledge to a smart contract. A cloud server, acting as a proxy, performs the necessary re-encryption, which ensures that the cloud server (proxy) does not have access to the plaintext data. In this approach all hospitals can request access by design. This again does not allow users to have full control over their data.

3.3 Self-Sovereign Identity based Approaches

The topic of decentralization has also opened new possibilities in the area of identity and access management. In 2021 Khurshid, et al. [69] proposed MediLinker, a blockchain application using a patient-centric identity management system in healthcare. The initial use cases of MediLinker were enrollment of the user, consenting and sharing of data between clinics and withdrawing access rights to shared data. The application ensures the security and confidentiality of the data and allows patients’ detailed control over what information they would like to share with different providers. In 2022 Harell, et

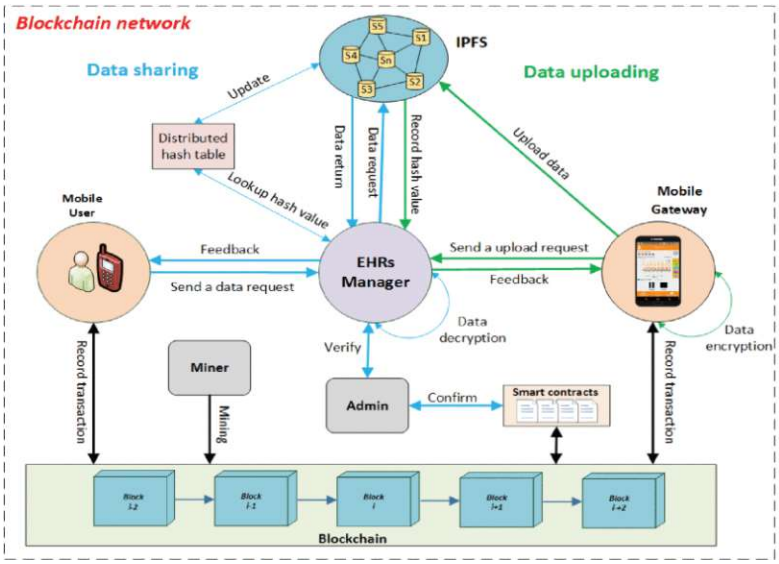

Figure 3.2: Overview of architecture proposed by Nguyen, et al. [81]

al. [52] describe the technical design and development of MediLinker. Compared to Khurshid, et al. [69], Harell, et al. [52] explain the technical details of Medilinker. The approach was built using Ethereum[41] and Hyperledger Indy[42], a blockchain platform specially designed for identity management, and different cloud service providers. The actual health data was stored off-chain in a digital wallet on a mobile device. Only schemes of each credential type and decentralized identifiers (DIDs) of each patient were stored on-chain. The drawback of MediLinker is the limitation regarding the amount of credentials for users. It is focused on providing basic identity information, such as name and birthday, in a machine-readable format, but not general medical data as they occur in EHRs and PHRs.

Bai, et al. [7] proposed a high-level system architecture of the SSI model for smart healthcare, highlighting the integration of Internet of Medical Things (IoMT) devices and the use of a verifiable data registry. The SSI model aims to provide patients with control over their EHR and give them the ability to selectively share their health information. The model integrates the Internet of Medical Things (IoMT) devices, such as wearable sensors and medical equipment, into the identity management system. The smart healthcare system issues verifiable credentials to stakeholders based on Decentralized Identifiers (DID), which allows the data owner to have complete control over the sharing of health data. The communication flow in the SSI model relies on DID communication between Edge Agents of users. The proposed SSI model utilizes blockchain technology, specifically the Hyperledger Indy permissioned blockchain, for storing and verifying the transactions related to identity management. The authors do not mention how exactly the data is stored, especially unstructured data such as large images and videos. The focus is on describing the self-sovereign identity management system for the smart healthcare

system and the area of IoT. Additionally, they do not describe other topics relevant for a potentially live long EHR, such as recovery.

Saidi, et al. [97] created a Decentralized Self-Management of Data Access Control (DSMAC) system that integrates blockchain technology and SSI to preserve privacy and enable patients to control access to their medical data. The system includes three layers, the IoMT Devices Layer, the F2C Computing Layer, and the User Layer. The proposed scheme combines Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) models. Role-Based Decentralized Access Control (RDAC) is used to assign default permissions based on user roles. Attribute-Based Decentralized Access Control (ADAC) is used in emergency cases and considers contextual constraints and attributes. Decentralized Identifiers (DIDs) and verifiable credentials (VCs) are used for user authentication and authorization. The proposed scheme integrates blockchain for storing access control policies, public keys, and credentials. The authors describe a different kind of usage for SSI. Compared to having the data itself as SSI credentials, it offers the possibility of access management using SSI.

In 2021, Javed, et al. [65] proposed a system, called Health-ID, utilizing blockchain technology to provide a decentralized identity management solution for remote healthcare, as shown in Figure 3.3. It involves four main actors: users, healthcare regulators, blockchain, and cloud storage. The architecture allows users to control their identities, while regulators validate the identities and issue attestations. Identity attributes are encrypted and stored on cloud storage, while hashes of the attributes are stored on the blockchain. By using smart contracts, various functions and interactions can be enabled, such as uploading and retrieving identity attributes, registering and verifying attestations, and providing public keys. The architecture provides a decentralized solution for remote healthcare identity management, where blockchain technology ensures the security, integrity, and auditability of identity data. This approach is dependent on smart contracts with the disadvantage that many transactions are written to a blockchain to validate hashes of identity attributes. This leads to high costs and limited scalability.

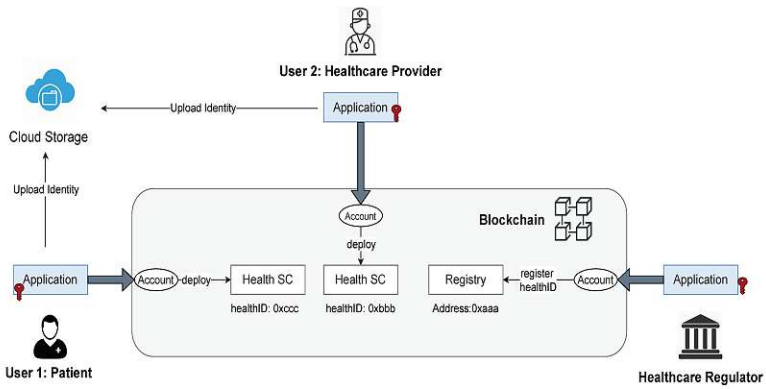

Figure 3.3: Overview of architecture proposed by Javed, et al. [65]

The architecture, proposed by Tcholakian, et al. [111], combines SSI, IPFS (Inter-

Planetary File System), and attribute-based encryption (ABE) [96] to strengthen access management for EHRs. It describes a scenario where a patient arrives at a healthcare institution and establishes a mutually authenticated channel using their verifiable credentials. The architecture utilizes AES encryption for privately storing medical records on IPFS and ABE encryption of AES keys to control access to the records based on attributes. Access policies can be modified by the patient, and access authorization is logged on to the blockchain. The approach of ABE can bring advantages in certain situations, but is inefficient by storing the record, the hash value of the record, the encrypted AES key and the hash value of the AES key to IPFS.

3.4 Existing solutions

There already exist implemented solutions for decentralized architectures in the context of eHealth. One of them is Medibloc [74]. This solution uses the Panacea blockchain and is designed to create a patient-centered health data ecosystem that can protect individual privacy. It uses decentralized identifiers and verifiable credentials and was partly funded by the EU [24]. Other SSI based projects within Europe and EU are eIDAS, the eIDAS Bridge and the European Self-Sovereign Identity Framework (ESSIF) [22]. A European solution for SSI based identities would make it even easier to create a decentralized and connected eHealth infrastructure.

3.5 Overview and Comparison

Considering the current solutions such as the Telematikinfrastruktur [44] in Germany or ELGA in Austria [112], we notice the eHealth infrastructures are using centralized or federated identity access and management systems and are not designed in the most privacy preserving ways possible. Another drawback, besides the privacy, is the missing interconnection between different healthcare systems and facilities. Also, the existing approaches in this field of research do not provide an ideal solution for a medical infrastructure. We have created a comparison of the existing solutions, as shown in Table 3.1. The following categories were selected for comparison because of their importance in the context of medical records in healthcare architectures. This decision was made based on our experience and existing literature.

- **Privacy** including protection of personal health data and restricted access for unauthorized entities
- **Scalability** refers to the ability of solutions to expand and perform well under increased load
- **Functionality** describes the compatibility with any kind of data as necessary for EHRs/PHRs, such as images

- **Consideration of GDPR Compliance** including analyzing compatibility with a specific architecture and its components
- **Generality** describes the compatibility with any kind of healthcare provider
- **Data ownership** means full data control for patients
- **Potentially life long support** describes whether the system was designed to be suitable for long-term storage and could support necessary features such as backups.
- **Efficiency** including performance criteria and additional overhead
- **SSI** describes whether the concept of SSI is used

Categories	Privacy	Scalability	Functionality	GDPR Compliance	Generality	Data ownership	Potentially life long support	Efficiency	SSI
Xia, et al. [122]	~	✓	✓	✗	✗	✗	✓	✓	✗
Xia, et al. [123]	~	~	✓	✗	✗	✗	✓	~	✗
Azaria, et al. [5]	~	~	✓	✗	✓	✗	✓	~	✗
Fan, et al. [39]	~	✓	✓	✗	✓	✗	✓	✓	✗
Zhuang, et al. [124]	✓	~	✓	✗	✓	~	✓	~	✗
Shen et al. [102]	✓	✓	✓	✗	✓	~	✓	~	✗
Liang, et al. [73]	✓	✓	~	✗	✗	~	✓	~	✗
Nguyen, et al. [81]	~	✗	✓	✗	✓	✗	✓	✓	✗
Sharma, et al. [101]	✓	✓	✓	✗	✓	✗	✓	✓	✗
Khurshid, et al. [69] & Harell, et al. [52]	✓	✓	✗	✗	✓	✓	~	✓	✓
Bai, et al. [7]	✓	✓	✗	✗	✓	✓	✓	✓	✓
Javed, et al. [65]	✓	~	✓	✗	✓	~	✓	~	✓
Tcholakian, et al. [111]	~	✓	✓	✗	✓	✓	✓	✗	✓
Proposed solution	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓= mostly comply, ~ = partly comply, ✗= not comply

Table 3.1: Comparison of related work

We chose these categories because they are consistent with the goal of the thesis, which is to create a privacy preserving eHealth architecture. The privacy and data ownership categories reflect the state of privacy in current architectures. To check the effectiveness

and efficiency in the currently available architectures, we use the categories of scalability, functionality, generality, potential lifetime support, and efficiency. The GDPR compliance category checks if the solutions compliy with the GDPR.

Most of the proposed solutions are focused on the topic of privacy. However, by using a blockchain or IPFS for storing personal data, the solutions provide a privacy issue by design, as the data is publicly available (albeit encrypted), as for example in the Tcholakian, et al. [111] and Nguyen, et al. [81]. We follow the approach that data which is not necessary for the architecture is not available. Additionally, one of the biggest issues for blockchains is scalability. Nguyen, et al. [81] for example describes the characteristics of scalability as an open issue in their approach. Even if most of the approaches claim their solutions are scalable, all solutions that rely on blockchain transactions could issues regarding scalability. Nearly all approaches try to create an EHR by connecting the individual data records stored by multiple healthcare providers. All these approaches provide the necessary functionality. In contrast, the existing solutions using a more self-sovereign approach (Hansen, Schwartz, and Cooper[69], Khurshid et al.[52]), provide not the necessary functionality such as storing unstructured data. Our approach will combine the necessary functionalities and self-sovereignty. In general, none of all the mentioned solutions considered the GDPR in their design process, which is part of this thesis. Additionally, our approach can be used by all kinds of healthcare providers and not just by specially selected ones. One of the most important topics in this thesis, also regarding privacy, is data ownership. Many approaches, such as Azaria, et al. [5] and Zhuang, et al. [124], store the data in multiple local databases and restrict access by using different kinds of access mechanisms. However, by having access to the systems underneath the healthcare provider could be able to access data without the consent of the user. With the proposed architecture of this thesis, no one can access the data without the consent of the patient. Furthermore, we consider the topics of longevity and efficiency in our solution. As in some solutions (e.g. Tcholakian et al. [111], Zhuang, et al.[124], Shen, Guo, and Yang[102]), the mechanisms to ensure trust, privacy and security are very inefficient and create a lot of overhead regarding data and computing capacity. In our solution we avoid this overhead by the concept of self-sovereign identity. Section 3.3 describes four approaches that use SSI as part of their architecture. None of the other concepts utilize this concept as part of their solution.

The final prototype of this thesis fulfills all the mentioned criteria. Our solution provides users with full control over their data, which is stored end-to-end encrypted and with appropriate access mechanisms to protect user privacy. Our approach to data storage includes various types of data such as text, audio, video, and images. We use a decentralized approach, utilizing the concept of SSI, to enable distributed storage and improve scalability. Additionally, we are evaluating the compatibility of our prototype with the General Data Protection Regulation and establishing efficient processes. This unique combination of an efficient and SSI-based approach to managing various healthcare data in the healthcare sector has not yet been researched and therefore represents a novel approach for an architecture for medical records.

CHAPTER 4

Evaluation & Results

This chapter presents the evaluation and results of the thesis. It includes all the necessary steps, from creating the requirements for the architecture to evaluating the results. It explains the design decisions made and serves to answer all three research questions. This chapter builds upon the previous ones. Chapter 2 provides the basic information needed to understand why SSI was chosen as the basis for this architecture and which medical data is being handled. Furthermore, we explained the concept of SSI and presented three architectural approaches for medical records in healthcare that serve as the basis for our architecture. Additionally, we discuss the legal considerations that must be considered to comply with the GDPR, which also impact the design decisions of this chapter. Chapter 3 provides an overview of existing solutions and highlights the unique aspects of our approach. This chapter utilizes the information obtained from the previous chapters to construct the architecture.

This chapter is structured as follows. Section 4.1 explains the requirements and scope of the architecture as it is not possible to consider all relevant aspects of a healthcare system within the scope of the master's thesis. The following section, Section 4.2, defines the architecture, including all relevant components, processes and cryptographic processes. These two sections address **RQ1** and **RQ3**. The final section of this chapter presents an evaluation of the defined architecture. It includes a proof-of-concept (4.3.1), a threat model (4.3.2), an assessment of the defined requirements (4.3.3) and their compatibility with GDPR (4.3.4), as well as an evaluation by experts (4.3.5). This section serves to answer all three research questions.

4.1 Requirement Engineering

This section outlines the guidelines that will be used to define the architecture. In the first section (4.1.1) we define the use cases, we want to consider during the creation process as it is not possible to consider all relevant aspects of a healthcare system within

the scope of the master's thesis. Furthermore, we identified objective criteria for the architecture in Section 4.1.2. These requirements are based on interviews with three experts and additional literature. In addition, we create a threat model in Section 4.3.2 to evaluate the security of the system, for which we explain the process at the end of this section.

4.1.1 Use Cases

To define a concept for the eHealth architecture, we identified the most important use cases as it is not possible to consider all the use cases, scenarios, and processes of a healthcare system within the scope of the master's thesis. These use cases cover both the standard situation of a patient visiting a healthcare provider and scenarios that address the healthcare needs of both the provider and patient in relation to medical records. We did not find sources for a list of necessary scenarios during our research. Therefore, we defined them based on our experience of the processes when patients interact with healthcare providers. These scenarios form the basis for the use cases and are shown in Table 4.1.

ID	Scenario
S1	A patient enters the healthcare provider (e.g. hospital), authenticates itself, and gives the medical staff a record from the last medical examination (e.g. blood test), which is necessary for future treatments.
S2	After the examination, before leaving the healthcare provider the patient receives a written report.
S3	The patient no longer needs some reports and decides to throw them away.
S4	For permanent monitoring, the patient uses an app to enter vital signs every day, which are automatically evaluated by the doctor or other software. Additionally, the patient has the possibility to change some values in their own entries.
S5	A patient changes doctors and no longer wants the first doctor to access the daily vital signs.
S6	A patient's health insurance needs a written report from a doctor to refund the treatment costs to the patient. The patient already has the report and sends a digital copy to its health insurance.
S7	The patient had a laboratory appointment and receives the first results a few days later. The next day, the laboratory sends the remaining results to the patient by extending the existing report.

Table 4.1: Scenarios

The scenarios were challenging to process as requirements for the architecture. Therefore, we divided them into individual user stories. Each user story represents a single activity, or feature of a patient or healthcare provider. We extracted the user stories from the described scenarios, which are presented in Table 4.2.

ID	User story
US1	A patient wants to share their own medical record with the healthcare provider when they are onsite with the healthcare provider.
US2	A patient wants to delete its own medical record.
US3	A patient wants to allow a healthcare provider to access its medical records at any time.
US4	A patient wants to access its own medical records.
US5	A patient wants to edit its own medical records, which are issued by itself.
US6	A patient wants to add an entry to its own medical records.
US7	A patient wants to receive its medical record from the healthcare provider when they are onsite with the healthcare provider.
US8	A patient wants to receive its medical record from the healthcare provider although he is not on site.
US9	A patient wants to withdraw the access permissions for a healthcare provider.
US10	A patient wants to allow a healthcare provider to edit/extend an existing medical record.
US11	A healthcare provider wants to issue and send a medical record to a patient.
US12	A healthcare provider wants to receive a medical record when the patient is onsite.
US13	A healthcare provider wants to receive a medical record when the patient is offsite.
US14	A healthcare provider wants to access medical data from a patient.
US15	A healthcare provider wants to edit/extend an existing medical record.

Table 4.2: User stories

The structure of the following two subsections is similar to that described by Armijo, McDonnell, and Werner [3].

4.1.1.1 Scope / Precondition

The use cases for this architecture cover the most common scenarios from the patient's and healthcare provider's perspective related to EHRs/PHRs. These include the CRUD functionalities for the patient and additional access management functionalities to manage the healthcare providers' access to personal data. A patient must meet the following requirements to engage with a healthcare provider and access their data: (i) possess an internet-capable smartphone (equipped with a camera), (ii) have an active internet connection, (iii) utilize either a service provider or host their own services, and (iv) use a software/application capable of using the appropriate protocols for communication, working with DIDs, and managing and displaying data (EHRs/PHRs). From the patient's perspective, the primary mode of communication within this architecture is through a smartphone, given its portability compared to a laptop. However, it is also possible to use any other internet-enabled device. A camera is not mandatory, but as described in

Section 2.2 it could allow to start an interaction by scanning a QR-code at the healthcare provider. For every patient interaction, an active internet connection is a prerequisite for communicating with the healthcare provider and other components of the architecture. Some approaches include other services, such as DWN or a cloud provider. These services may be hosted by a service provider or self-hosted. Furthermore, a patient requires specialized software to communicate with healthcare providers via appropriate protocols, read and possibly write DIDs, read and view EHRs/PHRs, and manage authorizations. The preconditions for a healthcare provider in such an architecture are like those of the patients, including (i) an internet-capable device, (ii) an active internet connection, (iii) either a service provider or self-hosted services, and (iv) a software/application capable of using the appropriate protocols for communication, working with DIDs, and managing and displaying data (EHRs/PHRs). While patients will use smartphones, the healthcare provider will use other devices such as laptops or tablets as working with them is easier.

4.1.1.2 Stakeholders / Actors

In the architecture, the following stakeholder groups occur:

- **Patients:** access their medical records, share their medical records with healthcare providers, delete their health records, create own data for PHRs, edit own data in PHRs, manage access rights, receive data from healthcare provider (onsite, offsite)
- **Healthcare providers:** receive medical records from patients (onsite, offsite), access medical data from patients at any time, request access rights for patients' data, issue and send health records, edit extend existing records from a patient
- **Additional services:** manage requests for sharing and accessing health data, manage access rights, manage operations on documents and handle documents

4.1.1.3 Specification

The user stories described in Table 4.2 represent the scenarios in a simplified form. To use these user stories as a basis for the architecture, we specify them as use cases to describe individual processes and conditions. This information is necessary for the architecture processes. In the following we specify the use cases using the methodology described by Cockburn [21]. To provide an overview and contextualize the described use cases, we created the case diagram presented in Figure 4.1.

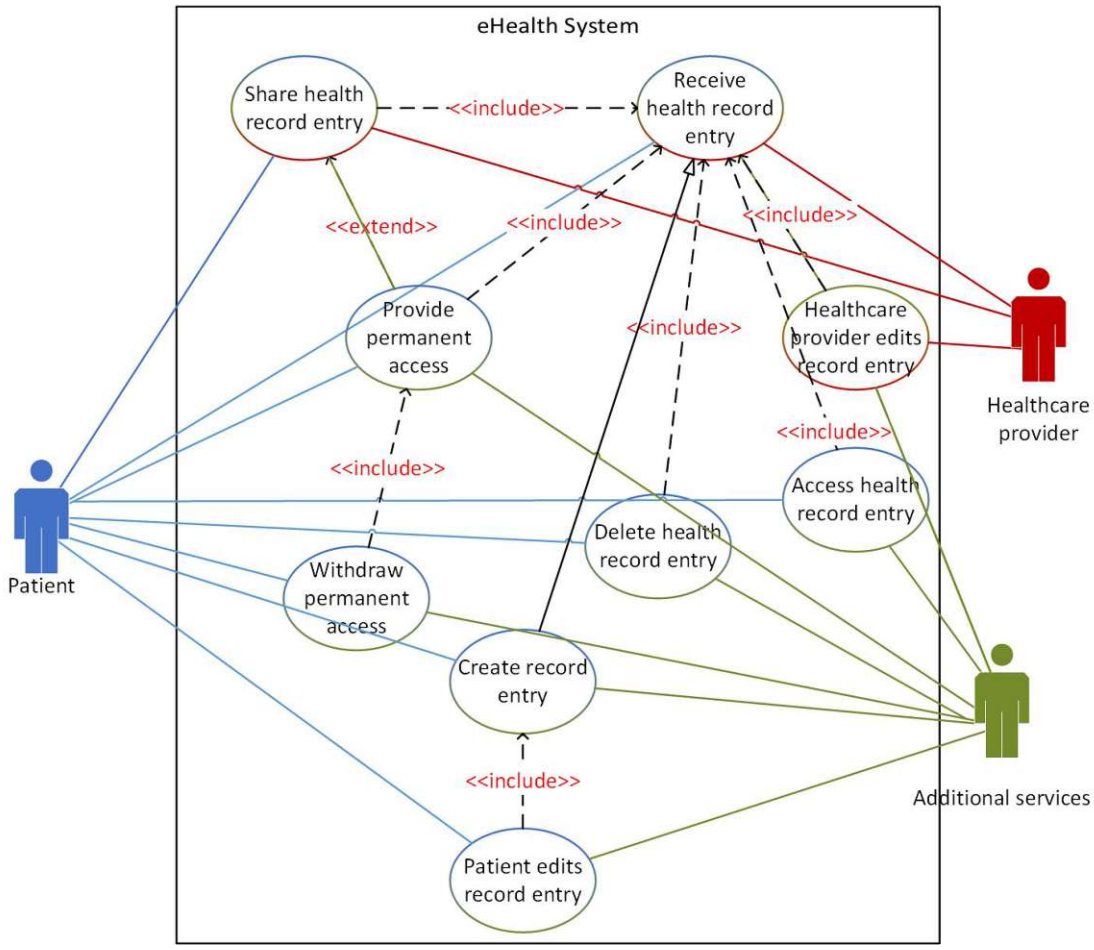

Figure 4.1: Use Case diagram

Name	Share health record entry
ID	UC1
Goal in Context	Patient owns medical record entry and wants to provide access to a healthcare provider
Precondition	Patient owns a valid medical record entry
Success End Condition	Healthcare provider can access medical record entry
Failed End Condition	Healthcare provider cannot access medical record entry
Actors	Patient, healthcare provider, additional services (cloud agent, DWN, etc.)
Trigger:	Patient initiates connection
Description	1. Patient scans QR code 2. Patient creates connection to healthcare provider 3. Healthcare provider requests health record entry 4. Patient share health record entry 5. Healthcare provider checks validity
Extensions	• 2a: Patient authenticates itself with credentials
Sub-variations	• 1: Patient starts remote connection, for example by clicking a link • 3: Patient selects health record entry • 4: Patients' sharing process includes additional services, a connection between healthcare provider and service will be initiated and the service shares the record entry

