Breaking Barriers in Healthcare: A Secure Identity Framework for Seamless Access

Antonio López Martínez[a,*], Montassar Naghmouchi[b], Maryline Laurent[b], Joaquín García Alfaro[b], Manuel Gil Pérez[a], Antonio Ruiz Martínez[a]

[a]*Department of Information and Communications Engineering, University of Murcia, Murcia, 30100, Spain*
[b]*SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France, Palaiseau, 91120, France*

Abstract

The digitization of healthcare data has heightened concerns about security, privacy, and interoperability. Traditional centralized systems are vulnerable to cyberattacks and data breaches, risking the exposure of sensitive patient information and decreasing trust in digital healthcare services. In addition, healthcare stakeholders use various standards and formats, creating challenges for data sharing and seamless communication. To address these points, this article identifies all the healthcare stakeholders and translates each useful element of a patient's electronic health record (EHR) into Fast Healthcare Interoperability Resources (FHIR), to propose a complete role-based access control model that specifies which FHIR resources an actor is allowed to access. To validate this role model, three new use cases are defined, in which the various stakeholders interact and access the FHIR resources. Moreover, specific smart contracts are detailed to implement the role model in