Table 4.3: Use Case 1

Name	Receive health record entry
ID	UC2
Goal in Context	Healthcare provider issues health record entry to a patient and the patient wants to receive this entry from the provider
Precondition	Healthcare provider has health data from the patient
Success End Condition	Patient has a new valid medical record entry
Failed End Condition	No new record entry/data exists
Actors	Patient, healthcare provider, additional services (cloud agent, DWN, etc.)
Trigger:	Healthcare provider informs patient about new issued health data (verbal or digital)
Description	1. Patient scans QR code 2. Patient creates connection to healthcare provider 3. Healthcare provider shares issued health record entry 4. Patient accepts and validates data
Extensions	• 2a: Patient authenticates itself with credentials
Sub-variations	• 1: Patient starts remote connection, for example by clicking a link • 1: Healthcare provider initiates connection by sending a request to the patient • 3: Patients' receiving process includes additional services, a connection between healthcare provider and service will be initiated and the service receives the data

Table 4.4: Use Case 2

Name	Access health record entry
ID	UC3
Goal in Context	Patient wants to access its own medical record entry
Precondition	Patient owns a valid medical record entry
Success End Condition	Patient can access medical record entry
Failed End Condition	Patient cannot access medical record entry
Actors	Patient, additional services (cloud agent, DWN, etc.)
Trigger:	Patient initiates access procedure
Description	1. Patient selects health data 2. Patient accesses data
Extensions	
Sub-variations	

Table 4.5: Use Case 3

Name	Delete health record entry
ID	UC4
Goal in Context	Patient wants to delete its own medical record entry
Precondition	Patient owns a valid medical record entry
Success End Condition	Health data no longer available
Failed End Condition	Health record entry still exists
Actors	Patient, additional services (cloud agent, DWN, etc.)
Trigger:	Patient initiates deletion procedure
Description	1. Patient selects health record entry 2. Patient deletes health record entry
Extensions	
Sub-variations	

Table 4.6: Use Case 4

Name	Provide permanent access
ID	UC5
Goal in Context	Patient wants to allow the healthcare provider to access its own medical record entry at any time
Precondition	Patient owns a valid medical record entry, patient knows identifier of healthcare provider
Success End Condition	Access management rule exists, healthcare provider can access medical record entry at any time
Failed End Condition	No access management rule exists, healthcare provider cannot access medical record entry
Actors	Patient, additional services (cloud agent, DWN, etc.)
Trigger:	Patient initiates access management rule creation procedure
Description	1. Patient selects health data 2. Patient enters identifier of healthcare provider 3. Patient sends request for creation to additional service
Extensions	• 1a: Patient sets an additional setting which allows the healthcare provider to edit/extend the health record entry
Sub-variations	

Table 4.7: Use Case 5

Name	Withdraw permanent access
ID	UC6
Goal in Context	Patient does not want to allow the healthcare provider to access its health record entry any more
Precondition	Patient owns a valid medical record entry, patient has active access management rule for healthcare provider
Success End Condition	Access management rule no longer exists, healthcare provider cannot access medical record entry
Failed End Condition	Access management rule still exists, healthcare provider can access medical record entry
Actors	Patient, additional services (cloud agent, DWN, etc.)
Trigger:	Patient initiates access management rule withdrawal procedure
Description	1. Patient selects access management rule 2. Patient sends request for withdrawal to additional service
Extensions	
Sub-variations	

Table 4.8: Use Case 6

Name	Create record entry
ID	UC7
Goal in Context	Patient wants to add an entry to its own medical records (PHR)
Precondition	
Success End Condition	New record entry exists
Failed End Condition	No new record entry exists
Actors	Patient, additional services (cloud agent, DWN, etc.)
Trigger:	Patient initiates creation procedure
Description	1. Patient selects local data 2. Patient adds necessary information 3. Patient creates new entry
Extensions	
Sub-variations	

Table 4.9: Use Case 7

Name	Patient edits record entry
ID	UC8
Goal in Context	Patient wants to edit its own medical record entry(PHR), Patient cannot edit valid medical record entry issued by healthcare provider
Precondition	Patient owns a valid medical record entry (PHR)
Success End Condition	Record entry changed
Failed End Condition	No change in record entry
Actors	Patient, additional services (cloud agent, DWN, etc.)
Trigger:	Patient initiates change procedure
Description	1. Patient selects health data 2. Patient enters information for change 3. Patient edits entry
Extensions	
Sub-variations	

Table 4.10: Use Case 8

Name	Healthcare provider edits record entry
ID	UC9
Goal in Context	Healthcare provider wants to edit/extend an existing medical record entry from a patient which was issued by the healthcare provider
Precondition	Patient owns existing health record entry, data was issued by healthcare provider, healthcare provider has access
Success End Condition	Medical record entry changed
Failed End Condition	No change in record entry
Actors	Patient, healthcare provider, additional services (cloud agent, DWN, etc.)
Trigger:	Healthcare provider initiates change procedure
Description	1. Healthcare provider accesses data 2. Healthcare provider edits data or enters new data 3. Healthcare provider submits changes
Extensions	
Sub-variations	

Table 4.11: Use Case 9

4.1.2 Requirements

The creation of a functional, secure, and practical design must be based on objective criteria. Given the diverse needs and limited research in this specific area, we conducted three expert interviews using a pre-established guideline. To access the interviews we use the methodical approach according to Meuser and Nagel [15]. Based on the interviews, we identified the most important requirements as a basis for the evaluation of our architecture. Furthermore, the requirements help us to make appropriate design decisions to answer research question three (**RQ3**). The last part of this section serves to define the methodology for creating the threat model. We create the actual threat model in Section 4 to review the security aspects of this architecture.

4.1.2.1 Interviews

The interview process comprises three phases: (i) developing the interview guidelines, (ii) conducting the interview, and (iii) evaluating the interview. To develop the interview guidelines, we based our approach on the requirements engineering process in software development. We carefully considered which information and requirements were relevant

to our architecture and used this to design the categories and questions for the guide. To select suitable interviewees, we enlisted the assistance of our peers. We searched in the university environment for individuals with relevant experience and knowledge in the field of medical records in healthcare. For our selection, we ensured that the experts had diverse backgrounds and experiences.

According to Edgar and Manz [35] the method chosen to select interview partners corresponds to a non-probabilistic sampling method, known as purposive sampling, which is used when population distributions are unknown. According to Baxter et al. [8], this method does not involve a truly representative sample of the population, but rather targets individuals with a specific characteristic of interest. In our case, our priority was to conduct interviews with experts in the field of digital healthcare. The purpose of our sampling method is to gather information on what is important to experts regarding medical records in healthcare. We decided to use this sampling method because we do not anticipate receiving any useful information on requirements for this topic from a target group that is not well-versed in the subject. However, the selection process introduced a corresponding bias [35]. The term for the bias that occurred in our case is pre-screening bias [99], which means that only a specific target group of participants were pre-selected for the study. The potential bias does not present any disadvantage for our goal of conducting interviews and establishing requirements. It is unlikely that we would have obtained any meaningful answers from a non-pre-selected sample.

To analyze the interview, we used the methodological approach of Meuser and Nagel [15]. We selected this method as it is more suitable for expert interviews than other qualitative analysis methods like Mayring and Fenzl [76].

Creation of the Interview Guideline and Interview Execution

The guideline is divided into four categories: (i) functional technical requirements, (ii) socio-demographic data, (iii) non-functional technical requirements, and (iv) security aspects and requirements. Functional requirements (i) should provide additional input to the use cases and depict what both users and healthcare providers value. The socio-demographic questions should provide information on the geographic and technical aspects of the infrastructure. Together with the non-functional requirements, these questions capture the performance criteria. The security aspects and requirements should form the basis for processes and safety considerations. We devised category-specific questions to identify the most important requirements to answer research question three (**RQ3**). In total, the guideline comprises twenty questions relevant to the topic of digital healthcare systems.

During this thesis, we conducted two rounds of expert interviews. The first round was to collect the requirements, and the second round was to evaluate the finished architecture. Each expert has unique areas of expertise. The first round of interviews was conducted with three of the experts. All of them have been involved in notable eHealth projects, including the German digital health record implementation. An overview of the experts is shown in Table 4.12. The first expert (EX1) possesses extensive experience in the planning

and execution of eHealth systems, having worked as Senior Systems Architect, Senior Software Architect, Senior Developer, and Development Lead for 4 years. He therefore has general expertise in all necessary areas related to eHealth systems. The second expert (EX2) has worked for four years in planning and developing eHealth systems for digital applications in the German healthcare system. He works as a software architect and software developer and has a total experience of 20 years in the eHealth sector. His expertise covers all general aspects of eHealth architectures and health data. The third expert (EX3) is a specialist in the field of security and has 18 years of experience in the eHealth sector. He designs, analyzes, and improves security architectures related to healthcare systems. His knowledge and skills include security and privacy-related domains. In our professional environment, we have identified the fourth expert (EX4) who has been working on healthcare projects for seven years. EX4 serves as a cybersecurity consultant and conducts risk analyses and assessments for healthcare applications, among other responsibilities. Therefore, EX4 possesses expertise in the security sector. We identified the fifth expert (EX5) with the assistance of peers in the university environment. EX5 has six years of experience in the eHealth sector and has previously worked in other areas of critical infrastructures. EX5 is responsible for the design and implementation of various architectures, making him an expert in the field of software architectures with a focus on eHealth.

Expert	**Experience**	**Expertise**	**Gender**	**Experience eHealth sector**	**Participation in interview**
EX1	Planning and execution of eHealth systems	All necessary areas related to eHealth systems	Male	4 years	1 & 2
EX2	Planning and development of eHealth systems	eHealth architectures and health data	Male	20 years	1 & 2
EX3	Designs, analyzes and improves security architectures	Security and related topics	Male	18 years	1 & 2
EX4	Creation, analyzing and assessment of security architectures	Security in eHealth and critical infrastructure	Male	7 years	2
EX5	Planning and implementation of architectures in eHealth context	Software architectures with eHealth focus	Male	6 years	2

Table 4.12: Overview of experts for the evaluation interviews

Each interview lasted one hour and was conducted as a semi-structured interview based on the guidelines. The interviews with EX1 and EX2 were conducted remotely and the interview with EX3 was held on-site. The interviews were audio recorded and the audio recording was deleted after the transcription.

Evaluation of the Interview
To evaluate the interviews we utilize the methodology of Meuser and Nagel [15], which includes six phases: (i) transcription, (ii) paraphrase, (iii) headlines, (iv) thematic comparison, (v) sociological conceptualization, and (vi) theoretical generalization. Our implementation of the individual phases is described below.

Transcription The interviews were recorded and transcribed, following the methodology's instructions to transcribe the interview as accurately and completely as possible. However, some parts may be omitted based on the discourse's direction and relevance to operational or contextual knowledge. We have accurately transcribed the necessary information while omitting irrelevant details.

Paraphrase In this phase, we paraphrased the content to follow the conversation chronology and capture the overall expressions of the experts. The aim was to reduce complexity and maintain the presentation of expert knowledge. We only made changes to content that was not specific and too complex and reduced the other interview content to the essentials. The resulting text contains only the answers of the experts. The questions from the interviewer were removed and the context was inserted into the answer if necessary.

Headlines This methodology step involves creating headings for each section of the interview and grouping similar passages together. This process was performed separately for each interview, as outlined in the methodology. Instead of using headings as suggested in the methodology, we used labels/tags and marked the individual passages in the text, as it is easy to continue working with this data in the next step. These labels were chosen so that they fit the text, but also regarding the superordinate categories for our requirements. We consulted the paper by Calvillo-Arbizu, et al. [18] for guidance. The reasons for choosing this paper will be elaborated in an upcoming phase. We utilized the following 23 tags: (i) privacy, (ii) usability, (iii) performance, (iv) data management, (v) geographic structure, (vi) costs, (vii) data link, (viii) non-functional requirement, (ix) scalability, (x) storage duration, (xi) regulatory measures, (xii) integrity, (xiii) accessibility, (xiv) authorization, (xv) security, (xvi) data synchronization, (xvii) identity management, (xviii) user needs, (xix) politics, (xx) data size, (xxi) access management, (xxii) authentication, (xxiii) additional considerations.

Thematic Comparison According to Meuser and Nagel [15], this level of the interview analysis requires thematically comparable text passages from different interviews to be brought together. These texts should be merged, and uniform headings created for

distinct sections. Additionally, differences should be noted. We merged segments with similar content, based on tags, and established a new heading for the corresponding topic. We worked out the similarities and differences between the experts and reduced the number of headlines to 13.

Sociological Conceptualization The previous text is simplified and summarized to make it more understandable. Our goal is to simplify further and provide clear explanations of the ideas presented.

Theoretical Generalization The final step of evaluating the interviews can be omitted based on the role of the expert interview in the research design. We have decided against carrying out this step since it would not contribute to gaining further knowledge. The data gathered from the evaluation was primarily used to generate or directly incorporate the appropriate requirements.

4.1.3 Creation of the Requirements

In this section, we use the results of the interviews to establish the requirements that will form the basis of the healthcare architecture. We have created the following categories for the requirements: (i) data lifecycle, (ii) trust, security, and privacy, and (iii) human-related issues. We found this categorization of requirements in a paper published by Calvillo-Arbizu, et al. [18] on the topic of Internet of things in healthcare. We selected this categorization because it aligns with the overall theme of our architecture and the results of the interviews. The data lifecycle category includes the non-functional requirements related to health data. This involves the criteria of scalability, flexibility, low latency, persistence, and other performance-related criteria. The second category comprises all the security related requirements, such as identity management, privacy, and access control. The last category includes the needs and capabilities of users. Users in this context are patients and healthcare providers. Specific requirements are accessibility, usability and regulatory measures.

The requirements are based on expert statements and designed for use and assessment in the evaluation chapter. We follow the SMART method [56], which is typically used for setting general goals, to guide our approach. We try to define the requirements as specific, measurable, attractive, and realistic as possible. The majority of the expert statements do not include any applicable requirements, therefore we analyze them and create requirements, which align with the overarching digital healthcare architecture goal. We will also add requirements based on literature that were not mentioned by the experts. EX2 provided us with a specification document [45], which contains performance criteria for the electronic health card and telematik infrastructure in Germany. He said that these are approximations that are not entirely realistic but can still be used as a reference. In general, the requirement applies to the entire system, unless it is explicitly specified.

Data Lifecycle

All three experts stated that the performance of a system should consider the criteria of access time, scalability, and availability. Furthermore, they believed specific requirements depend on several factors, including use cases. Regarding access time, EX1 stated that a duration of several seconds, e.g. 10 to 15, is unacceptable for a healthcare provider's process. The process's duration should be comparable to that of using paper documents, taking only a few seconds. For the creation of the access time requirements, we distinguish between the use cases and refer to the document cited in the preceding paragraph to determine the maximum duration of access. The precondition for the requirements is a stable internet connection with a minimum speed of 1 Mbps is required. The duration of the operations depends on the size of the recorded entry. To establish a reference point, we take the smallest possible object available for a given record. For instance, this could be a brief text document. The specification document measures the average reading process time at 2 seconds for every 100 KB of patient record data. Since there are few points of reference for performance criteria, we have decided to adopt the approximations of the Gemmatik specification [45]. We have set an average duration of 2 seconds for read requests, starting from the moment they are sent. This time frame is comparable to that mentioned in the specification. Deletion operations may require up to 1.1 seconds, as noted in the specification. Based on our experience, the upload speed is lower than the download speed. Additionally, the write operation is more time-consuming. Therefore, we have set the duration higher than for the other two operations. The calculation in the specification document is based on a 1024Kbit/s line, which yields an estimated 8.3 seconds for 100 KB. We have set our limit at the suggested 8.3 seconds, which should suffice for healthcare providers sending small data records to patients. All access management operations are similar to the delete operation in that they involve minimal data exchange. However, more processes take place in the background than when deleting data, which is why the average access time is set to 2 seconds. In practice, it is important to adapt the limits and speeds to the expected data volumes. Otherwise, downloading or uploading a file with several GB in size could take several hours.

EX1 and EX3 identified performance spikes as a scalability concern, due to different usage times of providers. However, no specific requirements were defined. Consequently, we established the three scalability requirements for record operations: read, write, and delete. Since access management is not as frequent as other operations, scalability requirements do not need to be created for it. Our system is intended to function decentral, not as one large system, so scalability in terms of performance peaks is already ensured by design. However, we would like to set requirements per person. Since we have no indication of which performance peaks need to be considered, we can only provide a rough estimate. It should be possible to retrieve one record entry per second, which would enable a user to view and download all records individually. Moreover, it should support uploading up to 5 records simultaneously, allowing service providers to store multiple files in a patient's record. Since deleting documents typically receives lower priority, the rate of deletion requests to be handled per second may be lower. Therefore, we define that one deletes request must be handled per second. However, this does not

imply that one deletion must occur every second.

Two of the interviewed experts (EX2 and EX3) believe that the availability of digital health systems must be guaranteed. To meet the requirements for a SMART goal, a 98% availability rate is defined. Since a system cannot achieve 100% availability and updates also must be carried out, we reduce the availability by two percent. Additionally, backup measures are crucial for ensuring availability, specifically for both data and access keys. Therefore, it is defined that a system backup will be performed daily at a consistent time. To ensure the protection of data access keys and their recovery in case of loss, an appropriate solution needs to be identified. We therefore come to the requirements, shown in Table 4.13.

ID	Scenario	Origin
R.1.1	The average time it takes to read a record entry with the least amount of data possible is 2 seconds.	EX1 & Telematik infrastructure specification document [45]
R.1.2	The average time it takes to delete a record entry with the least amount of data possible is 1.1 seconds.	EX1 & Telematik infrastructure specification document [45]
R.1.3	The average time it takes to write a record entry with the least amount of data possible is 8.3 seconds.	EX1 & Telematik infrastructure specification document [45]
R.1.4	Access management operations (i.e., allow access and delete access) have a average duration of 2 seconds.	EX1 & Telematik infrastructure specification document [45]
R.1.5	The architecture allows to query for one record entry per second.	EX1 & EX3
R.1.6	The architecture allows to handle five data uploads simultaneously.	EX1 & EX3
R.1.7	The architecture allows one deletion request per second.	EX1 & EX3
R.1.8	The digital healthcare system has an availability of 98% per day.	EX2 & EX3
R.1.9	A complete backup of the system's data is performed every day at 11:55 p.m.	EX2
R.1.10	A mechanism exists for the recovery of access keys in case of loss.	EX2

Table 4.13: Data Lifecycle requirements

Trust, Security, and Privacy

All experts identified the topics of security and privacy as one of the most important aspects in a digital healthcare architecture. According to EX1 and EX3 the most important security requirements are secure communication, secure storage of data, and

secure access. In addition to availability, confidentiality, authenticity, non-repudiation, and integrity are named as essential requirements for security. When we asked the experts about when a healthcare system is privacy-preserving, they uniformly indicated that patients must have sovereignty over their own data. Specifically, no other third party may access the data without the patient's consent. Therefore, we defined that access to the data is only possible with the patient's cryptographic keys. In addition, we established a requirement that only the patient can allow any other entity to access their data. Additionally, EX1 points out that the system must guarantee that patient activity and usage cannot be tracked using data retrieved from the system (traceability). This requirement also impacts the system's log records, which must utilize neutral identifiers to prevent entities from inferring which data was accessed by whom.

All experts mention authorization and access management for ensuring confidentiality and integrity. EX1 points out that service providers should only have access to data that is specifically necessary. Appropriate authorization concepts and access mechanisms must ensure this. Healthcare providers must use different types of data differently. Accordingly, there must be access mechanisms that allow fine-grained access management. Our requirement specifies that a patient can generate an access rule for a specific entity to access a solitary health record. Additionally, the patient is also able to delete this access rule and revoke the entity's access since the patient has data sovereignty.

It is necessary to have a mechanism that enables identifying the source of data alteration and ensures integrity of such a system. This is crucial for healthcare providers who must trust the information they obtain. The system's integrity must guarantee that data cannot be altered without detection. An example was proposed by EX1 regarding the prevention of patients from adding medication to their prescription themselves. This led to the creation of a suitable requirement. To maintain secure communication, it is essential to avoid transmitting data in plain text. All three experts recommend that the data be transmitted in encrypted form. As a result, one requirement has been established to ensure this is the case.

In the interviews, all three experts believed implementing multiple layers of security would enhance the overall security of a system. They also emphasized the importance of avoiding a single point of failure, which may lead to unauthorized access to other systems or data via the architectural component. Accordingly, we determine that at least three security mechanisms (authentication, authorization, encryption, etc.) are required to access data. EX3 stated that digital healthcare system operators have an incentive to minimize their data access, as this can limit their liability and enhance patient confidentiality. It is advantageous for operators to solely receive and store the encrypted binary blob, without accessing its contents. We thus conclude that an operator of said system must possess the corresponding keys to obtain any information. We therefore come to the requirements, shown in Table 4.14.

ID	Scenario	Origin
R.2.1	The patient's own data in the system can only be accessed with the patient's matching cryptographic keys in the standard configuration of the healthcare system.	EX1 & EX2 & EX3
R.2.2	Another entity can access the patient's data only if the patient sets up an access rule. In this case, the data can be retrieved using the entity's cryptographic keys.	EX1 & EX2 & EX3
R.2.3	Patient data can only be written to the system in encrypted form and read in encrypted form using a secure encryption mechanism for today's standards.	EX1 & EX2 & EX3
R.2.4	The system's standard configuration of the healthcare system restricts any entity other than the data owner (patient) from querying information of any kind, including meta data.	EX1 & EX2 & EX3
R.2.5	Log entries feature neutral IDs that prevent drawing conclusions about individual data entries or users.	EX1
R.2.6	Patients can create an access rules for their data that enable entities to access a single health record entry.	EX1
R.2.7	Patients can delete an access rule for their data and revoke the access for an entity.	EX1
R.2.8	An entity can determine the correctness and the issuer of a health record entry.	EX1 & EX3
R.2.9	The system architecture must incorporate at least three security mechanisms (such as authentication, authorization, and encryption) for data access.	EX1 & EX2 & EX3
R.2.10	The operator of a digital health system is only able to view an encrypted binary blob from which no information can be extracted.	EX3

Table 4.14: Trust, Security, and Privacy requirements

Human-Related Issues

The category of human-related includes the needs and capabilities of users, whereby users in this context are patients and healthcare providers. According to all experts the biggest demands from the average user are usability and functionality. While the system should have the capability to manage individual access rights, patients may not have the desire or ability to personally handle every single authorization. According to EX1 and EX2, there are technical terms that patients may not be familiar with requiring additional knowledge. In general, the topic of usability is not considered in our architecture since it is not essential for functionality, and thus we will not establish any usability requirements.

The experts EX1 and EX3 suggest that the health record of a patient should be designed to

endure a lifetime. This requires addressing regulatory requirements and other procedures, like re-encrypting access keys. There are laws that describe how long documents must be stored. As per Austria's laws, we define that the system must have the capacity to store data from the last seven years. Regarding re-encryption, we require that users be able to create a new key pair in our system to access their data. Another regulatory consideration is ensuring the interoperability of these systems across national borders. According to EX3, EU-wide efforts are being made to implement an electronic health record system throughout the EU. As a requirement, the system must be able to function in multiple countries while adhering to local laws. In the healthcare system, certain datasets must be viewed to be fully understood, such as vaccination records or medication lists. Insufficient visibility of such objects by healthcare providers can result in complications. It is therefore a requirement to consider such data sets in our architecture. Additionally, EX3 recommended that only smartphones be utilized for data management, eliminating the need for special devices. We therefore come to the requirements, shown in Table 4.15.

ID	Scenario	Origin
R.3.1	The architecture allows patients to store their medical record entries securely for at least seven years.	EX1 & EX3
R.3.2	The architecture allows the patient to specify a different key pair to access the health data.	EX3
R.3.3	The architecture allows the healthcare provider to create an object, which is only visible in its entire form.	EX1
R.3.4	The architecture allows the patient to use a smartphone to manage the health data.	EX3
R.3.5	A healthcare provider from another country can access a patient's data if the patient have configured the access accordingly. Special services or approvals from their home country are not required for the healthcare provider to do so.	EX2 & EX3

Table 4.15: Human-Related Issues requirements

4.1.3.1 Threat Model

To analyze the healthcare architecture from the security perspective, we will use the methodology of threat modeling. In this section we define which methodology we use for the threat model and then apply it in Section 4.3.2. Threat modeling is used to identify, communicate, and understand threats and mitigations, whereby the threat model represents all the information that affects the security of an application in a structured way [33]. For the evaluation we utilize the four question framework to organize threat modeling, as described by OWASP [33]. This approach allows us to create a threat model in a structured way and is suitable for our master's thesis goal in terms of both content and scope. The four questions that we deal with in the four question framework approach are: (i) What are we working on? (ii) What can go wrong? (iii) What are we going to

do about it? (iv) Did we do a good job? [33] All four questions represent a phase of the threat modeling process.

The first phase is to assess the scope and identify what we are working on. This includes the decomposition of the application. Specifically, we try to identify entry points for attackers, identify assets which are of interest to the attacker, and identify trust levels representing access rights. In this phase we will produce a data flow diagram to understand the application. [34]

The second phase determines and ranks threats. We use the STRIDE threat categorization methodology for this purpose as proposed by Kohnfelder and Garg [70]. In the STRIDE approach, threats are categorized according to the following attacker goals: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. The analyzation process involves the analysis of each aspect of the application's functionality, architecture, and design. The threats can then be ranked by risk, where risk models use different factors to calculate the risk. As described by OWASP, one possibility is to use the DREAD model. However, this is not widely used due to its subjectivity. We have therefore opted for the Common Vulnerability Scoring System v3.1 (CVSS) to evaluate the threats found. [34]

In the third phase we determine countermeasures and mitigations by using the STRIDE categorization from the last step [34]. The last phase is an assessment of the previous three steps and evaluate if the outcome is suitable for the architecture [33].

4.2 Architecture

In this section we describe all the relevant steps and design decisions for the process of creating the eHealth architecture. The aim of this section is to address the research questions one (**RQ1**) and three (**RQ3**) and create a privacy-preserving eHealth architecture. For the creation process, we use the information and the three models developed in Section 2.2. In addition, the outcomes from Section 2.3 are taken into account.

The first section (4.2.1) contains the actual creation process and all the necessary considerations and specifications required to create a digital health system. We explain and argue all decisions based on the use cases, requirements, and technologies. As cryptography is a necessary component for the security and privacy of the architecture we explain which cryptographic concept is used and where in the architecture in the next section (4.2.2). The following section (4.2.3) discusses the system workflows and shows the interactions between all involved entities.

4.2.1 System Design

To define the architecture of the eHealth system and its components, we begin by establishing the general structure. In Section 2.2, we created three possible approaches that describe how SSI can be used in the context of healthcare. All three approaches can

be used for different purposes. However, they are not all equally useful for the use case of a healthcare system. In the following, we use the designations AP1(2.2.2.1), AP2(2.2.2.2) and AP3(2.2.2.3) to refer to the respective approaches.

The first approach (AP1), using medical records directly as verifiable credentials, will not be sufficient for two main reasons. The biggest drawback of this solution is the lack of functionality to store and manage unstructured data. Therefore, another approach would have to be used anyway. This method is intended to be used for sharing credentials with small data size. As the approach in its original form does not require any additional services, all data would have to be stored locally on an end device. A smartphone will not be able to handle the amount of data in wallet apps necessary for an potentially lifelong eHealth system as defined in requirement **R.3.1**. These devices and applications are not designed to store gigabytes of health data for several years.

Since AP1 is not a solution for creating an architecture, we have to compare the remaining two approaches. In principle, both approaches have some overlaps. In the second method, a cloud agent is used for communication to make the data available to a healthcare provider. In the third approach, this cloud agent is the Decentralized Web Node, which pursues the same goal. The difference between the approaches is that the storage of the data in AP2 is based on a distributed file storage, such as IPFS, and in AP3 directly on the components. This results in different processes for the various use cases and corresponding advantages and disadvantages. The basic functionality and use cases can be covered by both approaches. Our decision is therefore based on the requirements of section 4.1.2 and the individual disadvantages of the approaches.

Regarding **R.1.9**, the need to carry out backups, AP2 has the advantage compared to AP3 by design. IPFS can be used to store record entries on multiple instances. This can be ensured by additional protocols. In contrast, AP3 needs to use the Sync interface of DWNs, and it must be ensured that several synchronized instances exist. However, AP2 has some disadvantages compared to AP3, especially regarding the requirements **R.2.4** and **R.2.9**. By design, IPFS can be viewed by anyone unless it is in a privately managed IPFS network. Anyone can therefore access the encrypted information. This contradicts requirements **R.2.4** and **R.2.9**. Regarding **R.2.4**, at least the information on how large the entry is can be retrieved and with **R.2.9** there is only one security mechanism (encryption) that prevents an attacker from accessing the data. Although it is possible to set up IPFS as a private network and protect it with access policies, this raises the question of who hosts the files and is responsible for access. In contrast, AP3 does not publish any information to the public by default to ensure **R.2.4** and allows to establish authentication and authorization mechanisms to increase the amount of security mechanism.

As AP2 does not fully cover the security requirements, we use the approach AP3 for the architecture. In the following, we describe all the necessary components and their properties to answer research question **RQ1** and **RQ3**. The following sections build on each other structurally so that all the necessary information for a section has already been answered in the previous one.

4.2.1.1 Verifiable Data Registry (VDR)

The verifiable data registry serves as the foundation of the architecture. Section 2.2 outlines two possibilities for the VDR - a DLT-based blockchain approach and a Key Event Log Infrastructure (KERI) approach. Both methods effectively register DIDs and create a decentralized public key infrastructure. However, the blockchain approach has some drawbacks when compared to KERI. In a blockchain-based approach, an algorithmic-root of trust is utilized, requiring involvement from other parties to host the infrastructure [90]. Whereas, in a KERI based approach we have a self-certifying root of trust, avoiding the need for additional parties [90]. By using the KERI approach, we can circumvent issues associated with identifying the nodes of the blockchain and selecting the most efficient consensus mechanism for our scenario. Additionally, in a blockchain-based approach, the decentralized identifiers (DIDs) are bound to a specific blockchain and cannot be transferred in case of ledger issues [90]. On the other hand, with a KERI-based base layer, the user can generate their own DIDs, which allows for complete self-sovereignty. In the interview with EX2, he advised us that self-signed certificates are preferable for our use case compared to a blockchain based approach as he sees no advantage in using a blockchain. As mentioned in Section 2.3 the append-only characteristics of blockchains conflicts with the GDPR, particularly with the "right to be forgotten" outlined Article 17 of the GDPR. In contrast, KERI provides a non-intertwined identifier trust base, that enables a user to delete the identifier's data [57]. The biggest disadvantage of KERI compared to a blockchain based approach is that the DID method of KERI does not support the "service" parameter, according to the current status in the unofficial draft to the did:keri method [118]. This "service" parameter is essential for the the main component of our architecture, the DWN. Since we prefer the functionality of the DWN in the architecture over perfect decentralized key management and the blockchain also covers all the necessary functionalities, it is used as the VDR in the architecture.

A blockchain has several characteristic properties. The first consideration is which type of blockchain is suitable for the application. Belchior, et al. [9] categorizes blockchains into two groups, public (permissionless) blockchains and private (permissioned) blockchains. Public blockchains, such as Bitcoin and Ethereum, do not require authentication for participants to access the ledger. In contrast, private or permissioned blockchains involve authenticated users. Examples of private blockchains include Hyperledger Fabric, Sovrin, and R3 Corda. Often, the choice of blockchain type is accompanied by the selection of a consensus mechanism. Guo and Yu [49] described multiple different consensus algorithms, including "Proof of Work (PoW)", "Proof of Stake (PoS)", "Practical Byzantine Fault Tolerance (PBFT)", and "Proof of Authority (PoA)". In the context of our SSI-based architecture, a public blockchain is most appropriate, as each patient could create their own DIDs. Another important aspect is not to build on an existing third party blockchain, such as Ethereum, because the entire electronic health system would be dependent on another party. The used blockchain should support standards and cryptographic primitives, which are necessary for a SSI infrastructure, such as schema and credential definitions. Blockchain technology is relatively new. Defining a blockchain and its

properties, as well as thoroughly researching the associated advantages and disadvantages in various application areas, is still an active area of research. Therefore, it is not possible to provide a precise specification within the scope of this work.

4.2.1.2 Decentralized Identifier

The VDR stores DID documents that entities can query with the corresponding DID, ensuring effective communication within our architecture. The DID documents require a specific structure to accomplish this goal. DID documents support three different data representations: (i) did+cbor, (ii) did+ld, and (iii) did+ld+json. A data representation is created by serializing the data model. Since the method did+cbor just shows binary data, we choose the representation of did+ld+json to represent the DID document data. The benefits of did+ld+json over did+json is its semantic interoperability and comprehensibility for both machines and humans. [114]

The DID document must include following properties: (i) @context, (ii) id, (iii) verificationMethod, (iv) authentication, (v) assertionMethod, (vi) keyAgreement, (vii) capabilityInvocation, and (viii) service.

The "@context" property establishes the specific terms utilized in our JSON-LD environment. We utilize "https://www.w3.org/ns/did/v1" as a reference to identify data as a DID document. The second URL is referring to the cryptographic suites, which are used in the verification method object. We have selected the "JSON Web Signature 2020" suite for our architecture, as it provides all necessary cryptographic methods. The "id" field indicates the DID assigned to the specific DID subject. The "verificationMethod" includes one or multiple cryptographic public keys that can be utilized, for example, in authentication procedures. The DID subject possesses the corresponding private key and other entities can interact with it using the public key specified in the "verificationMethode". [114]

In a DID document, the DID subject defines DID relationships. These relationships specify the correlation between the subject and a verification method. Each verification relationship serves a different purpose and allows for distinct key definitions. To cover the functionality of the use cases, we need the verification relationships "authentication", "assertionMethod", and "capabilityInvocation" in the DID documents. The DID specification allows for referencing from the verification relationship to the keys defined in the "verificationMethod".

The "authentication" property specifies the expected authentication method for a DID subject to access a DWN in our architecture. The "assertionMethod" is essential for issuers to define how a VC is issued. The keyAgreement property is utilized to establish mutual agreement between two parties on a symmetric key. In the architecture, this is required for the encryption and decryption of data and to transfer encrypted medical records from the issuer to the DWN. The "capabilityInvocation" allows a DID subject to invoke a cryptographic capability, which will be required in the authorization process of the DWN. We define unique keys for different verification relationships as they will be

used for authentication and authorization in the architecture. As defined in requirement **R.2.9** we want at least three security mechanisms. If one key gets compromised, an attacker can override two security mechanisms at once. We explain, why we chose which keys for which properties later in Section 4.2.2. [114]

The "service" paramater is necessary to specifically target a person's DWN through their DID. This property must include the "serviceEndpoint" parameter, which refers to the DID subject's DWN. We do not define the DID method as it is not necessary for the overall architecture principle. We assume that it is defined and provides the required functionality for creating, updating, resolving, and deactivating a DID.

4.2.1.3 System Interactions

The core components necessary for a functional architecture are (i) the DWN, (ii) a user/edge agent, (iii) a software for issuing and verifying credentials, (iv) a blockchain, (v) a DID resolver, and (vi) the healthcare authority. The overview of the architecture is shown on Picture 4.2, which is similar to the proposed architecture in the DWN specification of W3C [29].

According to the W3C recommendation on DIDs [114], a DID resolver is a system component that performs the DID resolution process by taking a DID as input and producing a conforming DID document as output. The resolution process represents the read operation of a DID method and is based on the used VDR for storing DIDs. The DID method specifies how the resolution is performed. As described by Preukschat and Reed [93] there are multiple approaches of DID resolvers. They can be implemented as a native library in an application, or a third party can provide a DID resolver as a DID service. Since we use a blockchain as a VDR and not every user wants to be a node for this blockchain and store all transactions on their device, we choose the approach with a remote DID resolver. To receive a DID document, the edge agents send a HTTP/HTTPS request to the DID resolver, which responds with the corresponding DID document. This method also allows each user to implement the appropriate DID method locally and is then no longer dependent on the remote DID resolver. In principle, any node in the blockchain with an appropriate implementation of the DID method can also be a DID resolver. Therefore, there can also be several DID resolvers in the architecture. The health authority serves as a central point of trust in the healthcare system, enabling healthcare providers to establish a sense of identification with one another.

The DWN can be either self-hosted or provided by a service provider. Every entity needs an edge agent to resolve DID documents from the DID resolver, and for communicating with the DWNs. The edge agent is responsible for secure messaging and communication, interoperability with other agents and networks, user-controlled data sharing, data storage security, identity authentication and verification, and identity recovery and revocation [1]. In this architecture, the edge agent's primary purpose is not to manage credentials or store data. That responsibility belongs to the DWN. Furthermore, the DWN handles access permissions for the data stored on it and sends messages to the edge agent. The

serviceEndpoint field within the DID document includes an HTTP/HTTPS URL to the entity's DWN, serving as the communication entry point between the edge agent and the DWN. All transmission between the edge agent and DWN relies on message objects. When a patient or healthcare provider sends a message to the DWN, there are two options depending on the use case. The DWN can either forward the message to the edge client of the other entity, which will respond with a response object, or the DWN can process the request itself and respond with a response object. We use the HTTP/HTTPS protocol for communication between the edge agent and the DWN for reasons of simplicity. The keys required for HTTPS are obtained by the sender from the DID document. The following image provides an overview of the described architecture, like the DIF Decentralized Web Node specification [29].

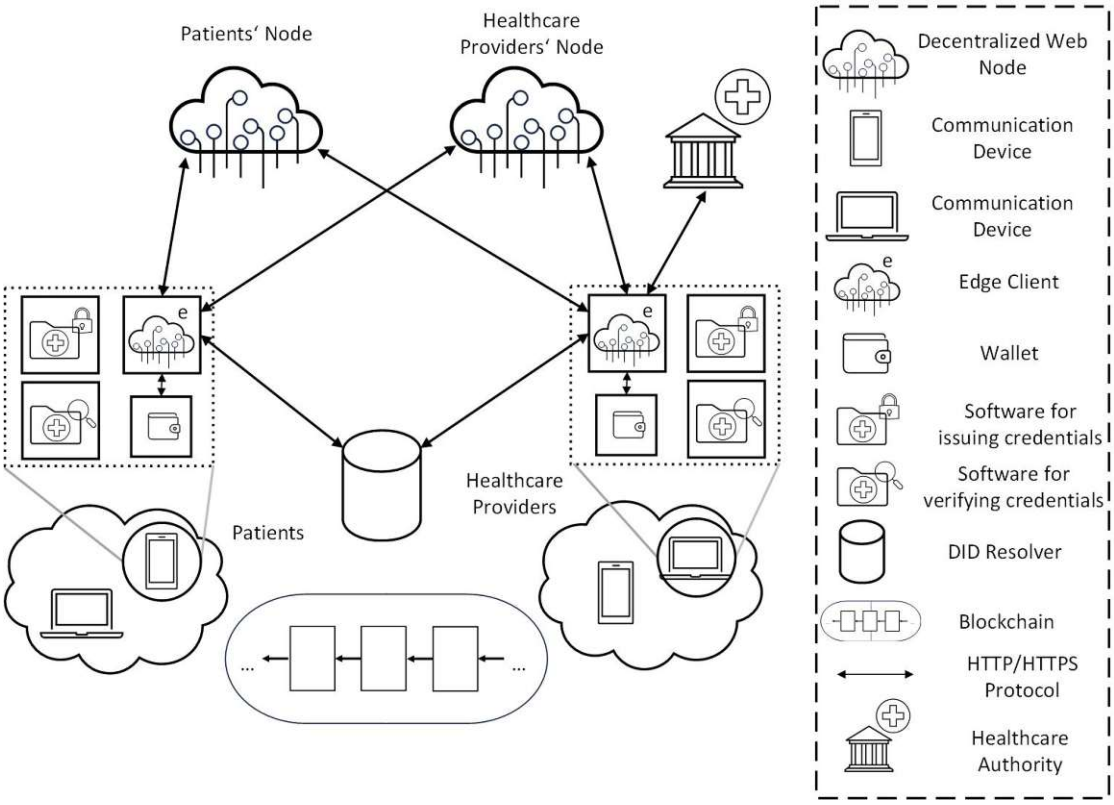

Figure 4.2: Proposed architecture

4.2.1.4 Decentralized Web Node

The DWN specification [29] outlines the concepts of a DWN, although it is still a draft and may be revised in the future. Consequently, we identify the essential components for DWNs independently if they are not clearly outlined in the official specifications. The DWN functions as a data storage and message relay system utilizing message-based communication. The base structures are called "Request Objects", which are JSON object

envelopes utilized for transmitting messages to the DWNs. The Request Objects always contain a "messages" property, which plays a vital role in our use cases by supporting various essential functionalities. These objects consist of message execution parameters, authorization material, authorization signatures, and signing/encryption information. In addition, the message objects must contain some properties, including the obligatory "descriptor" property, which describes the message, its type, and other relevant details. The "data" property holds the data, while the "processing" property contains information about the message's internal processing across the target DID's DWN. It is feasible to write data to the DWN via a different transmission path. In both cases, however, the "dataFormat" property must be specified to describe which data is associated with the message.

As described in Section 2.2 DWNs support customizable features through the Feature Detection Interface. All interfaces are special types of messages and are defined within the message's "descriptor" property. The functionality of our architecture requires the Records interface, Protocols interface and the Permissions interface. The Records interface enables creating, reading, and deleting data. The Protocols interface defines the rules for entities within the architecture to interact with the Records and the DWN. However, the Protocols interface does not cover the entire spectrum of use cases, as our requirements are too complex. In addition, the Permissions interface is required for the other use cases. To respond to interface messages there exist Response Objects containing the requested data and a status similar to HTTP codes. [29]

Records Interface

According to the DIF specification of DWNs [29], the Records interface provides a mechanism to store data relative to shared schemas. The interface includes four categories of Records: (i) RecordsQuery, (ii) RecordsWrite, (iii) RecordsCommit, and (iv) RecordsDelete. Each one of these records serves a functional purpose, equivalent to that of read, write, update, and delete.Therefore, in order to ensure the functionality of our architecture, we need to implement all four categories of records.

The RecordsWrite message writes data to the DWN with specific properties outlined in the specification. The important properties for our architecture are "encryption" and "schmema". The "encryption" property ensures that data is encrypted, using JSON Web Encryption (JWE) as described later in section 4.2.2. The "data" property contains the JWE-specified data. The "schmema" property contains a URI string that indicates the schema of the associated data. Each RecordsWrite message represents one health record entry. The RecordsQuery message reads DWN data. The "filter" property defines the criteria for filtering and querying specific Records. Criteria can be filtered individually or combined based on criteria such as ID, protocol, or schema. The RecordsCommit message represents the update functionality. The "recordId" property must be identical to that of the logical record that the entry corresponds with. The last type is the RecordsDelete message, which includes the essential details required to delete an entry.[29].

Protocols Interface

Protocols can define a structure for records, including the relationships between them, how entities interact with the protocol and data-level requirements. These protocols must be configured in advance. We set up the record structure such that there is one HealthRecordEntry object for each record entry. Each HealthRecordEntry can then contain multiple HealthCredential and HealthData entries. The details of the HealthCredential and HealthData contents are described in Section 4.2.1.5. The owner of the DWN can read any record entry written to it. No further protocol definitions are necessary because, by default, data access is limited to the DWN owner.

Permissions Interface

The Permissions interface enables entities to request access to data and functionalities of another entity's Decentralized Web Node (DWN), using a capabilities-based approach for authorization. This approach allows for the delegation of authorized capabilities to others, based on Decentralized Identifiers (DID), with the owner's consent. [29]

There are four types of permissions, which are PermissionsRequest, PermissionsGrant, PermissionsRevoke, and PermissionsQuery. PermissionsRequests are necessary for requesting permission, particularly in our use cases when a provider needs access to a patient's data. A PermissionsRequest allows the requester to specify in detail what permissions it requires, including what kind of action can be performed on which record and whether the data is encrypted or signed. The important properties for our architecture are "scope", "conditions" and "authorization". The "scope" property defines the scope of the permission, such as the record ID to access. The "conditions" property defines the conditions under which access is allowed. This includes encrypting and signing the corresponding Records requests. This is necessary for our architecture based on the Requirement **R.2.3**. The "authorization" property is necessary for the authorization process. The DWN uses the JSON Web Signature (JWS) generated from the DID document's key to determine whether the patient is authorized to access the data. The PermissionsGrant message includes granted capabilities for parties and can be created either as a reply to the PermissionsRequest message or by a user agent without the initial Permissions. It contains almost the same information as the PermissionsRequest message, with the exception of the encryption key. The data is encrypted with the key given by the DWN owner, presented in the "encryptionKey" property in JSON Web Encryption format. The PermissionsRevoke message is the last permission required to cover all use cases. It is necessary to revoke a permission. The PermissionsQuery message enables querying of Permissions. However, it is unnecessary to cover the use cases of our architecture.

The standard specification does not provide information on securely sharing a verifiable credential. As described in Section 2.2, a verifiable presentation is utilized to share the data of the VC. In our architecture, the DWN must create a VP for each request, containing the specifically chosen claims, instead of granting an entity access. Therefore, we must expand the default properties of the PermissionRequest and PermissionsGrant

message by adding a VC property that includes the relevant claims.

4.2.1.5 Data Definition

This section outlines the data structure to cover all the necessary data types of EHRs and PHRs. We categorize the data into structured and unstructured types. Structured data, according to Raghupathi and Raghupathi [95], consists of data that can be stored, queried, analyzed, and manipulated by machines. Structured data includes diagnostic results, blood cell pictures, consent forms, care plans, and laboratory orders. These documents consist mainly of text and can be interpreted by machines. Therefore, this data can be used directly as verifiable credentials utilize the advantages of VCs, including integrity and authenticity. Furthermore, storing data as a VC allows for selective disclosure and enables precise data selection to be shared. In contrast, unstructured data like images and audio recordings, which cannot be stored as VCs, will be stored as binary data. As described in Section 4.2.1.4 one health record entry is represented as a HealthRecordEntry. This HealthRecordEntry can then contain multiple HealthCredential and HealthData. The HealthCredential object represents the structured data and the HealthData object represents the unstructured data.

In the eHealth industry, various standards, including DICOM, define the format and structure of medical image data. To avoid restricting ourselves to one standard or developing a new one for medical data, we store it as binary objects in the DWN. When a user needs to access this data, the binary object is shared between the user and the DWN with the corresponding entity, preserving the data's standardized structure. However, this solution alone cannot ensure authenticity or validity of the data. Therefore, we require the sender of the data to sign the data directly on the RecordsWrite message as described in the previous section. Furthermore, to ensure the connection between the data and the functionality of a DWN, a schema must be established and utilized in the DWN's messages, particularly for the Records interface. In both scenarios, it is necessary to specify the dataFormat property to indicate which data the message corresponds to. For structured data, the specified dataFormat should be "application/vc+jwt", while for unstructured data, it should be "application/octet-stream".

The verifiable credential standard uses the JSON-LD standard [116]. As mentioned by Halpin [50], the JSON-LD standard uses the RDF format for normalizing the claims of the VC, which is necessary for signing data. However, this RDF dataset normalization may introduce potential vulnerabilities. While solutions to address these concerns are currently in development, they have not yet been fully implemented (e.g., [117]). The current JSON-LD standard (v 1.1), released on 16th July 2020 [116], still relies on the RDF dataset normalization approach as described by Halpin [50]. The new RDF dataset canonicalization approach is still working in progress and has not been implemented in the JSON-LD standard. As a result, the current standard is vulnerable for signature exclusion and signature replacement attacks according to Halpin. For these reasons we will not use the proposed JSON-LD standard for the health credentials. We would reconsider this choice if JSON-LD uses the new approach of RDF dataset canonicalization.

We discussed the issue in more detail in Section 2.2.3. As an alternative solution we use the IETF JSON Web Tokens (JWT) [67] serialization for the verifiable credentials. This approach provides the necessary functionality, including a secure issuing and verification process, along with selective disclosure. If the future versions of JSON-LD will implement RDF dataset canonicalization algorithm, the described structures can be altered to the JSON-LD format.

Structured Data

The general structure and contents of a VC are defined in the W3 specification [119]. The important part for the architecture is the content located in the "credentialSubject" property. This contains the health information associated with the subject of the VC and represents the text-based health data such as the results of a blood count. Different data structures may be utilized according to the needs of healthcare providers. The "credentialSchema" property in a VC defines a schema to determine if the credential is well formed. Such a scheme must be created to meet the general requirements of healthcare providers. It is possible to modify and expanded at a later date.

Since we do not use the VC standard with JSON-LD but via JWT, the initial process for an issuer to support selective disclosure with JWT is to generate a standard JWT credential. This credential enables the issuer to modify the JWT to a Selective Disclosure JWT (SD-JWT). The internet Engineering Task Force's most recent draft on "Selective Disclosure for JWTs (SD-JWT)" [40] specifies the mechanism for transforming our JWT-based verifiable credential to an SD-JWT. This SD-JWT is then written to the DWN of the patient. To share the information with a verifier, the verifiable credential is shared together with the claims the patient wants to disclose.

Unstructured Data

In the case the patient wants to share unstructured data, the patient provides the binary data itself. In contrast to the structured data, we do not utilize VCs. A DWN permits the storing of signed messages and data to ensure authenticity and integrity. The VC method offers the advantage of selective claim disclosure, which cannot be applied to binary data or files. Therefore, VCs only add unnecessary complexity to this case. A patient or healthcare provider directly sends the binary data to a DWN either with a RecordsWrite message or via an external channel, depending on the data size, including a signature on the RecordsWrite message.

4.2.2 Cryptographic Concepts

The architecture's security and privacy rely mainly on cryptography. We require three specific cryptographic methods, namely (i) public key signatures, (ii) public key encryption, and (iii) private key encryption. Moreover, we necessitate hash functions for the signature algorithms and the selective disclosure property in JSON Web Tokens.

4.2.2.1 Signature

In our architecture, signatures fulfill the purpose of authenticity and integrity. They are utilized for signing a verifiable credential, signing a record for the DWN, and authentication and authorization on the DWN. All signatures are based on JSON Web Signature (JWS) as described in the DID, VC and DWN specifications [29, 114, 119]. We define the usage of the ECDSA signature, which uses the elliptic curve P-256 with the SHA-256 hash function. This method is recommended by Internet Engineering Task Force (IETF) in RFC 7518 for JWS [66]. Moreover, it enables an issuer to sign a verifiable credential in the JWT form since the JWT includes a JWS [119]. The security of the ECDSA signature is based on the Discrete Logarithm problem.

All four signatures are based on the public keys provided in the DID documents of the corresponding entities. The signature of a VC and the signature of a DWN record is referring to the "assertion" property in the DID document. Furthermore, JWS is attached to every message to achieve authentication on the DWN. This public key of this signature refers to the "authentication" property in the DID document, while the authorization signature granting data access on the DWN is associated with the "capabilityInvocation" field in the DID document.

4.2.2.2 Encryption

Data encryption is necessary in the architecture for storing and transmitting data between entities, as stated in Requirement **R.2.3**. This ensures confidentiality and privacy of the user data. The encryption process uses a hybrid encryption scheme that utilizes the entity's public key for encrypting the symmetric private key, which then encrypts and decrypts the data. The encryption process is described by the Internet Engineering Task Force (IETF) in RFC 7516 for JSON Web Encryption [68]. The DWN specification [29] supports one asymmetric encryption scheme and two symmetric encryption schemes. For the asymmetric encryption scheme, we use the X25519 elliptic curve Diffie-Hellman key exchange protocol [72]. For the symmetric encryption scheme, we use the AES-GCM encryption scheme as this is recommended by the IETF in RFC 7518 on "JSON Web Algorithms" [66]. The public key required for the key exchange can be derived from the DID document's "keyAgreement" property.

4.2.2.3 Selective Disclosure in Verifiable Credentials

The concept of selective disclosure allows a patient to disclose just a subset of the claims stored in variable credentials. According to the most recent version of "Selective Disclosure for JWTs (SD-JWT)" by the IETF [40], a JWT can be generated with the ability to selectively disclose information [40]. In our architecture the patient selects which claims to disclose in a verifiable presentation that is then shared with a healthcare provider.

In the process of issuing a JWT, the issuer provides a digitally signed JSON object in the form of a JWT. In contrast to the issue of a standard JWT, the issuer provides a

signed JSON document containing digests over the selectively disclosable claims with the disclosures outside the document. This allows the patient to omit disclosures without breaking the signature of the JWT. The digest values in the JWT ensure the integrity of the disclosures and are calculated using a hash function. Each disclosure contains a cryptographically secure random salt, the claim name, and the claim value. The user can share only the necessary subset of these disclosures with the other entity. Other procedures are also specified for elements within the JWT that contain more complex data structures. [40]

4.2.3 Processes

In this section, we describe the processes of our architecture. This applies to the use cases and all the processes required for them, such as key exchange. To illustrate the interactions between the different modules in our system, we employ sequence diagrams.

4.2.3.1 Key distribution

Interactions between two entities require the respective public keys of each entity. These keys are in the other entity's DID document. Therefore, the first step is to exchange keys. Since a blockchain is used as a VDR and a DWN is also a message relay node, both entities only need each other's DID. We need to distinguish between three scenarios for the key exchange process. In the first scenario, the patient is physically present and wants to perform the key exchange in person. In the second scenario, the patient is not present and wants to perform the key exchange remotely. In the third scenario, the patient is not directly involved. This scenario describes the situation where a healthcare provider needs to share the patient's DID with another healthcare provider. Each scenario is similar in that only the DID of that entity is required.

In the first scenario, the patient is on-site with a healthcare provider. The corresponding process is shown in the diagram 4.3. The first step is for the healthcare provider to generate a QR code or link that the patient then scans with the edge agent. This QR code or link contains both the DID and a nonce to uniquely identify the request. After scanning, the edge agent sends a request with the DID to the DID resolver and receives the DID document with the service endpoint. The edge agent then sends a signed message with its DID and the nonce from the QR code to the healthcare provider's DWN. The DWN forwards the message to the provider's edge agent, which retrieves the patient's DID document from the DID resolver. This means that both entities have each other's keys. Finally, the DID documents are sent to the DWN.

In the second scenario, the patient is not on-site with the healthcare provider. There are several options for the key exchange in this scenario. One possible solution is to execute the process in the same way as in the first scenario. For example, providing a QR code or link through the healthcare provider's website. However, involving additional healthcare providers can increase the complexity of the process. In the following we describe an approach involving a second healthcare provider where every party requires access to all

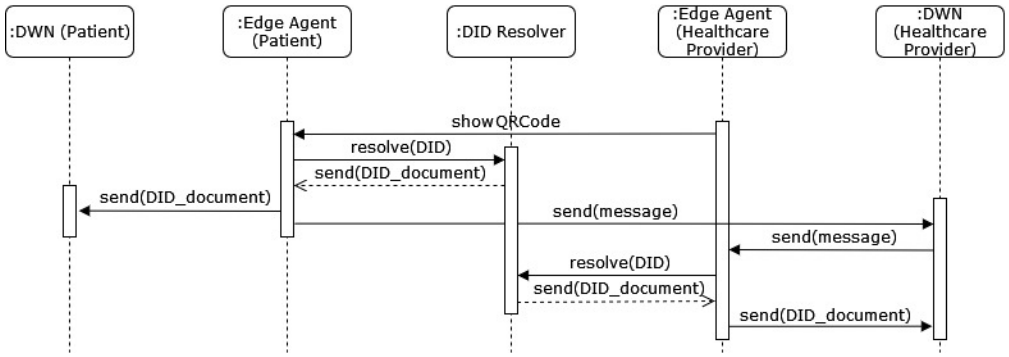

Figure 4.3: Interaction for on-site key distribution

keys. The corresponding process is shown in Diagram 4.4. For this process we assume that either the first or the second scenario was carried out before. When a healthcare provider sends the required data to the second healthcare provider, such as the requirements for a blood count, the patient's DID can be sent at the same time. This sending process can be done through an internal system or through the provider's DWN. The second healthcare provider retrieves the corresponding DID document by using a DID resolver. However, the patient also requires the DID document from the second healthcare provider. To achieve this, the first healthcare provider transmits a signed message, containing the DID of the second healthcare provider and the explanation why a second healthcare provider is involved, to the patient's DWN. The DWN forwards the message to the patient's edge client and resolves the DID document from the DID resolver. Consequently, the patient now has access to DID documents from both healthcare providers, which can be utilized for further interactions.

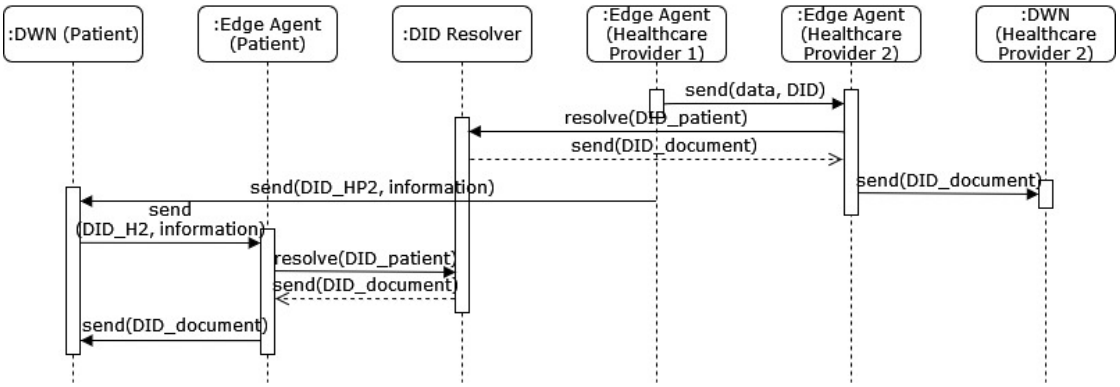

Figure 4.4: Interaction for remote key distribution with a second entity, HP...Healthcare Provider

4.2.3.2 DID Authentication

The first step in the interaction between an edge agent and a DWN is the DID authentication. This process uses the public key found in the accessing entity's DID document. Diagram 4.5 shows a simplified representation of the process. The accessing entity then generates a signature with its private key and presents it to the DWN. The DWN sees this signature as proof of identity and then issues an access token to the identity. Each message that is sent by the entity to the DWN will append an access token. This ensures authenticity without the need to authenticate the entity every time a message is sent.

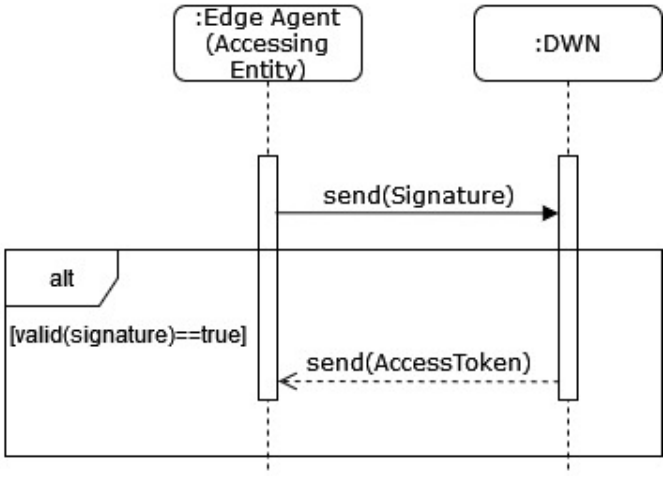

Figure 4.5: DID authentication process

4.2.3.3 Provide Access

By default, only the patient is authorized to access or make changes to the data on the DWN. Healthcare providers must request appropriate permissions for each activity, including accessing, updating, and writing data. The Permissions interface on the DWN handles these permissions. The process of creating an authorization can be initiated by either the healthcare provider or the patient, as shown in the Diagram 4.6. If the process is started by the healthcare provider, the healthcare provider's edge agent sends a PermissionsRequest message with the required authorizations to the patient's DWN, which forwards the message to the patient's edge agent. If the patient agrees to the requested authorizations, the edge agent responds with a PermissionsGrant message to the healthcare provider's DWN, which forwards the message to the healthcare provider's edge agent. Additionally, the patient's edge agent commits the PermissionGrant message to the patient's DWN. If the process is initiated by the patient, everything is done as described except that the healthcare provider does not create a PermissionsRequest message. In such cases, the Patient must select their authorization details. This process represents the use case 4.7.

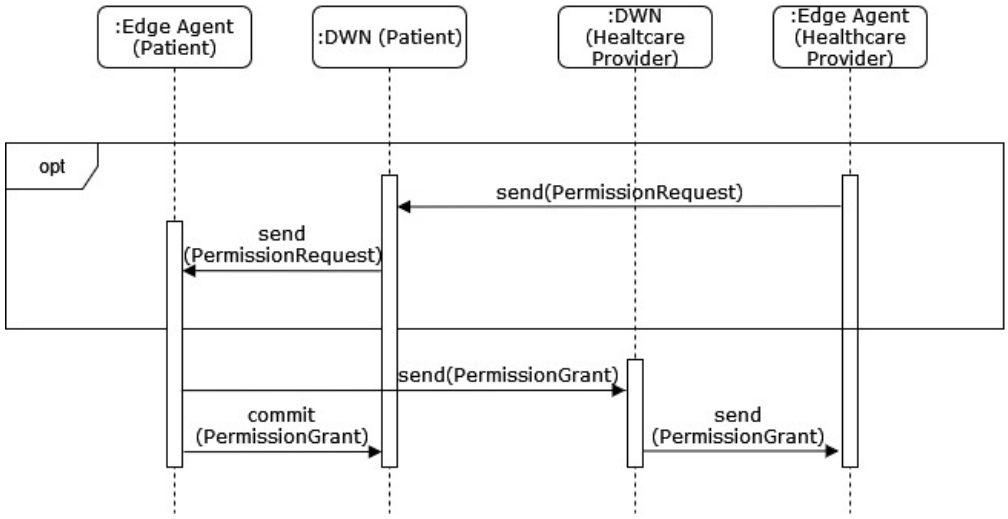

Figure 4.6: Access authorization process

4.2.3.4 Withdraw Access

The act of revoking access is a use case (4.8) that directly involves the DWN and the patient, as shown in Diagram 4.7. A corresponding access management rule must already be in place as a prerequisite for this use case. It is possible for a user of the DWN to query all Permissions beforehand to see which permissions currently exist. To revoke an active permission, the patient's edge agent sends a PermssionsRevoke message to its DWN.

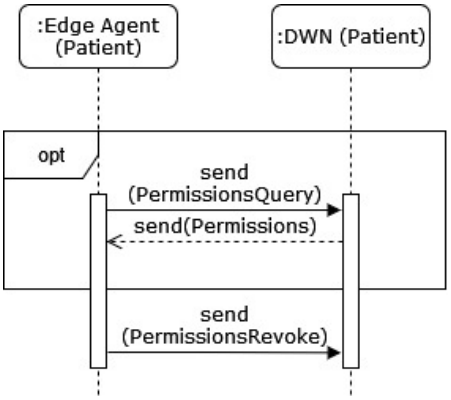

Figure 4.7: Access withdrawing process

4.2.3.5 Write Data

We have identified two use cases for the storage of patient data on the DWN. In use case 4.4, a healthcare provider stores the data on the patient's DWN, while in use case

4.9, the patient themselves stores their own data on the DWN. Use case 4.4 represents a EHR, while use case 4.9 represents a PHR. The processes of both use cases are almost identical. The only difference is that a healthcare provider must request access rights in advance, whereas patients have these by default on their DWN. The request for access rights was described in the section 4.2.3.3. Both structured and unstructured data are written to the patient's DWN using the same method. Use case 4.4 is described in the following section. Diagram 4.8 illustrates this interaction of this process.

The initial step for the healthcare provider is to generate data, differentiating between text-based data for VC issuance and other forms of data. This data is then encrypted in JSON Web Encryption (JWE) format and inserted into the RecordsWrite message. To ensure the authenticity of the data, it is signed, and the corresponding signature is inserted into the RecordsWrite message in the JSON Web Signature (JWS) format. The authorization data is also included in the message in the JWS format, as the write process is permission-based. In the case of a VC, the credential is not only stored as a selective disclosure JWT, but each disclosure is also saved.

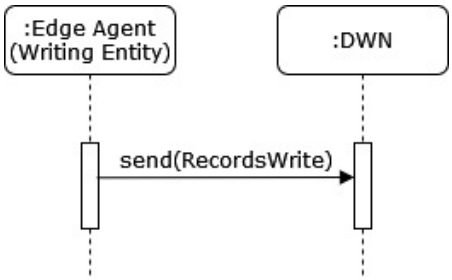

Figure 4.8: Data writing process

4.2.3.6 Reading Data

This section covers two use cases related to data reading. The first use case (4.3) involves sharing a patient's health data with a healthcare provider, while the second use case (4.5) concerns the patient's access to their own health data. The difference between these use cases is that a healthcare provider is required to request access, whereas the patient has access by default. The Section 4.2.3.3 explains the necessary procedure to request access rights prior to reading the data. It is important to distinguish between structured and unstructured data when querying the data. When querying for a VC, the DWN must provide the JWT and the permitted disclosures in the form of a VP. All other data can be sent directly. By default, the DWN does not have any private keys to decrypt data. The keys required to decrypt the data were included in the PermissionsGrant message when the access to the HealthRecordEntry was granted. When the healthcare provider reads the data, he or she must also be able to verify the authenticity of the data, such as a pharmacy verifying that the prescription was issued by a physician. For example, in a given country, a person must meet certain requirements to practice medicine as a doctor. This is regulated by an authority. This authority ensures that a healthcare provider is

indeed an authorized healthcare provider. The challenge in this context is to ensure that the record is created by an authentic healthcare provider. The solution to this problem is shown in simplified form in the diagram 4.9.

There are two possible methods to confirm that the other party is indeed a healthcare provider. Either there must be a central server that publishes a list of all healthcare providers and their associated DIDs, or there must be a way to obtain this information through verifiable credentials. To ensure that the verification of all records is not dependent on one system, we choose the decentralized approach of verifiable credentials. In this approach, each healthcare provider must know the DID of the authority that establishes their status as an authorized healthcare provider. This authority then issues a valid VC to each healthcare provider, confirming their official status as a healthcare provider. This VC is signed with the authority's keys to validate its authenticity. The healthcare provider can then publicly store the VC on a server or its DWN for anyone to access. This allows anyone to verify the validity of the credential.

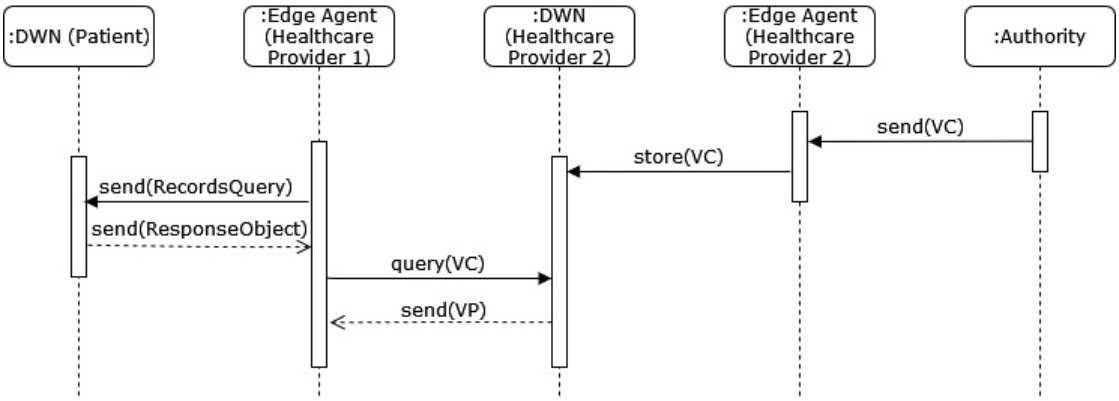

Figure 4.9: Interaction for checking validity of a record

4.2.3.7 Delete Data

The deletion of data from the DWN is restricted to the owner of the DWN as elaborated in use case 4.6. The procedure involves a direct interaction between the DWN and the patient's edge agent. The prerequisite for this use case is that corresponding data already exists in the DWN. To delete data from the DWN, the patient's edge agent sends a RecordsDelete message to its DWN, as illustrated in Diagram 4.10. This also deletes all existing permissions for this object at the same time.

4.2.3.8 Update Data

This section outlines the process for updating data as described in the use cases 4.10 and 4.11. The main distinction between the two is that healthcare providers must request access before updating data. To ensure authenticity, each data update requires a signature. To maintain the authenticity of the message, the data of one healthcare

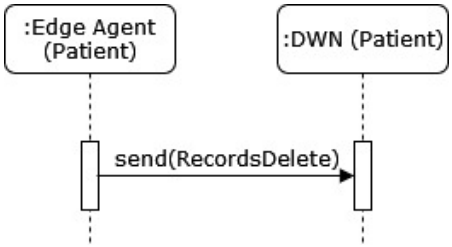

Figure 4.10: Data deletion process

provider may only be overwritten by another healthcare provider in this architecture. However, if they have the necessary authorization, a healthcare provider can overwrite a patient's data, which can be helpful in correcting any inaccuracies entered by the patient. To update data on the DWN, the edge agent of the patient or healthcare provider sends a RecordsCommit message to the DWN, which includes the updated data. This process is illustrated in Diagram 4.11

Figure 4.11: Data update process

4.2.3.9 Example of a Complete Process

The previous subsections have described the individual processes and served to answer **RQ1** and **RQ3**. In this subsection, we will utilize the defined processes to provide an overview of how they interact by visualizing an entire use case based on the proposed architecture. We have selected UC2, which illustrates a patient visiting a healthcare provider and receiving a new health record entry at the end. The complete workflow for this process is shown in Figure 4.12.

The initial step is the key exchange, represented by numbers 1 through 7. The healthcare provider initiates the key exchange by generating a QR code containing their DID. Numbers 8 and 9 represent the DID authentication process. In the subsequent step, represented by numbers 10 through 13, the patient establishes permissions for the healthcare provider. In this case, steps 10 and 11 represent requests from the healthcare provider, which may be optional if the patient already has the necessary information for the permissions. Step 14 represents the actual issuance of the data from the healthcare provider to the patient.

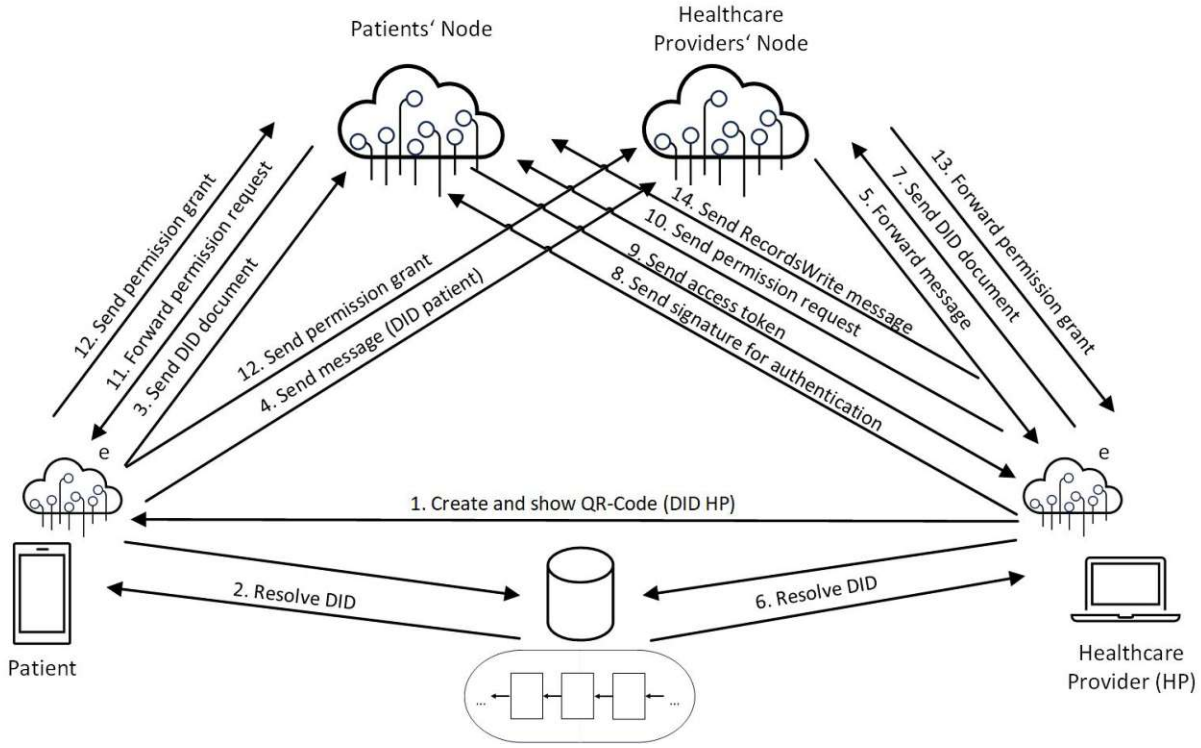

Figure 4.12: Complete workflow for use case 2

4.3 Assessment

This section evaluates the defined architecture and addresses all three research questions. Section 4.3.1 describes the implementation of the proposed concept to validate it in practice. A proof-of-concept is conducted to verify the effectiveness of the specified architecture. Furthermore, a threat model will be created in Section 4.3.2 to identify any security vulnerabilities in the architecture and suggest improvements, given the importance of security in the healthcare sector. In Section 4.3.3 we evaluate whether the architecture and implemented prototypes meet the defined requirements. Section 4.3.4 applies GDPR principles to the architecture and discusses its compliance, conflicts, and potential for improvement. Additionally, in the last part of the evaluation, in Section 4.3.5, we describe the processes and outcomes of the interviews with external experts. Their expertise is incorporated in the evaluation. This section presents a clear and concise argument for the security and functionality of our architecture, demonstrating its superiority to current standards.

4.3.1 Proof-of-Concept

This section describes the implementation of the architecture as a proof of concept to verify the defined requirements. Due to the scope of this thesis, we limit ourselves to the

requirements that can only be evaluated with concrete implementation, specifically the performance of the basic functionalities: reading (**R.1.1**), deletion (**R.1.2**), and writing (**R.1.3**) of data. However, during the implementation of these use cases, we considered all security mechanisms, including authentication, authorization, and encryption.

4.3.1.1 Implementation

The aim of the implementation is to recreate the architecture as shown in Figure 4.2. This includes two DWNs, the entities of the patient and the healthcare provider, a blockchain, and a DID resolver. According to our research, there is currently one provider of a framework for DWNs. The company that created the framework is called TBD [58]. They provide the edge agent and the DWN server, as well as the related infrastructure. They also provide a library for DIDs and VCs. For compatibility, we use both the DWN frameworks/services and the libraries for the DIDs and VCs. Since the framework is still under development, we had to implement some parts ourselves, as described later.

TBD's DWN framework uses ION [32] as its public key infrastructure. ION is an open, public, permissionless layer 2 protocol, based on the Bitcoin blockchain. It is specifically used for decentralized identifiers. It offers the possibility to create and use your own testnets. However, we have decided to use an existing test testnet from Microsoft, as it has no impact on the verifiability of the feasibility. The DID resolver is also provided by Microsoft.

To implement the architecture, we use a framework for the edge agent called Web5 SDK [110] and a deployable Docker container representing the DWN. Both frameworks are written in Typescript. Although there are some examples and templates for a web-based implementation of the framework, we decided to create a Typescript project without a web interface. The reason for this is the structure of the project and the resulting possibility for automatically evaluation.

Since the Web5 framework is still under development (version 0.83), some essential features are not yet available. In addition, not all standards were considered as defined by the Decentralized Identity Foundation [29]. Therefore, we have built workarounds to replicate the architecture as closely as possible. Web5 does not yet allow setting permissions. This is currently only possible through the Protocols interface. So, we configured the protocol to allow all entities to access the patient's data. Since authentication and authorization are still taking place, the general process and the processing time does not change. The next workaround concerns the encryption of the data. During implementation, we were unable to store the data using the specified encryption algorithms due to the lack of support for the defined elliptic curve (X25519) in the framework. As a result, we implemented our own encryption. In our defined processes, entities obtain the public keys for encryption from the DID documents. For the sake of simplicity and feasibility, we assumed in the implemented scenario that the entities already possess the public keys. The key exchange was completed beforehand, but it does not affect the evaluation of

the performance requirements. The generated keys were used to create a JSON Web Encryption Object, which was sent as the payload of the message object.

4.3.1.2 Test Procedure

For the test procedure, we utilized two clients: one to simulate the healthcare provider and the other to simulate the patient. An overview of the implemented prototype is shown in Figure 4.13. Prior to the test, we created the necessary structures and access rights for the DWNs using the Protocols interface. During the test, only the patient's DWN was used for reading, writing, and deleting data. The DWN servers were deployed as Docker containers, and we employed two different approaches to simulate two hosting situations. These are the approach of using a hosting provider, and using a fully provided infrastructure. Specifically, we utilized an Ubuntu 22.04 VM on Microsoft Azure for one approach and the server infrastructure provided by TBD for another.

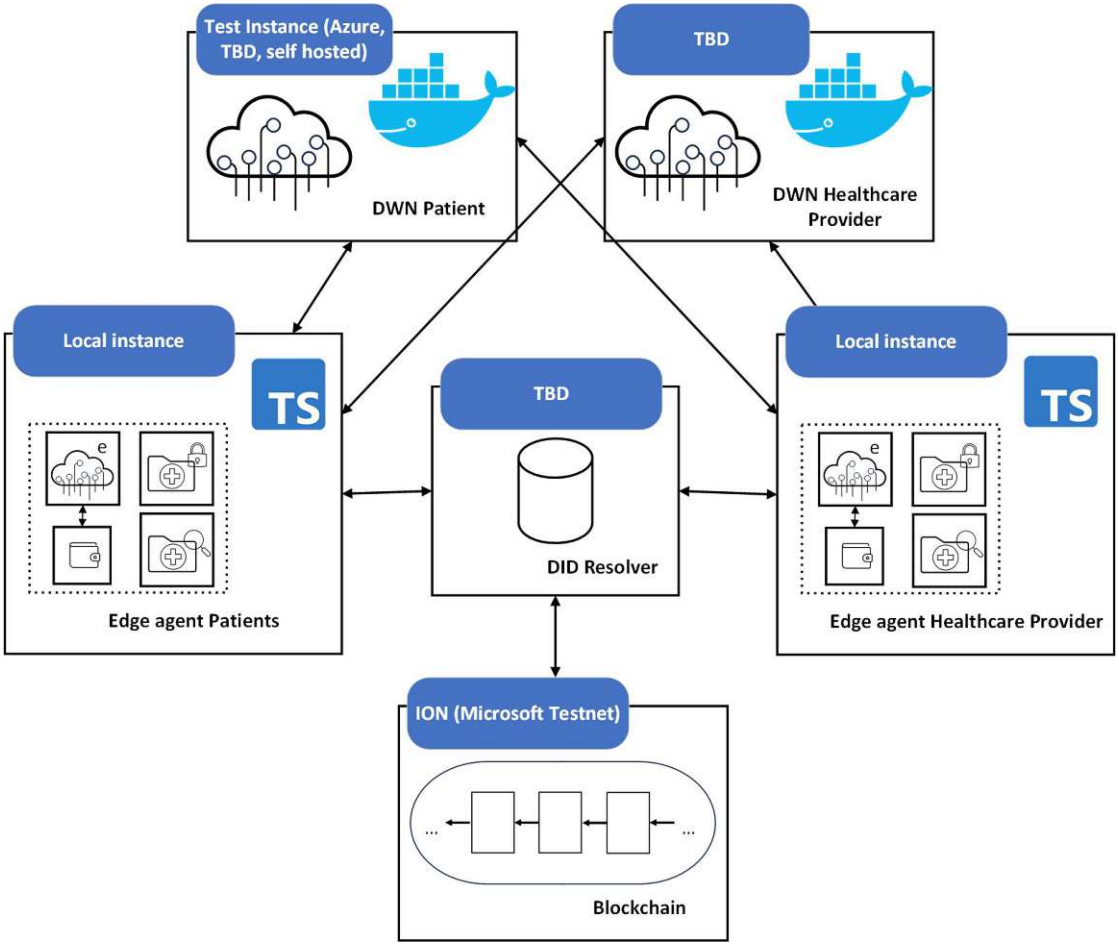

Figure 4.13: Overview of the implemented prototype

We conducted tests using both structured and unstructured data. To represent a

laboratory finding in the structured data, we created a JSON object as a payload for the VC, as shown in Figure 4.1. As for the unstructured data, we utilized a self-created JPG image with a size of 1043 bytes.

Listing 4.1: Sample data for verifable credential

```
{
    "Erythrocytes": {
        "value": "5.00",
        "unit": "T/l",
        "reference_range": "4.5 - 6.0"
    },
    "Leukocytes": {
        "value": "6.00",
        "unit": "G/l",
        "reference_range": "4.0 - 10.0"
    }
}
```

To evaluate the performance of reading, writing, and deleting data, we measured the time required for each scenario. The time required to bring the data into the transferable format (VC, blob) was included for writing data, while the time required to convert the object into a readable format was included for reading. Each scenario was repeated 1000 times for each data type. The upload rate for the client representing the healthcare provider was 20 Mbps, and the download rate was 105 Mbps. Every interaction involves the processes of authorization, authentication, encryption and decryption of data.

4.3.1.3 Test Results

The processing time measurements for the two hosting approaches are analyzed below. The results for each request type and data type are described, providing performance measurements under the tested conditions.

Microsoft Azure VM

This approach involved creating an Ubuntu 22.04 VM on Microsoft Azure and starting the DWN server as a Docker container. The scenario simulates a patient using a hosting provider to host their DWN. The DWN server was accessible via a public IP address, which was used as the serviceEndpoint in the patient's DID document stored on the blockchain. Figure 4.14 shows the processing times of the operations with the credentials and files. All average processing times comply with the defined requirements. Although there are a few outliers in the credentials and files, with request times taking a few seconds, the approach with Microsoft Azure still meets the requirements.

TBD Infrastructure

TBD offers an infrastructure for utilizing and testing SSI and DWNs, allowing developers to create applications without the need for their own setup. This approach involves utilizing the DWN servers provided by TBD to test access times, simulating a scenario in

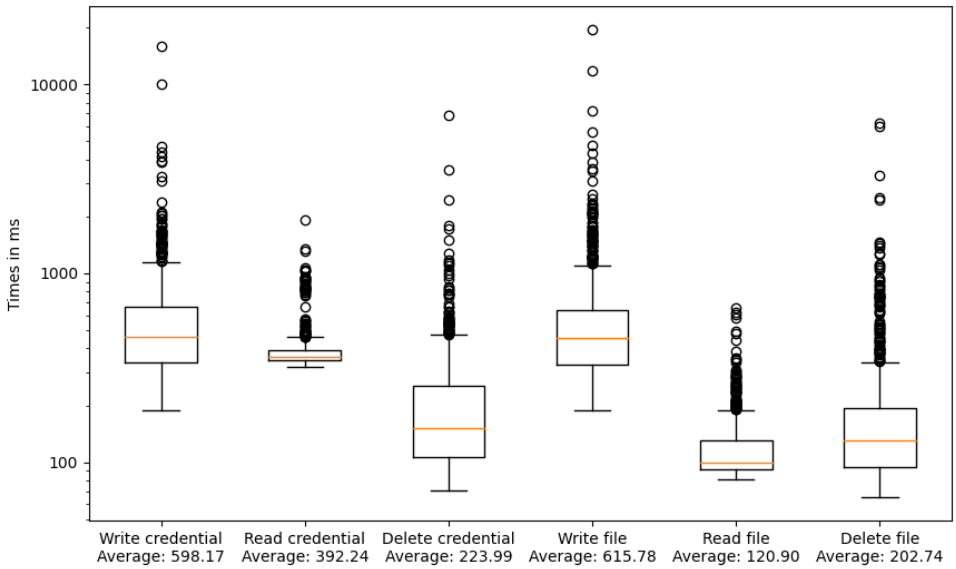

Figure 4.14: Processing times in Microsoft Azure VM

which an entity (such as a state) provides the DWN server for patients without requiring them to manage the infrastructure themselves. In this approach, all average times correspond to the defined requirements, as shown in Figure 4.15. However, compared to the Azure-based approach, the processing times are considerably higher. The boxplots show that the processing times are already higher. When comparing the values, it becomes clear that the first requests were comparably fast to the requests on Microsoft Azure. As we conducted our test over a longer period, we noticed an increase in processing times. We assume that TBD intentionally configured the servers to produce this effect.

4.3.2 Threat Model

The aim of this section is to evaluate the overall security of the architecture, identify possible vulnerabilities, and provide potential improvements. The methodology employed is threat modeling, as it allows us to identify, communicate and understand threats and mitigations. Furthermore, the threat model represents all the information that affects the security of an application in a structured way [33]. For creating the threat model we chose the Four Questions Framework by OWASP [33] as described in Section 4.1.3.1. We structure this section into these four steps representing the four questions of: (i) What are we working on?, (ii) What can go wrong?, (iii) What are we going to do about it?, and (iv) Did we do a good job?.

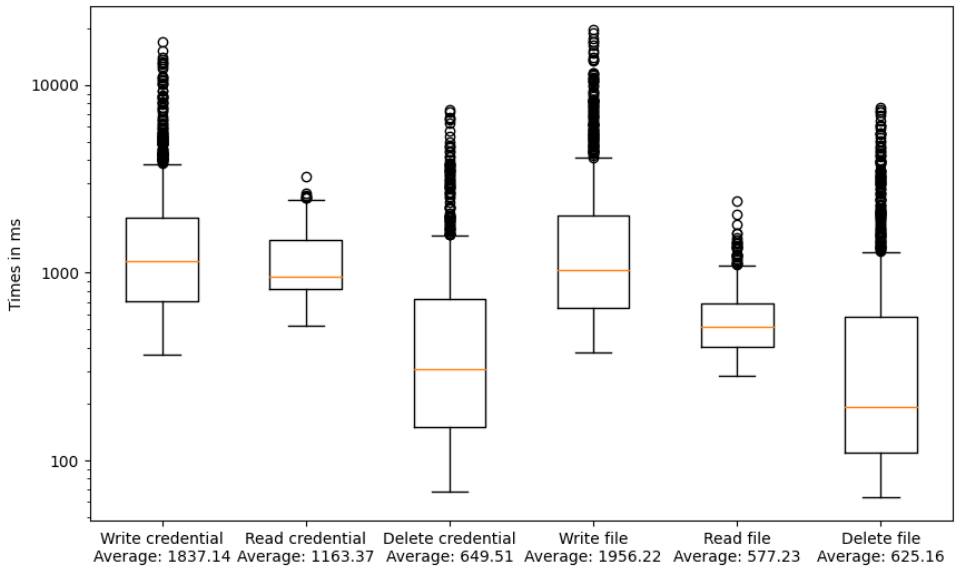

Figure 4.15: Processing times using the TBD infrastructure

4.3.2.1 Application Assessment and Decomposition

The creation of the threat model impacts the entire application. Therefore, there are no restrictions on the scope of the architecture. The first step in the threat model creation process is to gather information about the system and its individual components. Since we defined the architecture, we already possess the required information on use cases, workflows, stakeholders, and external interactions. As per OWASP's threat model process, we must identify entry points to determine where potential attackers could interact with the application, identify assets, and determine trust levels.

A potential attacker can interact with the four main components of the architecture, which are the DWN, the edge agent, the DID resolver, and the blockchain. Except for the edge agent, all are publicly accessible. Regarding the architecture, an attacker may be interested in two pieces of data: the medical data stored in a DWN, and the private keys managed by an edge agent. The data may contain information that could be useful to an attacker. Whereas the private keys provide access to the data and, in the case of a healthcare provider, access to multiple DWNs and the ability to create fake data. Another potential targets are the DID resolver, as it can deliver fake DID documents, and the blockchain, as manipulation can compromise the integrity of the entire architecture.

To understand the data flows in our architecture we utilize a data flow diagram (DFD), as shown in Figure 4.16. As described by OWASP [34] a DFD shows the different paths through the system and highlights the privilege boundaries. The proposed DFD describes

the system interaction between two entities and will help to identify the vulnerabilities. These two entities are represented as external entities in the DFD and interact with other services through the edge agents. The edge agent may be a mobile app on a smartphone, requiring user authentication, such as a fingerprint sensor. This authentication represents the first trust boundary. The DID resolver and corresponding blockchain node storage are publicly available and have no trust boundaries. The edge agent's wallet component contains the actual keys. This wallet may be hardware or software-based and can be protected by a password or biometric sensor, representing an additional trust boundary. To access the DWN service and the data stored in the DWN database, an edge agent has to authentication and authorization to the DWN. These two mechanisms represent two additional trust boundaries.

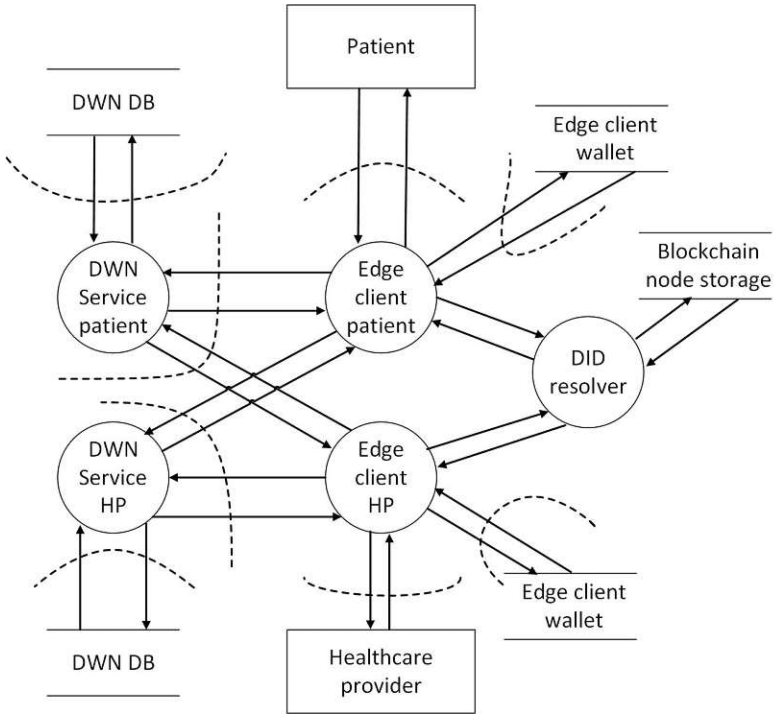

Figure 4.16: Data flow diagram of the proposed architecture, HP...Healthcare provider

4.3.2.2 Threat Identification

To identify the threats, we use the methodological approach of STRIDE, as described by Kohnfelder and Garg [70]. In the following subsections we will determine the threats for each of the types of STRIDE based on the DFD from the previous section. Some of the described threats in the following section are based on the identified threats on SSI by Grüner, et al. [48]. We adopted these threats to our architecture and mark them in the corresponding bullet points. Additionally, we calculate the corresponding CVSS v3.1 Base Score, which describes the severity of the threat to rate each of the threats.

These scores evaluate the threat on the condition that a matching vulnerability exists for a threat. For this reason, some parameters are not final, and the score serves only as a guideline. Table 4.16 contains an overview of each of the types of STRIDE. The table's structure is based on the one described by OWASP [34].

Type	**Description**	**Security Control**
Spoofing	Spoofing is a threat in which an attacker impersonates a valid system user or resource to gain access to the system. [70]	Authentication
Tampering	Tampering is the unauthorized modification of data. This includes altering files, memory spaces, or network configurations or communications. [104]	Integrity
Repudiation	Repudiation refers to the threat of performing prohibited or malicious operations in a system that cannot trace these actions. [34]	Non-Repudiation
Information disclosure	Information disclosure allows entities to get information who are not supposed to be. [70]	Confidentiality
Denial of service	The threat of denial of service temporarily disables or make it unusable such that valid users could not access it normally. [70, 104]	Availability
Elevation of privilege	Elevation of privilege refers to the threat of an unprivileged user gaining privileged access to obtain unauthorized information or compromise a system. [34, 70]	Authorization

Table 4.16: Overview of STRIDE types

Spoofing

- **Acquire private keys [48]:** The private keys allow an entity to access the DWN and the stored data. An acquisition of the private keys would allow any entity to access the DWN of the owner and the DWNs with permitted access to these keys. CVSS-B: 6.4 (AV:L/AC:H/PR:H/UI:N/S:U/C:H/I:H/A:H)

- **Steal or covertly access edge agent device [48]:** An attacker could steal the user's device and interact with the DWNs by using the edge agent on the device. CVSS-B: 6.1 (AV:P/AC:H/PR:H/UI:N/S:U/C:H/I:H/A:H)

- **Exploit recovery mechanism [48]:** A key recovery mechanism enables a user to restore keys that are stored on an edge client. This mechanism could be exploited by an attacker to obtain private keys. CVSS-B: 7.5 (AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:H)

- **Misuse private keys [48]:** A member of the healthcare provider with access to the private keys can misuse them to sign incorrect data or messages. CVSS-B: 6.8 (AV:P/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

- **Spoofing VDR entries [48]:** A vulnerability in the blockchain consensus algorithm or DID method implementation may allow an attacker to update the DID document and change the valid keys. CVSS-B: 6.6 (AV:N/AC:H/PR:H/UI:N/S:U/C:H/I:H/A:H)

- **Spoofing endpoint [48]:** An attacker could create a fake DWN which is reachable under the URL in the DID document. A possible attacker is, for example a malicious hosting provider. CVSS-B: 5.7 (AV:L/AC:H/PR:H/UI:N/S:U/C:H/I:N/A:H)

Tampering

- **Tampering communication:** Similar to the "Spoofing endpoint" from the previous section, an attacker could use this fake endpoint to forward a wrong DID in the key exchange mechanism. CVSS-B: 6.0 (AV:L/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:H)

- **Alter the message and VC value [48]:** To gain an advantage, an attacker may alter the values of an existing message or credential. CVSS-B: 4.1 (AV:L/AC:H/PR:H/UI:N/S:U/C:N/I:H/A:N)

- **Tampering DID resolver:** To distribute fake DID documents, an attacker can take over a DID resolver service which returns the wrong DID documents for given DIDs. CVSS-B: 8.0 (AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:H)

Repudiation

- **Repudiation of VCs and messages [48]:** A malicious entity can delete VCs or messages, issued by an healthcare provider to deny its existence. CVSS-B: 2.0 (AV:L/AC:L/PR:H/UI:R/S:U/C:L/I:N/A:N)

- **Repudiation of DID resolver:** A compromised DID resolver could repudiate that it send a specific DID document for a requested DID as the DID resolver does not sign the DID document before sending it. CVSS-B: 5.7 (AV:L/AC:L/PR:H/UI:N/S:C/C:L/I:L/A:L)

Information Disclosure

- **Gain unauthorized access [48]:** An adversary who can circumvent the access control on a DWN can obtain confidential data. CVSS-B: 4.1 (AV:L/AC:H/PR:H/UI:N/S:U/C:H/I:N/A:N)
- **Request unnecessary data [48]:** A healthcare provider with malicious intent may request unnecessary data, resulting in the disclosure of additional information. CVSS-B: 4.4 (AV:L/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:N)
- **Traffic interception [48]:** Communication between an edge client and a DWN can be intercepted, allowing an attacker to obtain metadata of a message object if the communication contains a message. CVSS-B: 4.4 (AV:P/AC:H/PR:H/UI:N/S:U/C:H/I:N/A:L)

Denial of Service

- **Steal or break edge agent device [48]:** If an entity no longer possesses its device, it cannot interact with the architecture or access the data. CVSS-B: 6.1 (AV:P/AC:H/PR:H/UI:N/S:U/C:H/I:H/A:H)
- **Delete private key [48]:** An attacker could delete the private keys from an entities device to prevent the entity to access its data. CVSS-B: 4.1 (AV:L/AC:H/PR:H/UI:N/S:U/C:N/I:N/A:H)
- **Access revocation [48]:** A person with malicious intent who has access to the edge agent could delete the permissions, thereby denying access to the DWN and preventing other entities in the ecosystem from accessing the data stored on the DWN. CVSS-B: 4.1 (AV:L/AC:H/PR:H/UI:N/S:U/C:N/I:N/A:H)
- **Data revocation [48]:** An attacker who has write access to the data could override an existing data record to prevent other entities and the owner from reading the correct data. CVSS-B: 5.3 (AV:L/AC:H/PR:L/UI:N/S:U/C:N/I:H/A:L)
- **Invalidate DID documents:** One DID method describes the deactivation of a DID document on a verifiable data registry, which prevents a DID resolver to resolve a valid DID document. Without a valid DID document the entity could not interact with other entities from the ecosystem. An attacker could use a vulnerability in the DID method to deactivate DID documents of other users. CVSS-B: 4.4 (AV:N/AC:H/PR:H/UI:N/S:U/C:N/I:N/A:H)
- **DoS of DID resolvers:** The architecture includes one or multiple DID resolvers. For example, an attacker can overload all instances of DID resolvers to prevent the resolving process for DID documents. CVSS-B: 6.8 (AV:N/AC:H/PR:N/UI:N/S:C/C:N/I:N/A:H)

- **Malicious hosting provider:** A malicious hosting provider could prevent the communication between a DWN and an edge agent. CVSS-B: 4.1 (AV:L/AC:H/PR:H/UI:N/S:U/C:N/I:N/A:H)

Elevation of Privilege

- **Take-over role [48]:** A malicious actor working for a healthcare provider could take over a higher privileged role to execute actions. CVSS-B: 6.7 (AV:L/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H)
- **Upload malicious files:** An attacker in the role of the patient can upload a malicious file to their own DWN. This file contains malware. When the file is read by a healthcare provider, the provider becomes infected and the attacker gains control over the provider. CVSS-B: 8.0 (AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H)

4.3.2.3 Determine Countermeasures and Mitigations

This section aims to identify countermeasures and mitigations for the identified threats. If a threat is found without any suitable countermeasures, it represents a vulnerability of the architecture and highlights the need to find suitable mechanisms.

Spoofing

Five out of the six spoofing threats describe the risk of taking over an entity's private keys. The threat of "Acquire private keys" is a general term and does not represent a specific threat. Therefore, a concrete countermeasure does not exist. The private keys are secured in the wallet of the edge agent. For accessing these keys, an attacker must overcome the security mechanism of the wallet itself, even if the attacker steals the edge agent device. A secure access mechanism, such as biometric data, helps to prevent access to the private keys. In addition, a remote revocation feature, as proposed by Grüner et al. [48] could help to delete the keys from a stolen device. With a CVSS-B score of 7.5 is the exploit of the recovery mechanism the highest rated spoofing threat. The security of the private keys depends on the specified recovery mechanism. For example, if a seed phrase is used to recover the private keys, the security of the private keys depends on the complexity of this seed phrase. A mitigation to this thread is to set a required minimum length for the seed phrase. However, a concrete countermeasure does not exist. The risk of "Misuse private keys" is a structural issue. Healthcare providers must trust their employees to handle patient data appropriately. An access and logging system for private keys could help healthcare providers mitigate security threats. The threat of "Spoofing VDR entries" is considered low, as an attacker must overcome the consensus algorithm or find an implementation vulnerability in the DID method. An analysis of the algorithm and the implementation mitigates the risk of a threat. In the event of endpoint spoofing, the attacker is typically the hosting provider who creates a fake DWN. To mitigate this risk, it is important for the user to ensure that the hosting provider is

trustworthy. Other scenarios of endpoint spoofing are not unique to this architecture and will not be discussed further.

Tampering
As described in the previous section a verification of trustworthiness of the hosting provider could mitigate the threat of "Tampering communication". To counter the threat of "Alter the message and VC value" it is recommended to use a secure signing algorithm as described in Section 4.2.2. This ensures that a message cannot be forged. With a CVSS-B score of 8.0 the threat of "Tampering DID resolver" has the highest score of all threats. The security of the entire architecture may be compromised if the DID resolvers are compromised. To counter this, it is recommended to use a dedicated DID resolver and host the blockchain. Alternatively, an edge agent can assign multiple DID resolvers to the resolving process for a request and verify that the result contains the same keys.

Repudiation
Based on national laws, a healthcare provider has the obligation to keep health documents for a specified amount of time. To mitigate the threat of "Repudiation of VCs and messages", the concerned entity could interact with the issuing entity to confirm the existence of the VC or message. The countermeasure to the threat "Repudiation of DID resolver" is, as already mentioned, to use your own DID resolver or to query the same DID document from different trusted DID resolvers.

Information Disclosure
The risk of "Gain unauthorized access" does not have a concrete countermeasure, as it generally highlights the problem of information disclosure, but does not describe a specific attack scenario. Since the access mechanisms protect against unauthorized access, analyzing them is an effective mitigation measure regardless of the specific threat. To mitigate the threat of "Request unnecessary data", the application's UI design should be clear and display a list of permissions. This issue does not require technical expertise but can be addressed through user awareness. To mitigate the risk of "Traffic interception" it is recommended to use encrypted communication protocols such as HTTPS or WSS, as defined in the architecture.

Denial of Service
A countermeasure for the "Steal or break edge agent device" threat is to implement an appropriate recovery mechanism, as described by Grüner, et al. [48]. This countermeasure also helps in case an attacker deletes the private keys on a device.As previously mentioned in the spoofing section, implementing a security mechanism for the edge agent and wallet is an effective countermeasure to prevent attackers from accessing private keys. This also mitigates the risk of "Access revocation". Given that healthcare providers are only allowed to overwrite and not delete data in a patient's DWN, versioning is a countermeasure to prevent the risk of "Data revocation". This ensures that patients have access to previous versions of their documents and can verify if any data has been

overwritten incorrectly. To mitigate the risk of invalidating DID documents, appropriate access mechanisms and implementation analysis are necessary for DID methods. The most promising countermeasure to the threat of "DoS of DID resolvers" is to use a dedicated DID resolver and host the blockchain. It is worth mentioning that the more DID resolvers there are, the less threatening this issue becomes, as anyone can host the blockchain and act as a DID resolver. Furthermore, the threat of a "Malicious hosting provider" can be countered by the self-hosting approach for the DWN. It is important to note that not all users are capable of self-hosting, so it is recommended to ensure that the hosting provider is trustworthy.

Elevation of Privilege
To counter the threat of a "Take-over role", appropriate access mechanisms should be implemented to restrict access to private keys to authorized members only. The highest ranked threat, with a CVSS-B rating of 8.0, is "Upload malicious files". As explained in a later subsection of this section, the interviewed expert EX3 mentions that current systems use anti-virus software to scan data before storing it on centralized servers. Since this infrastructure is decentralized and we must assume that the patient could be potentially malicious, the anti-virus software should be used before accessing the files on the edge client of the accessing entity. This would mitigate the risk.

4.3.2.4 Verification

Table 4.17 shows an overview of the identified threats, their CVSS-B ratings, and their countermeasures and mitigations. The biggest threats based on the CVSS-B ratings are "Tampering DID resolver", "Upload malicious files", and "Exploit recovery mechanism". The architecture's weakest component, as evidenced by multiple threats, is the DID resolver. This component is centralized, unlike the rest of the architecture, making it a single point of failure. To maintain the security of the architecture, it is necessary to have multiple DID resolvers. The risk of "Exploit recovery mechanism" poses a significant threat if left unaddressed. It is crucial to use a recovery mechanism with a to prevent potential damage. The security of the private keys relies on this mechanism. Therefore, it is necessary to define a minimum according to current standards in a specific application. Many risks, such as "misuse of private keys" or "tampering communication", are inherent to the structure and cannot be entirely prevented by technical measures. These risks rely on trust, such as trusting that healthcare provider staff with access to private keys will use them correctly. Mitigations, such as access and logging mechanisms, can help reduce these risks. These issues are not specific to eHealth architecture and can occur in any field. It is important to note that all other threats have corresponding countermeasures and mitigations to ensure the security of the architecture.

STRIDE category	Threat	CVSS-B rating	Countermeasure & mitigation
Spoofing	Acquire private keys	6.4	No concrete countermeasure or mitigation
	Steal or covertly access edge agent device	6.1	Access mechanisms in edge agent and wallet & Remote revocation
	Exploit recovery mechanism	7.5	No concrete countermeasure or mitigation
	Misuse private keys	6.8	Access and logging mechanism
	Spoofing VDR entries	6.6	Analysis of the used VDR
	Spoofing endpoint	5.7	Ensure trustworthiness of hosting provider
Tampering	Tampering communication	6.0	Ensure trustworthiness of hosting provider
	Alter the message and VC value	4.1	Secure signing algorithm
	Tampering DID resolver	8.0	Usage of own DID resolver & Query DID document from multiple DID resolvers
Repudiation	Repudiation of VCs and messages	2.0	Interaction between concerned entities
	Repudiation of DID resolver	5.7	Usage of own DID resolver & Query DID document from multiple DID resolvers
Information Disclosure	Gain unauthorized access	4.1	No concrete countermeasure or mitigation
	Request unnecessary data	4.4	UI design of the application
	Traffic interception	4.4	Communication protocols with encryption
Denial of Service	Steal or break edge agent device	6.1	Recovery mechanism
	Delete private key	4.1	Recovery mechanism & Access mechanisms in edge agent and wallet
	Access revocation	4.1	Access mechanisms in edge agent and wallet
	Data revocation	5.3	Versioning
	Invalidate DID documents	4.4	Access mechanism for update & Analysis of implementation
	DoS of DID resolvers	6.8	Usage of own DID resolver & Redundancy of DID resolver
	Malicious hosting provider	4.1	Ensure trustworthiness of hosting provider
Elevation of Privilege	Take-over role	6.7	Appropriate access mechanisms
	Upload malicious files	8.0	Anti-virus software on edge clients

Table 4.17: Overview of identified threats

4.3.3 Requirement Evaluation

Section 4.1.2 defines 25 requirements based on expert interviews and existing literature. Each requirement is analyzed to determine if the defined architecture meets it. An overview of the evaluation of requirements is provided at the end of this section.

4.3.3.1 Data Lifecycle

The following 10 requirements represent the non-functional requirements related to health data. This involves the criteria of scalability, flexibility, low latency, persistence, and other performance-related criteria.

R.1.1 The average time it takes to read a record entry with the least amount of data possible is 2 seconds
Section 4.3.1 practically evaluated this requirement. Each of the three approaches used met the requirement of a 2-second average access time for reading data.

R.1.2 The average time it takes to delete a record entry with the least amount of data possible is 1.1 seconds
As with the previous paragraph, this requirement was also evaluated in practice and described in Section 4.3.1. The average processing time for a deletion request is less than the defined duration of 1.1 seconds, thus fulfilling the requirement.

R.1.3 The average time it takes to write a record entry with the least amount of data possible is 8.3 seconds
The final requirement, which was evaluated in Section 4.3.1, tests the average duration of the process of creating a new record entry. This process fulfills our defined requirement and takes less than the average time of 8.3 seconds.

R.1.4 Access management operations (i.e., allow access and delete access) have a average duration of 2 seconds
Due to the framework used, we were unable to carry out the measurements as the required access management operations were not implemented. Implementing the entire Permissions interface would exceed the scope of the proof-of-concept. However, as all other measurements meet the defined requirements and an access management request involves a similar amount of data as a delete request, we estimate that the average processing time for an access management request will also fall within the defined time window. However, this assertion still needs to be verified, so this requirement cannot be confirmed with certainty.

R.1.5 The architecture allows to query for one record entry per second
As demonstrated in the measurements outlined in Section 4.3.1, a read request typically takes less than one second. The only exception is the credential read request in the infrastructure provided by TBD, where the average request takes 1.163 seconds. It is

important to note that this requirement does not mandate the delivery of data within one second, but rather that an entity can send a read request to the DWN every second. A web server can communicate with a client through either a websocket or an HTTP connection. For instance, a client can send a read request every second, which the server can process with adequate resources. Thus, we can conclude that the requirement has been fulfilled if it is properly implemented.

R.1.6 The architecture allows to handle five data uploads simultaneously
Similar to the previous requirement, the fulfillment of the requirement depends on the implementation. A DWN can manage multiple independent sessions, allowing for simultaneous uploads from different entities. Additionally, depending on the implementation, a single entity can upload multiple files simultaneously. Therefore, this requirement is fulfilled if enough resources are available, and it is properly implemented.

R.1.7 The architecture allows one deletion requests per second
Section 4.3.1 shows that a deletion request takes less than one second on average. However, the requirement is not for the data to be deleted within one second, but for an entity to be able to send a deletion request to the DWN every second. As with requirement **R.1.5** and **R.1.6**, the fulfillment of this requirement depends on the implementation and can be accomplished through websocket (WSS) or HTTP/HTTPS requests. Therefore, we consider the requirement fulfilled here.

R.1.8 The digital healthcare system has an availability of 98% per day
An availability of 98% per day is equivalent to 175.2 hours of downtime per year. During this time, the system must be maintained, and updates installed. To ensure availability, there exists the Sync interface [29] for DWNs. This allows multiple DWNs to communicate and synchronize. According to the specification, availability is guaranteed by the redundancy of the sync interface. A DID document allows for the specification of multiple service endpoints, which can be used to interact with an entity, providing redundancy for the patient. Additional security and monitoring mechanisms of the hosting instances make it possible to ensure availability. This requirement can only be evaluated retrospectively. However, different hosting and redundancy strategies can make it possible to calculate the availability on a particular instance using empirical values. We conclude that this requirement can neither be confirmed nor refuted.

R.1.9 A complete backup of the system's data is performed every day at 11:55 p.m
DWNs have a built-in synchronization mechanism through the sync interface. One possible backup strategy is to create an additional DWN instance that is not accessible from the outside but synchronizes with the other instances every day at 11:55 p.m. This DWN instance serves as the backup. Additional backup strategies, such as offline backups, can also be implemented. Therefore, this requirement can be met with the appropriate configuration.

R.1.10 A mechanism exists for the recovery of access keys in case of loss
The private keys corresponding to the public keys published in the DID document serve as access keys to the data. These private keys are stored in the wallet on the edge agent device. According to Preukschat and Reed's [94], there are various methods to guarantee key recovery. Firstly, the keys can be exported from the edge agent and stored in a secure location. Adding password protection to the key file provides an extra layer of security. Another option is to use a seed phrase to recover the private keys stored in the wallet. Additionally, there are concepts such as social recovery and multi-device recovery that use the principle of secret sharing. Our architecture enables key recovery, and the specific mechanism chosen becomes relevant only during implementation. Therefore, this requirement is fulfilled by one of the mechanisms described.

4.3.3.2 Trust, Security and Privacy

These 10 requirements cover all the security related requirements, such as identity management, privacy, and access control.

R.2.1 The patient's own data in the system can only be accessed with the patient's matching cryptographic keys in the standard configuration of the healthcare system
The Permissions interface allows a patient to configure access for other entities to access the patient's data. In the initial situation, unless otherwise configured, only the patient (owner) is allowed to manage the data. However, if the Protocols interface has been used to define a protocol and appropriate permissions have been defined for it, then other entities will also have access to the data. Therefore, this requirement is met unless otherwise configured.

R.2.2 Another entity can access the patient's data only if the patient sets up an access rule. In this case, the data can be retrieved using the entity's cryptographic keys
This requirement is met by using the Permissions interface. The process showing how this works is described in Section 4.2.3.3. This authorizes another entity to access the corresponding data.

R.2.3 Patient data can only be written to the system in encrypted form and read in encrypted form using a secure encryption mechanism for today's standards
According to the specification [29], a DWN cannot be configured to allow only encrypted communication. This is the responsibility of the entities. They must ensure that they only create JWE objects that contain the encrypted data. Since the specification is still a draft and not everything has to be done according to the specification, care can be taken in a concrete implementation to ensure that only valid JWE objects can be stored. This way, this requirement can also be met. The JWE allows the definition of different

encryption methods. As described in section 4.2.2, we have chosen to use the AES-GCM encryption scheme.

R.2.4 The system's standard configuration of the healthcare system restricts any entity other than the data owner (patient) from querying information of any kind, including meta data
As described in the evaluation of requirement R.2.1, access to the data itself is not possible without appropriate permission. This also applies to metadata. However, we must also ensure that no other entity that has access to the communication between a DWN and a patient, such as a DWN hosting provider, can extract metadata from the communication. Since the data itself is encrypted, nothing can be extracted from it. However, the message descriptor object is critical in this context. The message descriptor is transmitted in clear text and can be viewed as metadata, for example, an attacker can infer what information the message contains from the "schema" property. To prevent this, clear text communication between the DWN and the edge agent must not be used. An encrypted connection, such as HTTPS or WSS, must always be used. The keys required for HTTPS are obtained by the sender from the DID document. This approach fulfills this requirement.

R.2.5 Log entries feature neutral IDs that prevent drawing conclusions about individual data entries or users
Logging is an implementation issue. Therefore, compliance with this requirement depends on the specific implementation. From an architectural perspective, we specify that only neutral IDs may be used in a log entry.

R.2.6 Patients can create an access rules for their data that enables entities to access a single health record entry
As described in the evaluation of requirement R.2.2, the Permissions interface provides a way to define which entity has access to which data. This requirement is therefore satisfied. The process showing how this works is described in Section 4.2.3.3.

R.2.7 Patients can delete an access rule for their data and revoke the access for an entity
In Section 4.2.3.4, we describe the process for an entity to withdraw an active access management rule. This access management rule authorized an entity to access data on the DWN. Therefore, this requirement is satisfied.

R.2.8 An entity can determine the correctness and the issuer of a health record entry
To evaluate this requirement, we must distinguish between the two types of data, structured and unstructured. For unstructured data, the data is stored directly on the DWN as a RecordsWrite message. The integrity and authenticity of the message is ensured by a signature. Structured data is stored as VCs. In response to a request, the DWN

sends a VP of the credential. This VP contains the issuer and a signature that ensures integrity and authenticity. Thus, both types of data satisfy this requirement.

R.2.9 The system architecture must incorporate at least three security mechanisms (such as authentication, authorization, and encryption) for data access

The proposed architecture employs three distinct security measures, namely authentication, authorization, and encryption, which an attacker must overcome to access the data. Signatures ensure authentication and authorization, which are checked by the DWN upon access. However, these signatures are based on different private keys, requiring an attacker to possess two different private keys. The third mechanism involves encrypting the data, which requires a third private key to decrypt the data sent in the form of JWE. This fulfills the necessary requirement.

R.2.10 The operator of a digital health system is only able to view an encrypted binary blob from which no information can be extracted
Regarding the defined architecture, this requirement specifies that a hosting provider must not be able to extract any information from the stored data in the DWN. It is crucial to consider the implementation and availability of the DWN. For instance, if the DWN uses an unencrypted database that the service provider can access without much additional effort, the provider can read the message descriptor and access more than just a binary blob. This requirement cannot be evaluated without a concrete implementation.

4.3.3.3 Human-Related Issues

The following 5 includes the needs and capabilities of patients and healthcare providers. Specific requirements are accessibility, usability and regulatory measures.

R.3.1 The architecture allows patients to store their medical record entries securely for at least seven years
From a functional perspective, meeting this requirement is achievable with suitable backup and storing mechanisms.

R.3.2 The architecture allows the patient to specify a different key pair to access the health data
To assess this requirement, it is essential to have the ability to modify the keys for the three security mechanisms. The keys for authentication and authorization are obtained from the DID document by the DWN. Therefore, updating the DID document is sufficient to change these keys. The data is stored on the DWN as a JWE object, which is encrypted with a symmetric key (Content Encryption Key, CEK). This key is then encrypted individually for each participant (Key Encryption Key, KEK). If only the KEKs require changing, updating the keys in the DID document and the 'recipients' property of the JWE object is sufficient. Additionally, the record or permission must be updated. A

problem arises when the CEK needs to be changed. Although there are approaches ([25, 98]) for symmetric key re-encryption directly on the DWN without decryption and encryption, these approaches do not have widespread practical application. The solution for changing the CEK involves downloading the data, decrypting it, encrypting it with a new CEK, and uploading it again. Although this solution is inconvenient, it allows the patient to change all used keys, partly fulfilling the requirement.

R.3.3 The architecture allows the healthcare provider to create an object, which is only visible in its entire form
This requirement can be met by appropriate implementation. The required data structure or permissions for these objects must be considered in a specific implementation. The requirement can be implemented within the architecture and is therefore fulfilled.

R.3.4 The architecture allows the patient to use a smartphone to manage the health data
The edge agent used to manage the data is platform-independent, making it possible to use a smartphone to manage health data. Therefore, this requirement is fulfilled.

R.3.5 A healthcare provider from another country can access a patient's data if the patient have configured the access accordingly. Special services or approvals from their home country are not required for the healthcare provider to do so
The architecture is designed to be country independent. The individual DWNs are accessible via the Internet with appropriate addressing. As long as the network traffic itself is not blocked by the countries, this requirement is met.

4.3.3.4 Overview

Table 4.18 presents an overview of the evaluation of the requirements in our architecture. None of the defined requirements are incompatible with the proposed architecture. However, five out of twenty-five requirements cannot be fully confirmed for compatibility. This is because an evaluation requires a specific implementation. Specifically, we need to measure processing times, evaluate how requests were handled, obtain values that can only be evaluated retrospectively, or inconveniently fulfill the requirement. However, it is important to note that meeting these five requirements is possible only in theory. The interpretation of the requirements is also a matter of consideration. It is important to emphasize R.3.2, as it can be fulfilled in a cumbersome way. This case is marked as only partially fulfilled, but it could also be marked as fulfilled or not fulfilled depending on the point of view. In conclusion, it can be stated that the architecture meets the specified requirements.

Data lifecycle		Trust, security and privacy		Human-related issues	
R.1.1	✓	**R.2.1**	✓	**R.3.1**	✓
R.1.2	✓	**R.2.2**	✓	**R.3.2**	?
R.1.3	✓	**R.2.3**	✓	**R.3.3**	✓
R.1.4	?	**R.2.4**	✓	**R.3.4**	✓
R.1.5	✓	**R.2.5**	?	**R.3.5**	✓
R.1.6	✓	**R.2.6**	✓		
R.1.7	✓	**R.2.7**	✓		
R.1.8	?	**R.2.8**	✓		
R.1.9	✓	**R.2.9**	✓		
R.1.10	✓	**R.2.10**	?		

✓= fulfills requirement, ? = requirement cannot be fully confirmed

Table 4.18: Overview of requirement evaluation

4.3.4 GDPR Evaluation

As the architecture must comply with European data protection guidelines, specifically the GDPR as defined in **RQ2**, it is important to take it into account. Section 2.3 analyzes the laws that must be observed for the corresponding technologies in our system. Based on the findings of Section 2.3, we will evaluate the compliance of our architecture with GDPR and identify areas for improvement. To evaluate the architecture, we examine each of the seven key principles of the GDPR.

Our provided architecture includes two controllers as defined in the GDPR, which are the healthcare provider in the role of the issuer and verifier, the blockchain component, and the DID resolver. The healthcare provider processes the patients' data, and the blockchain stores the public keys in the DID document, which are considered personal data in the context of the GDPR. In the context of SSI and the defined architecture, patients represent holders, while healthcare providers act as the issuers and verifiers. Specifically, the DWN and the edge agent of the healthcare provider serve as the issuer and verifier, respectively. The purpose of this evaluation is to assess the compatibility of the proposed architecture with the GDPR. Our responsibility is limited to the edge agent of the healthcare provider. Typically, the agent is responsible for reading and writing data, while any further processing is performed by the healthcare provider elsewhere. We are unable to evaluate this additional processing, as it falls under the responsibility of the individual healthcare providers. Additionally, we will not analyze the DID resolver separately since it is considered a blockchain node and any resulting issues are already included in the blockchain analysis.

4.3.4.1 Compatibility With Lawfulness, Fairness and Transparency

This section discusses the GDPR principle of lawful, fair, and transparent personal data processing. As outlined in Section 2.3.2.1, we analyze the compatibility of SSI with

this principle. We confirm that the lawfulness of SSI and its corresponding components is fulfilled as patients have given their consent to data processing. Naik and Jenkins (2020) describe fairness and transparency as the ability for users to monitor any potential mishandling of personal data and stay informed regarding the complete processing. Our architecture allows for tracking data access and corresponding logging mechanisms to record when data was accessed and by whom. However, the healthcare provider must inform the patient directly about how their data is processed. This can be done, for example, by sending messages with the relevant information to the patient. The use of a public blockchain as a VDR in the architecture allows for anyone to publicly track how data is processed, ensuring fairness and transparency. To store a DID document with public keys on the blockchain, the user must record it themselves through a transaction, indirectly giving consent to processing.

4.3.4.2 Compatibility With Purpose Limitation

This principle ensures that the architecture meets the purpose limitation requirement. The healthcare provider must access data in accordance with the principle and create appropriate documentation to comply with Article 15 of the GDPR. The article outlines the "Right of access by the data subject". It is the responsibility of the healthcare provider to demonstrate how the data is processed. In the provided architecture, blockchain is used to write, store, and provide public access to data through public keys. This principle is fulfilled as the purpose is clearly defined and not limited.

4.3.4.3 Compatibility With Data Minimisation

The defined architecture enables patients to create and revoke access policies for each data record, allowing them to decide which data healthcare providers can access and keeping it to a minimum. As outlined in Section 2.3.2.1, conflicts with GDPR arise in connection with the blockchain components due to the replication of data with multiple nodes and the append-only characteristic. Currently, blockchains are not GDPR-compliant as they store personal data, including the public keys in DID documents which are considered personal data under Article 4 (1) GDPR. Data protection authorities have issued official statements, as described in Section 2.3.2.1, arguing that blockchains can only retain data due to their structure. However, the General Data Protection Regulation (GDPR) does not currently address the unique challenges posed by blockchain technology, and therefore the established architecture may not be fully compatible with this principle.

4.3.4.4 Compatibility With Accuracy

As described in the previous principle, the characteristics of the blockchain result in conflicts with the GDPR. Specifically, Article 16 "Right to rectification" and Article 17 "Right to erasure" of the GDPR are not compatible with the functioning of a blockchain. In our proposed architecture, patients' personal data is managed on their own DWN. The healthcare provider only accesses the data on the patient's DWN and does not manage

it directly. If the healthcare providers do not additionally store the data on their own systems for processing purposes, this principle does not apply to them.

4.3.4.5 Compatibility With Storage Limitation

This principle is also incompatible with the GDPR due to the mentioned blockchain issues, as are the principles of "data minimization" and "accuracy". In addition, the healthcare provider must ensure that it is no longer possible to identify patients from the data once it has been processed. This is not influenced by the architecture and therefore it fulfills this principle.

4.3.4.6 Compatibility With Integrity and Confidentiality

In Section 4.3.2, we developed a threat model to assess the security of our architecture. This model helped us determine the security of personal data. Based on the threat model, it can be concluded that this principle is fulfilled in relation to our architecture. It is important to note that the healthcare provider must comply with security standards when processing data, regardless of the architecture's security. Furthermore, we must also evaluate security on the selected blockchain to evaluate the compliance with this principle.

4.3.4.7 Compatibility With Accountability

This principle requires the data processor to demonstrate compliance with the other principles. A healthcare provider can achieve this through suitable documentation mechanisms. However, as mentioned in Section 2.3.2.1, demonstrating accountability in relation to the blockchain is more challenging since every node of the blockchain may be considered to be processing data. Therefore, it is not possible to directly define an entity that proves accountability, and each entity would have to do so. In a public blockchain, as defined for this architecture, it is not possible to achieve this, unlike in a blockchain with a proof-of-authority consensus mechanism, for example. Therefore, compliance with this principle must be evaluated based on a specific implementation.

4.3.4.8 Overview

Table 4.19 provides an overview of the compatibility of our architecture's controller with GDPR regulations. As stated earlier in this section, the healthcare provider is responsible for complying with most of the GDPR's articles as they process the data. Therefore, our architecture ensures compliance with all principles for the healthcare provider as the verifier and issuer within this context.

In contrast, the blockchain component does not comply with three of the seven GDPR principles. This is because blockchain is a new technology that was not considered in the GDPR. Although the technology brings more privacy to some use cases (depending on the specific instance), its characteristics conflict with the GDPR. It is possible that the

GDPR may be amended in the future to allow for the use of such technologies without violating its principles.

	Healthcare provider	Blockchain
Compatibility with lawfulness, fairness and transparency	✓	✓
Compatibility with purpose limitation	✓	✓
Compatibility with data minimisation	✓	✗
Compatibility with accuracy	✓	✗
Compatibility with storage limitation	✓	✗
Compatibility with integrity and confidentiality	✓	∼
Compatibility with accountability	✓	∼

✓= fulfills GDPR principle, ∼ = GDPR principle cannot be fully confirmed/ has to be evaluated on concrete instance ✗= does not fulfill GDPR principle

Table 4.19: Overview of GDPR evaluation

In Section 2.3, it was determined that any hosting provider that hosts the DWNs can only be considered a processor. However, this is only applicable if the entity using the hosting service is defined as a controller. This is the case for the healthcare provider's DWN, as the healthcare provider is a controller within the meaning of the GDPR. As stated in Section 2.3, the controller must verify that the hosting provider meets the security and confidentiality requirements before using them. If the requirements are met, the hosting provider can be considered a GDPR-compliant option.

4.3.5 Evaluation Interviews

In the previous sections, we evaluated the proposed architecture based on feasibility, security and our specified requirements. However, the general functionality and processes of the architecture has not yet been discussed and evaluated. For this reason, we conducted five expert interviews, using a pre-established guideline. The experts evaluated our architecture based on their opinions and experience, highlighting its strengths and weaknesses. This includes the topics of functionality, performance, security, and privacy. In addition, the experts compare our architecture with existing and future architectures. The process of conducting the interviews is similar to the interviews for the requirements, as described in Section 4.1.2. As with the first interview, the interview process consists

of three phases: (i) developing the interview guidelines, (ii) conducting the interview, and (iii) evaluating the interview.

4.3.5.1 Creation of the Interview Guideline and Interview Execution

The purpose of this interview is to assess the strengths and weaknesses of various aspects of the architecture. To achieve this, we have categorized the questions into four areas: (i) functionality, (ii) performance, (iii) security and privacy, and (iv) individual evaluation. The categories (i) and (ii) serve to find out whether the proposed architecture satisfies the required functionality and performance criteria from the experts' perspective and how it compares to current systems and concepts. The questions about the architecture's security help identify any security vulnerabilities and components not already identified in the threat model. Additionally, we ask questions to clarify the privacy of the patients in the presented architecture. The final category of questions provides a conclusive evaluation of the architecture.

We interviewed five experts who possess expertise in eHealth architectures. Table 4.12 in the Requirement Engineering chapter provides an overview of the experts. Three of the experts were previously interviewed in the first round. Each interview in this round lasted one hour and followed a semi-structured format based on the provided guidelines. The interviews with EX1, EX2, and EX4 were conducted remotely, while the interviews with EX3 and EX5 were conducted on-site. All interviews were audio recorded and the recordings were deleted after transcription.

4.3.5.2 Evaluation of the Interview

The evaluation interviews followed the methodology of Meuser and Nagel [15], which was also used in the first round of interviews for the requirements. The methodology is described in detail in Section 4.1.2. The audio recorded interviews were transcribed accurately and completely in the first step, following the methodology's instructions. In the paraphrase phase, we paraphrased the content of the interview so that the complexity was simplified and only the experts' answers remained. In the next phase we marked the individual passages in the interview texts with labels/tags. These tags represent the categories or strengths and weaknesses of the architecture as highlighted by the experts. We utilized the following 22 tags: (i) Key management, (ii) Access management, (iii) Re-encryption, (iv) Categories, (v) Meta data , (vi) Standardized data formats , (vii) Performance, (viii) Complexity, (ix) Support, (x) Security, (xi) Privacy, (xii) Personal assessment, (xiii) Key recovery, (xiv) Additional use cases, (xv) Feasibility, (xvi) Availability, (xvii) Encryption, (xviii) Malicious content, (xix) Emergency access, (xx) Legal, (xxi) Functionality, (xxii) Usability,

In the following step, we analyzed the connections and distinctions among the five interviews. This led us to reduce the number of tags to 14 and label similar passages from the different interviews with standardized tags. Like the creation of the requirements, we summarized the content and outcomes from the various interviews to provide clear

explanations regarding the evaluation of the experts. Once again, we omitted the last step of "theoretical generalization", as it is not required for the evaluation of the interviews.

4.3.5.3 Results of the Expert Assessment

In this section, we use the results of the interviews to evaluate our architecture from the experts' perspective. We use the same categorization int following subsections as in the interview guideline.

Functionality
All five experts agree that the specified use cases in Section 4.1.1 can be fulfilled by the architecture's functionality. The architecture also covers the functionality provided by existing systems in this area. However, four out of five experts suggest that additional use cases are necessary for a practical architecture. EX2 and EX3 argue that emergency access to the data is essential for medical personnel. EX2 proposes a solution where the user sets up a permission, and a trustee manages the corresponding private keys. EX2 and EX5 highlight the use case of reporting, where research institutes can access patients' health records to use medical data for research. This use case is not possible in the proposed architecture without additional actions by the patients. EX1 identifies missing elements in the data structure, specifically the absence of metadata and standardized formats in the current model. However, he does not consider this to be contradictory to the specified architecture.

One relevant issue in practice is the topic of key recovery. According to experts EX2, EX3, and EX4, managing keys themselves is too complex for the average user, and therefore additional mechanisms are needed. EX2 and EX3 suggest current mechanisms as a solution, in which two parties each manage one half of the key, and the user can merge the holdings of both parties by confirming the entity. However, this approach contradicts the goal of the architecture, which is to be self-sovereign and privacy-preserving, as it requires trusting another party to manage the keys. The key recovery mechanism remains an open problem for the practical use of the architecture. While there are various theoretical solutions, such as offline backups and multi-device recoveries, these are often too complex for the average user.

Performance
In this category, four out of five experts describe the performance of this architecture as like existing systems. Only EX3 refrains from making a general statement, citing the complexity of the performance issue and the difficulty of evaluating several factors in theory. EX4 emphasizes that the availability of this architecture is even better than that of existing systems due to its decentralized nature. EX3 and EX5 have identified the edge agents running on smartphones as a performance weakness of the architecture. They refer to the decryption of health records on the smartphone. Since the architecture specifies that the data is end-to-end encrypted and the DWNs have no way of decrypting the data, decryption must be performed directly on the device. This results in performance

losses. As an alternative, EX3 suggests that the DWN itself has access to the data and performs decryption on request. This indicates that the data is decrypted before it is sent to the client, allowing for more processing power in the background.

Security & Privacy

Three of the experts are satisfied with the security of the architecture, but EX3 and EX4 have expressed criticism. EX3 criticizes the scenario of re-encryption in the event that the symmetric key used to encrypt the data needs to be changed. The encrypted data must be downloaded, decrypted, encrypted with the new key, and uploaded again. EX3's criticism relates to inefficiency, among other things. Additionally, EX3 emphasizes the need for a solution to prevent patients from introducing malicious content that could infect healthcare providers' systems. Currently, there is no end-to-end encryption in place, so this check is performed centrally when data is uploaded. As outlined in Section 4.3.2, one countermeasure is to run a virus scanner on the healthcare providers' systems. According to EX4, the DID resolver is considered the weakest component of the system since it reintroduces centralization. Therefore, it is crucial to ensure its security or have multiple resolvers in practical use.

Four out of five experts rated the architecture's privacy positively. The only exception was EX3, who criticized the privacy aspect due to the blockchain. EX3 argued that an implementation error could result in information being transferred to the blockchain that cannot be deleted. While this may be a good solution for technically savvy users in terms of security and privacy, it raises security concerns for standard users. Four of the experts also highlight the importance of end-to-end encryption. Only EX5 does not mention this explicitly. EX1, EX2, and EX3 are working on a project related to the German health record application. They report that the next version of this system no longer has end-to-end encryption, which negatively impacts privacy. The proposed architecture addresses this issue and offers more privacy in these areas. In his statement, EX4 refers to the Austrian health record, which also lacks end-to-end encryption. He views this property as a positive aspect of the proposed architecture. EX5 rates privacy positively and believes that all the criteria are met. He describes the architecture as a "dream landscape for many data protectionists".

Personal Assessment

In the following the personal assessment of each of the experts is stated individually.

- **EX1**: EX1 highlights the significance of end-to-end encryption, as the current trend is moving away from data protection. He rates the prototype positively, but has some relevant comments for practical use, such as including meta data, supporting general file formats, and implementing categorization. However, he clarifies that these suggestions do not conflict with the proposed architecture. The biggest hurdles in practice are the complexity for the average user and the lack of support from healthcare providers and health insurance companies.

- **EX2**: The expert EX2 is critical of blockchain and prefers a solution with centralized components to address the general issues of a blockchain, such as determining who should be a node and how to get entities to host it. Similar to EX1, he views the architecture as too complex for practical use, as it will be difficult for the general user community to understand and there will be only a few providers who will centralize the concept.
- **EX3**: EX3 views the architecture as a interesting alternative to existing systems. He believes that such architectures provide new impulses and perspectives. Similar to EX1, he views end-to-end encryption positively. However, like EX1 and EX2, they believe that the prototype is too complex for most people to use in practice. Above all, he emphasizes the individual user's responsibility regarding key recovery.
- **EX4**: EX4 suggests that the proposed prototype offers a distinct approach from the current federated systems and could serve as a potential alternative in the future. However, he notes that practical use must consider legal aspects, specifically the European NIS Directive and other country-specific directives. He mentions that the proposed architecture does not align with the current legal situation. This is because it specifies who is authorized to operate such availability-critical infrastructure. This area is subject to strict regulation and currently requires a federated approach, which makes the self-sovereign approach impractical in the current legal context.
- **EX5**: EX5 views this architecture as exciting and believes it should be discussed at the conference. Like other experts, EX5 also believes the prototype is too complex for most users. An app with good usability for administration would be necessary. EX5 sees this architecture as realistic in 5 to 10 years.

CHAPTER 5

Conclusion and Future Work

This chapter provides a summary of the thesis in Section 5.1, presents the conclusion in Section 5.2, and suggests potential future work in Section 5.3.

5.1 Summary

This thesis addresses the issue that medical data is primarily managed by companies or government institutions rather than patients themselves. Medical data is highly sensitive and should only be accessible to authorized individuals. The proposed architecture offers a solution to this problem. Patients can manage their own data by utilizing the principles of self-sovereign identity and Decentralized Web Nodes. Additionally, we have considered the legal implications of GDPR when designing our architecture. The architecture is based on three expert interviews, from which we have developed requirements and use cases that represent common interactions between patients and healthcare providers in the healthcare sector. The presented architecture consists of six core components that facilitate the management of medical records. To assess the results, we conducted a proof-of-concept, developed a threat model, evaluated the requirements based on the architecture, analyzed the compatibility of the architecture with the GDPR, and obtained expert opinions through interviews.

The final architecture presents a new approach that prioritizes data and patients, allowing them to regain control over their own data. As noted by experts, there are still unresolved issues regarding personal user responsibility, for which there are currently limited technical solutions. Additionally, this concept is not compatible with current European data protection standards outlined in the GDPR, due to the way it was drafted and its intended purposes. In summary, this architecture presents a new and privacy-preserving approach that could be implemented in healthcare in the future and extended to other fields.

5.2 Discussion of Research Questions

In the following we discuss the outcomes of this theses based on the research questions.

RQ1 How might self-sovereign identity be used in the context of healthcare?

Section 2.2.2 discusses the use of SSI in eHealth architectures, presenting three approaches: using SSI alone, using SSI with distributed file storage, and using SSI with Decentralized Web Nodes. The last approach is the most suitable for eHealth systems due to the identified requirements. It combines SSI and DWN technology as a message relay mechanism and data storage. The architecture's most important component is the decentralized identifiers used for addressing. The provision of public keys and serviceEndpoint properties enables secure and easy-to-establish communication between the entities of the ecosystem. Verifiable credentials, listed as further essential components in the SSI concept, are of lesser importance for the architecture. Although message objects of the architecture could replace them, selective disclosure enables precise data selection. The architecture containing healthcare providers as issuers and verifiers, and patients as holders, covers the trust triangle of the SSI concept. Section 4.2 provides a detailed description of the necessary components for a functioning healthcare architecture for medical records based on SSI.

RQ2 What laws and regulations need to be considered when storing and providing eHealth data related to self-sovereign identity in order to create an eHealth infrastructure that complies to the European laws of data protection?

To clarify compliance with European data protection laws, we have examined the GDPR. It defines seven key principles that protect the data of users in Europe. Although the GDPR has individual laws, most of them pertain to these key principles. The main issue with GDPR and architecture lies in the fact that GDPR is primarily designed to protect natural persons in centralized and federated architectures when it comes to processing personal data [38]. However, the concept of SSI does not conform to this model, which raises questions as there are no concrete definitions or laws for the roles and components of SSI. Our analysis of the proposed architecture shows that it is compatible with GDPR, with one exception: the blockchain. The characteristics of a blockchain, such as data replication with multiple nodes and append-only, contradict the principles and laws of GDPR. Although official statements [46] argue that the structure of blockchain does not violate GDPR, it is important to note that a blockchain used for a decentralized public key infrastructure without adaptation of GDPR would be considered a violation of GDPR.

RQ3 How to construct a privacy preserving eHealth architecture for interconnecting healthcare facilities based on self-sovereign identity, considering theoretical concepts and implementations?

In this thesis, we identified a lack of general requirements and framework conditions for eHealth systems. In practice, laws form the basis for such architectures, with particular mention of the NIS/NIS2 Directive or country-specific laws, according to EX4 mentioned in Section 4.3.5. For the thesis on the topic of SSI, we designed the prototype based on requirements gathered from expert interviews. The current legal situation does not permit SSI for eHealth architectures, according to EX4. Furthermore, a healthcare architecture for medical records must encompass practice-relevant use cases, as defined in Section 4.1.2. We have begun to specify the individual components and their process flows based on these requirements and use cases, as well as the concepts outlined in **RQ1**. To ensure functionality, a proof-of-concept is necessary in the next step. The final step in constructing a privacy-preserving eHealth architecture is evaluation. This involves checking the architecture's requirements, security, and compatibility with data protection. This process can be found in Section 4.1, 4.2, and 4.

5.3 Future Work

Our proposed architecture is limited to 15 use cases, which represent the basic needs of healthcare providers and patients regarding medical records. To make this architecture practical, we need to consider additional use cases, such as access to emergency data or guardianship of one person for another patient. Extending the existing architecture with such use cases is a promising research question for future work.

Additionally, two of the specified requirements, namely the key recovery requirement **R.1.10** and the data re-encryption requirement **R.3.2**, cannot be efficiently addressed. Since the concept of SSI is based on individual user control and responsibility, there is no possibility of another party assisting with key recovery. This presents a significant issue, as the user will lose access to all of their data if the keys are lost. A question for future work is what options exist for key recovery to prevent complete loss while maintaining security. The issue of data re-encryption was also not efficiently resolved if the symmetric key was compromised, and the data needed to be re-encrypted. Although several symmetric key proxy re-encryption concepts exist, this text will address the question of which efficient mechanisms can be used to re-encrypt symmetrically encrypted data without decrypting the data beforehand.

One criticism, particularly in the interviews, was the complexity of this architecture. This raises the question of how to ensure that even non-technical users can operate the architecture without risking the security and privacy of their data.

Finally, healthcare architectures are often subject to the framework conditions set by other laws. Therefore, it would be worthwhile to consider not only the General Data Protection Regulation (GDPR) but also other relevant laws and directives, such as the NIS/NIS2 directive, when designing practice-relevant architectures.

Appendix

Requirements Expert Interview Guide

1. Personal Information
    a) How often and in what roles have you been involved in the design or implementation process of a health care system?
    b) How many years of experience do have in your role?
2. Functional Requirements
    a) From your perspective and experience, what are the most important functions that a digital healthcare system must cover?
    b) What do you think is important for patients?
    c) What do you think is important for the healthcare provider?
3. Socio-demographic data
    a) What percentage of people could use a smartphone application with a good UI to manage their health data?
    b) What is the required geographical infrastructure for a digital healthcare system?
    c) What technical infrastructure is required to build a digital healthcare system?
4. Non-functional requirements
    a) In the context of eHealth, the design of such a system requires special non-functional or performance requirements. What are these requirements?
    b) What do the performance criteria of response time, throughput, scalability, or availability look like in relation to eHealth systems?
5. Security
    a) What are the most essential security requirements in the context of eHealth?

b) What are the most vulnerable parts of such an architecture?

c) Privacy is an important aspect of my architecture. When should a software system be considered privacy-preserving? What is your definition of privacy-preserving in relation to such systems?

6. Final question

a) What factors should be considered when building a decentralized architecture?

Evaluation Expert Interview Guide

1. Personal Information

a) How often and in what roles have you been involved in the design or implementation process of a health care system?

b) How many years of experience do have in your role?

2. Functional Requirements

a) Do you believe that the proposed architecture can cover all the necessary functionalities required in today's healthcare systems, given your experience?

b) Are there areas for improvement or potential problems? If not, why?

3. Non-Functional Requirements

a) Are there any performance criteria, such as response time, throughput, scalability, availability, or backup, that you believe cannot be achieved with this architecture?

b) What criteria are improved or worsened compared to current systems?

4. Security

a) What security vulnerabilities do you notice?

b) What are the most vulnerable parts of such an architecture?

c) How does the security of this architecture compare to those you have experience with?

d) Does the architecture meet the privacy requirements for such a system? If the architecture does not meet the requirements, please explain why.

e) Does the architecture provide greater privacy than existing systems?

5. Final question

a) What is your final opinion on architecture?

List of Figures

List of Tables

Bibliography

[1] *Agents | TBD*. 2024. URL: https://developer.tbd.website/docs/web5/learn/agents (visited on 04/05/2024).

[2] Christopher Allen. *The Path to Self-Sovereign Identity*. 2016. URL: https://www.lifewithalacrity.com/article/the-path-to-self-soverereign-identity/ (visited on 04/05/2024).

[3] Dan Armijo, Cheryl Mcdonnell, and Kristen Werner. *Electronic Health Record Usability Evaluation and Use Case Framework HEALTH IT*. 2009. URL: https://digital.ahrq.gov/sites/default/files/docs/citation/09-10-0091-1-EF.pdf (visited on 04/05/2024).

[4] Oscar Avellaneda et al. "Decentralized Identity: Where Did It Come From and Where Is It Going?" In: *IEEE Communications Standards Magazine* 3.4 (2019), pp. 10–13. DOI: 10.1109/MCOMSTD.2019.9031542.

[5] Asaph Azaria et al. "MedRec: Using Blockchain for Medical Data Access and Permission Management". In: *2016 2nd International Conference on Open and Big Data (OBD)*. 2016, pp. 25–30. DOI: 10.1109/OBD.2016.11.

[6] László Babai. "Groups, Graphs, Algorithms: The Graph Isomorphism Problem". In: *Proceedings of the International Congress of Mathematicians (ICM 2018)*, pp. 3319–3336. DOI: 10.1142/9789813272880_0183.

[7] Pinky Bai et al. "Self-Sovereignty Identity Management Model for Smart Healthcare System". In: *Sensors* 22.13 (2022). ISSN: 1424-8220. DOI: 10.3390/s22134714.

[8] Kathy Baxter, Catherine Courage, and Kelly Caine. "Choosing a User Experience Research Activity". In: *Understanding your Users* (Jan. 2015), pp. 96–112. DOI: 10.1016/B978-0-12-800232-2.00005-5.

[9] Rafael Belchior et al. "A Survey on Blockchain Interoperability: Past, Present, and Future Trends". In: *ACM Comput. Surv.* 54.8 (2021). ISSN: 0360-0300. DOI: 10.1145/3471140.

[10] Rafael Belchior et al. "SSIBAC: Self-Sovereign Identity Based Access Control". In: *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. 2020, pp. 1935–1943. DOI: 10.1109/TrustCom50675.2020.00264.

[11] Juan Benet. *IPFS - Content Addressed, Versioned, P2P File System (Draft 3)*. 2014.

[12] Matt Blaze, Gerrit Bleumer, and Martin Strauss. "Divertible protocols and atomic proxy cryptography". In: *Advances in Cryptology — EUROCRYPT'98*. Ed. by Kaisa Nyberg. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 127–144. ISBN: 978-3-540-69795-4.

[13] Matt Blaze, Gerrit Bleumer, and Martin Strauss. "Divertible protocols and atomic proxy cryptography". In: *Advances in Cryptology — EUROCRYPT'98*. Ed. by Kaisa Nyberg. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 127–144. ISBN: 978-3-540-69795-4.

[14] European Data Protection Board. *Guidelines 07/2020 on the concepts of controller and processor in the GDPR*. 2020. URL: https://edpb.europa.eu/our-work-tools/documents/public-consultations/2020/guidelines-072020-concepts-controller-and_en (visited on 04/05/2024).

[15] Alexander Bogner and Wolfgang Menz. "Das theoriegenerierende Experteninterview". In: *Das Experteninterview: Theorie, Methode, Anwendung*. Ed. by Alexander Bogner, Beate Littig, and Wolfgang Menz. Wiesbaden: VS Verlag für Sozialwissenschaften, 2002, pp. 71–93. ISBN: 978-3-322-93270-9. DOI: 10.1007/978-3-322-93270-9.

[16] Mohammed Amine Bouras et al. "Distributed Ledger Technology for eHealth Identity Privacy: State of The Art and Future Perspective". In: *Sensors* 20.2 (2020). ISSN: 1424-8220. DOI: 10.3390/s20020483.

[17] Pflege und Konsumentenschutz Bundesministerium Soziales Gesundheit. *eHealth*. 2019. URL: https://www.sozialministerium.at/Themen/Gesundheit/eHealth.html (visited on 04/05/2024).

[18] Jorge Calvillo-Arbizu, Isabel Román-Martínez, and Javier Reina-Tosina. "Internet of things in health: Requirements, issues, and gaps". In: *Computer Methods and Programs in Biomedicine* 208 (2021), p. 106231. ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2021.106231.

[19] Jeremy J. Carroll. "Signing RDF Graphs". In: *The Semantic Web - ISWC 2003*. Ed. by Dieter Fensel, Katia Sycara, and John Mylopoulos. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 369–384. ISBN: 978-3-540-39718-2.

[20] Andrés Chomczyk Penedo. "Self-sovereign identity systems and European data protection regulations: an analysis of roles and responsibilities". In: *Open Identity Summit 2021*. Ed. by Heiko Roßnagel, Christian H. Schunck, and Sebastian Mödersheim. Vol. P-312. Lecture Notes in Informatics (LNI) - Proceedings. Gesellschaft für Informatik, 2021, pp. 95–106. ISBN: 978-3-88579-706-7.

[21] Alistair Cockburn. "Goals and use cases". In: *Journal of object-oriented programming* 10.5 (1997), pp. 35–40.

[22] CORDIS European Commission. *About SSI eIDAS Bridge | Joinup*. 2024. URL: https://joinup.ec.europa.eu/collection/ssi-eidas-bridge/about (visited on 04/05/2024).

[23] European Commission. *eHealth : Digital health and care*. URL: https://health.ec.europa.eu/ehealth-digital-health-and-care,urldate={2024-04-05},year={2024}.

[24] European Commission. *Protection and privacy of hospital and health infrastructures with smart Cyber security and cyber threat toolkit for data and people*. 2022. URL: https://cordis.europa.eu/project/id/826293/de (visited on 04/05/2024).

[25] D.L. Cool and A.D. Keromytis. "Conversion and proxy functions for symmetric key ciphers". In: *International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II*. Vol. 1. 2005, 662–667 Vol. 1. DOI: 10.1109/ITCC.2005.115.

[26] DAIC. *Cardiac MRI Advances | DAIC*. 2016. URL: https://www.dicardiology.com/article/cardiac-mri-advances (visited on 04/05/2024).

[27] Omar Dib and Khalifa Toumi. "Decentralized identity systems: Architecture, challenges, solutions and future directions". In: *Annals of Emerging Technologies in Computing* 4 (5 2020), pp. 19–40. ISSN: 2516029X. DOI: 10.33166/AETIC.2020.05.002.

[28] *DIDComm*. 2024. URL: https://didcomm.org (visited on 04/05/2024).

[29] DIF. *Decentralized Web Node*. 2024. URL: https://identity.foundation/decentralized-web-node/spec (visited on 04/05/2024).

[30] DIF. *Sidetree Protocol*. 2024. URL: https://identity.foundation/sidetree/spec/ (visited on 04/05/2024).

[31] *Digital Imaging and Communication in Medicine*. 2024. URL: https://www.dicomstandard.org/ (visited on 04/05/2024).

[32] Pamela Dingle. *ION – Booting up the network - Microsoft Community Hub*. 2020. URL: https://techcommunity.microsoft.com/t5/security-compliance-and-identity/ion-booting-up-the-network/ba-p/1441552 (visited on 04/05/2024).

[33] Victoria Drake. *Threat Modeling | OWASP Foundation*. 2024. URL: https://owasp.org/www-community/Threat_Modeling (visited on 04/05/2024).

[34] Victoria Drake. *Threat Modeling Process | OWASP Foundation*. 2024. URL: https://owasp.org/www-community/Threat_Modeling_Process (visited on 04/05/2024).

[35] Thomas W. Edgar and David O. Manz. "Exploratory Study". In: *Research Methods for Cyber Security* (Jan. 2017), pp. 95–130. DOI: 10.1016/B978-0-12-805349-2.00004-2.

[36] Nabil El Ioini and Claus Pahl. "A Review of Distributed Ledger Technologies". In: *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*. Ed. by Hervé Panetto et al. Cham: Springer International Publishing, 2018, pp. 277–288. ISBN: 978-3-030-02671-4.

[37] Christian Esposito et al. "Blockchain: A Panacea for Healthcare Cloud-Based Data Security and Privacy?" In: *IEEE Cloud Computing* 5.1 (2018), pp. 31–37. DOI: 10.1109/MCC.2018.011791712.

[38] European Parliament and Council of the European Union. *Regulation (EU) 2016/679 of the European Parliament and of the Council*. of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). May 4, 2016. URL: https://data.europa.eu/eli/reg/2016/679/oj (visited on 08/10/2023).

[39] Kai Fan et al. "MedBlock: Efficient and Secure Medical Data Sharing Via Blockchain". In: *Journal of Medical Systems* 42 (8 Aug. 2018), pp. 1–11. ISSN: 1573689X. DOI: 10.1007/S10916-018-0993-7/FIGURES/7.

[40] Daniel Fett, Kristina Yasuda, and Brian Campbell. *Selective Disclosure for JWTs (SD-JWT)*. Internet-Draft draft-ietf-oauth-selective-disclosure-jwt-07. Work in Progress. Internet Engineering Task Force, Dec. 2023. 82 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt/07/ (visited on 04/05/2024).

[41] Ethereum Foundation. *Ethereum website*. 2024. URL: https://ethereum.org/en/ (visited on 04/05/2024).

[42] Hyperledger Foundation. *Hyperledger Indy website*. 2024. URL: https://www.hyperledger.org/projects/hyperledger-indy (visited on 04/05/2024).

[43] Ulrich M Gassner. "Blockchain in EU e-health-blocked by the barrier of data protection?" In: *Compliance Elliance Journal* 4.2 (2018).

[44] gematik. *Telematikinfrastruktur*. 2024. URL: https://www.gematik.de/telematikinfrastruktur (visited on 04/05/2024).

[45] gematik. *Übergreifende Spezifikation Performance und Mengengerüst TI-Plattform V.2.30.1*. 2023. URL: https://fachportal.gematik.de/fachportal-import/files/gemSpec_Perf_V2.30.1.pdf (visited on 04/05/2024).

[46] Alexandra Giannopoulou. "Data Protection Compliance Challenges for Self-sovereign Identity". In: *Blockchain and Applications*. Ed. by Javier Prieto et al. Cham: Springer International Publishing, 2020, pp. 91–100. ISBN: 978-3-030-52535-4.

[47] Remo Glauser. "Self-Sovereign Identities in Cardossier". MA thesis. ETH, 2019. URL: https://pub.tik.ee.ethz.ch/students/2018-HS/MA-2018-34.pdf (visited on 04/05/2024).

[48] Andreas Grüner et al. “Analyzing and comparing the security of self-sovereign identity management systems through threat modeling”. In: *International Journal of Information Security* 22 (5 Oct. 2023), pp. 1231–1248. ISSN: 16155270. DOI: 10.1007/S10207-023-00688-W/FIGURES/10.

[49] Huaqun Guo and Xingjie Yu. “A survey on blockchain technology and its security”. In: *Blockchain: Research and Applications* 3.2 (2022), p. 100067. ISSN: 2096-7209. DOI: https://doi.org/10.1016/j.bcra.2022.100067.

[50] Harry Halpin. “Vision: A Critique of Immunity Passports and W3C Decentralized Identifiers”. In: *Security Standardisation Research*. Ed. by Thyla van der Merwe, Chris Mitchell, and Maryam Mehrnezhad. Cham: Springer International Publishing, 2020, pp. 148–168. ISBN: 978-3-030-64357-7.

[51] Marit Hansen, Ari Schwartz, and Alissa Cooper. “Privacy and Identity Management”. In: *IEEE Security Privacy* 6.2 (2008), pp. 38–45. DOI: 10.1109/MSP.2008.41.

[52] Daniel Toshio Harrell et al. “Technical Design and Development of a Self-Sovereign Identity Management Platform for Patient-Centric Healthcare Using Blockchain Technology”. In: *Blockchain in Healthcare Today* 5.S1 (Apr. 2022). ISSN: 2573-8240. DOI: 10.30953/bhty.v5.196.

[53] *Health Level Seven International - Homepage | HL7 International*. 2024. URL: https://www.hl7.org/ (visited on 04/05/2024).

[54] GE HealthCare. *Viosworks Imaging Application for SIGNA Works*. 2024. URL: https://www.gehealthcare.com/products/magnetic-resonance-imaging/signa-works/viosworks (visited on 04/05/2024).

[55] Tsipi Heart, Ofir Ben-Assuli, and Itamar Shabtai. “A review of PHR, EMR and EHR integration: A more personalized healthcare and public health policy”. In: *Health Policy and Technology* 6.1 (2017), pp. 20–25. ISSN: 2211-8837. DOI: https://doi.org/10.1016/j.hlpt.2016.08.002.

[56] Bundesministerium des Innern und für Heimat. *SMART-Regel / SMART-Methode*. 2021. URL: https://www.orghandbuch.de/Webs/OHB/DE/OrganisationshandbuchNEU/4_MethodenUndTechniken/Methoden_A_bis_Z/SMART_Regel_Methode/SMART_Regel_Methode_node.html (visited on 04/05/2024).

[57] *Home - Keri.one*. 2024. URL: https://keri.one/ (visited on 04/05/2024).

[58] *Home Page | TBD*. 2024. URL: https://www.tbd.website/ (visited on 04/05/2024).

[59] Bahar Houtan, Abdelhakim Senhaji Hafid, and Dimitrios Makrakis. “A Survey on Blockchain-Based Self-Sovereign Patient Identity in Healthcare”. In: *IEEE Access* 8 (2020), pp. 90478–90494. DOI: 10.1109/ACCESS.2020.2994090.

[60] Haiping Huang et al. “A blockchain-based scheme for privacy-preserving and secure sharing of medical data”. In: *Computers Security* 99 (2020), p. 102010. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2020.102010.

[61] *IPFS website*. 2024. URL: https://ipfs.tech/ (visited on 04/05/2024).

[62] ISO. *ISO 13606-1:2019 - Health informatics — Electronic health record communication — Part 1: Reference model*. 2024. URL: https://www.iso.org/standard/67868.html (visited on 04/05/2024).

[63] Pramod David Jacob. “Chapter 3 - Management of patient healthcare information: Healthcare-related information flow, access, and availability”. In: *Fundamentals of Telemedicine and Telehealth*. Ed. by Shashi Gogia. Academic Press, 2020, pp. 35–57. ISBN: 978-0-12-814309-4. DOI: https://doi.org/10.1016/B978-0-12-814309-4.00003-3.

[64] Tibor Jager, Kenneth G Paterson, and Juraj Somorovsky. “One Bad Apple: Backwards Compatibility Attacks on State-of-the-Art Cryptography.” In: *NDSS*. 2013.

[65] Ibrahim Tariq Javed et al. “Health-ID: A Blockchain-Based Decentralized Identity Management for Remote Healthcare”. In: *Healthcare* 9.6 (2021). ISSN: 2227-9032. DOI: 10.3390/healthcare9060712.

[66] Michael B. Jones. *JSON Web Algorithms (JWA)*. RFC 7518. May 2015. DOI: 10.17487/RFC7518. URL: https://www.rfc-editor.org/info/rfc7518 (visited on 04/05/2024).

[67] Michael B. Jones, John Bradley, and Nat Sakimura. *JSON Web Token (JWT)*. RFC 7519. May 2015. DOI: 10.17487/RFC7519. URL: https://www.rfc-editor.org/info/rfc7519 (visited on 04/05/2024).

[68] Michael B. Jones and Joe Hildebrand. *JSON Web Encryption (JWE)*. RFC 7516. May 2015. DOI: 10.17487/RFC7516. URL: https://www.rfc-editor.org/info/rfc7516 (visited on 04/05/2024).

[69] Anjum Khurshid et al. “Designing and testing a blockchain application for patient identity management in healthcare”. In: *JAMIA Open* 4.3 (Feb. 2021), ooaa073. ISSN: 2574-2531. DOI: 10.1093/jamiaopen/ooaa073.

[70] Loren Kohnfelder and Praerit Garg. “The threats to our products”. In: *Microsoft Interface, Microsoft Corporation* 33 (1999).

[71] Galia Kondova and Jörn Erbguth. “Self-sovereign identity on public blockchains and the GDPR”. In: *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. SAC ’20. Brno, Czech Republic: Association for Computing Machinery, 2020, 342–345. ISBN: 9781450368667. DOI: 10.1145/3341105.3374066.

[72] Adam Langley, Mike Hamburg, and Sean Turner. *Elliptic Curves for Security*. RFC 7748. [Online; accessed 02-November-2023]. Jan. 2016. DOI: 10.17487/RFC7748. URL: https://www.rfc-editor.org/info/rfc7748.

[73] Xueping Liang et al. “Integrating blockchain for data sharing and collaboration in mobile healthcare applications”. In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2017, pp. 1–5. DOI: 10.1109/PIMRC.2017.8292361.

[74] MediBloc Limited. *Own your health data. It's rightfully yours.* 2024. URL: https://medibloc.com (visited on 04/05/2024).

[75] Yi Lu, Willi Meier, and Serge Vaudenay. “The Conditional Correlation Attack: A Practical Attack on Bluetooth Encryption”. In: *Advances in Cryptology – CRYPTO 2005*. Ed. by Victor Shoup. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 97–117. ISBN: 978-3-540-31870-5.

[76] Philipp Mayring and Thomas Fenzl. “Qualitative Inhaltsanalyse”. In: *Handbuch Methoden der empirischen Sozialforschung*. Ed. by Nina Baur and Jörg Blasius. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pp. 633–648. ISBN: 978-3-658-21308-4. DOI: 10.1007/978-3-658-21308-4_42.

[77] Alexander Mühle et al. “A survey on essential components of a self-sovereign identity”. In: *Computer Science Review* 30 (2018), pp. 80–86. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2018.10.002.

[78] Nitin Naik and Paul Jenkins. “Your Identity is Yours: Take Back Control of Your Identity Using GDPR Compatible Self-Sovereign Identity”. In: *2020 7th International Conference on Behavioural and Social Computing (BESC)*. 2020, pp. 1–6. DOI: 10.1109/BESC51023.2020.9348298.

[79] Nitin Naik et al. “An evaluation of potential attack surfaces based on attack tree modelling and risk matrix applied to self-sovereign identity”. In: *Computers Security* 120 (2022), p. 102808. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2022.102808.

[80] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008. URL: https://bitcoin.org/bitcoin.pdf (visited on 04/05/2024).

[81] Dinh C. Nguyen et al. “Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems”. In: *IEEE Access* 7 (2019), pp. 66792–66806. DOI: 10.1109/ACCESS.2019.2917555.

[82] NIH. *Personal Health Record*.

[83] *openEHR*. 2024. URL: https://www.openehr.org/ (visited on 04/05/2024).

[84] Kai Petersen et al. “Systematic Mapping Studies in Software Engineering”. In: *12th International Conference on Evaluation and Assessment in Software Engineering, EASE 2008* (June 2008). DOI: 10.14236/EWIC/EASE2008.8.

[85] Daniela Pöhn and Wolfgang Hommel. “An overview of limitations and approaches in identity management”. In: ARES '20. Virtual Event, Ireland: Association for Computing Machinery, 2020. ISBN: 9781450388337. DOI: 10.1145/3407023.3407026.

[86] Alex Preukschat and Dummond Reed. “Self-Sovereign Identity”. In: 1st ed. Manning, 2021, pp. 3–12. ISBN: 978-1617296598.

[87] Alex Preukschat and Dummond Reed. “Self-Sovereign Identity”. In: 1st ed. Manning, 2021, pp. 21–38. ISBN: 978-1617296598.

[88] Alex Preukschat and Dummond Reed. “Self-Sovereign Identity”. In: 1st ed. Manning, 2021. ISBN: 978-1617296598.

[89] Alex Preukschat and Dummond Reed. “Self-Sovereign Identity”. In: 1st ed. Manning, 2021, pp. 33–34. ISBN: 978-1617296598.

[90] Alex Preukschat and Dummond Reed. “Self-Sovereign Identity”. In: 1st ed. Manning, 2021, pp. 220–247. ISBN: 978-1617296598.

[91] Alex Preukschat and Dummond Reed. “Self-Sovereign Identity”. In: 1st ed. Manning, 2021, pp. 189–219. ISBN: 978-1617296598.

[92] Alex Preukschat and Dummond Reed. “Self-Sovereign Identity”. In: 1st ed. Manning, 2021, pp. 94–101. ISBN: 978-1617296598.

[93] Alex Preukschat and Dummond Reed. “Self-Sovereign Identity”. In: 1st ed. Manning, 2021, pp. 157–168. ISBN: 978-1617296598.

[94] Alex Preukschat and Dummond Reed. “Self-Sovereign Identity”. In: 1st ed. Manning, 2021, pp. 202–205. ISBN: 978-1617296598.

[95] Wullianallur Raghupathi and Viju Raghupathi. “Big data analytics in healthcare: Promise and potential”. In: *Health Information Science and Systems* 2 (1 Feb. 2014), pp. 1–10. ISSN: 20472501. DOI: 10.1186/2047-2501-2-3/TABLES/2.

[96] Amit Sahai and Brent Waters. “Fuzzy Identity-Based Encryption”. In: *Advances in Cryptology – EUROCRYPT 2005*. Ed. by Ronald Cramer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 457–473. ISBN: 978-3-540-32055-5.

[97] Hafida Saidi et al. “DSMAC: Privacy-Aware Decentralized Self-Management of Data Access Control Based on Blockchain for Health Data”. In: *IEEE Access* 10 (2022), pp. 101011–101028. DOI: 10.1109/ACCESS.2022.3207803.

[98] Kouichi Sakurai, Takashi Nishide, and Amril Syalim. “Improved proxy re-encryption scheme for symmetric key cryptography”. In: *2017 International Workshop on Big Data and Information Security (IWBIS)*. 2017, pp. 105–111. DOI: 10.1109/IWBIS.2017.8275110.

[99] Scribbr. *Sampling Bias and How to Avoid It | Types Examples*. 2023. URL: https://www.scribbr.com/research-bias/sampling-bias/ (visited on 04/05/2024).

[100] *Self-Sovereign Identity | The Moxy Tongue*. 2016. URL: https://www.moxytongue.com/2016/02/self-sovereign-identity.html (visited on 04/05/2024).

[101] Bhavye Sharma, Raju Halder, and Jawar Singh. "Blockchain-based Interoperable Healthcare using Zero-Knowledge Proofs and Proxy Re-Encryption". In: *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*. 2020, pp. 1–6. DOI: 10.1109/COMSNETS48256.2020.9027413.

[102] Bingqing Shen, Jingzhi Guo, and Yilong Yang. "MedChain: Efficient Healthcare Data Sharing via Blockchain". In: *Applied Sciences* 9.6 (2019). ISSN: 2076-3417. DOI: 10.3390/app9061207.

[103] Shuyun Shi et al. "Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey". In: *Computers and Security* 97 (2020), p. 101966. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2020.101966.

[104] Cyber Security Agency Singapore. *GUIDE TO CYBER THREAT MODELLING*. 2021. URL: https://www.csa.gov.sg/docs/default-source/csa/documents/legislation_supplementary_references/guide-to-cyber-threat-modelling.pdf (visited on 04/05/2024).

[105] Samuel Smith. *KERI website*. 2024. URL: https://keri.one/ (visited on 04/05/2024).

[106] Samuel M. Smith. *Key Event Receipt Infrastructure (KERI)*. 2021. arXiv: 1907.02143 [cs.CR].

[107] Manu Sporny and Leonard Rosenthol. *Cryptographic Hyperlinks*. Internet-Draft draft-sporny-hashlink-07. Work in Progress. Internet Engineering Task Force, May 2021. 11 pp. URL: https://datatracker.ietf.org/doc/draft-sporny-hashlink/07/ (visited on 04/05/2024).

[108] National Institute of Standards and Technology. *Engineering Trustworthy Secure Systems*. Tech. rep. NIST Special Publication 800-160v1r1. Washington, D.C.: U.S. Department of Commerce, 2022. DOI: 10.6028/NIST.SP.800-160v1r1.

[109] National Institute of Standards and Technology. *Foundations of a Security Policy for Use of the- National Research and Educational Network*. Tech. rep. NISTIR 4734. Washington, D.C.: U.S. Department of Commerce, 1992. DOI: 10.6028/NIST.IR.4734.

[110] TBD. *Web5 SDK*. 2024. URL: https://developer.tbd.website/docs/web5/ (visited on 04/05/2024).

[111] Marie Tcholakian et al. "Self-Sovereign Identity for Consented and Content-Based Access to Medical Records Using Blockchain". In: *Security and Communication Networks* 2023 (2023). DOI: 10.1155/2023/6025789.

[112] *Technischer Aufbau im Überblick - ELGA*. 2024. URL: https://www.elga.gv.at/technischer-hintergrund/technischer-aufbau-im-ueberblick/ (visited on 04/05/2024).

[113] W3C. *Credentials Community Group*. 2024. URL: https://www.w3.org/community/credentials/ (visited on 04/05/2024).

[114] W3C. *Decentralized Identifiers (DIDs) v1.0*. 2022. URL: https://www.w3.org/TR/did-core/ (visited on 04/05/2024).

[115] W3C. *Encrypted Data Vaults v0.1*. 2022. URL: https://identity.foundation/edv-spec (visited on 04/05/2024).

[116] W3C. *JSON-LD 1.1*. 2020. URL: https://www.w3.org/TR/json-ld11/ (visited on 04/05/2024).

[117] W3C. *RDF Dataset Canonicalization*. 2023. URL: https://www.w3.org/TR/rdf-canon/ (visited on 04/05/2024).

[118] W3C. *The did:keri Method v0.1*. 2021. URL: https://identity.foundation/keri/did_methods/ (visited on 04/05/2024).

[119] W3C. *Verifiable Credentials Data Model v1.1*. 2022. URL: https://www.w3.org/TR/vc-data-model/ (visited on 04/05/2024).

[120] W3C. *Verifiable Credentials Data Model v2.0*. 2024. URL: https://www.w3.org/TR/vc-data-model-2.0/ (visited on 04/05/2024).

[121] Yong Wang et al. "Cloud-Assisted EHR Sharing with Security and Privacy Preservation via Consortium Blockchain". In: *IEEE Access* 7 (2019), pp. 136704–136719. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2943153.

[122] Qi Xia et al. "BBDS: Blockchain-Based Data Sharing for Electronic Medical Records in Cloud Environments". In: *Information* 8.2 (2017). ISSN: 2078-2489. DOI: 10.3390/info8020044.

[123] Qi Xia et al. "MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain". In: *IEEE Access* 5 (2017), pp. 14757–14767. DOI: 10.1109/ACCESS.2017.2730843.

[124] Yan Zhuang et al. "A Patient-Centric Health Information Exchange Framework Using Blockchain Technology". In: *IEEE Journal of Biomedical and Health Informatics* 24.8 (2020), pp. 2169–2176. DOI: 10.1109/JBHI.2020.2993072.

[125] Renpeng Zou, Xixiang Lv, and Jingsong Zhao. "SPChain: Blockchain-based medical data sharing and privacy-preserving eHealth system". In: *Information Processing Management* 58.4 (2021), p. 102604. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2021.102604.

[126] Guy Zyskind, Oz Nathan, and Alex 'Sandy' Pentland. "Decentralizing Privacy: Using Blockchain to Protect Personal Data". In: *2015 IEEE Security and Privacy Workshops*. 2015, pp. 180–184. DOI: 10.1109/SPW.2015.27.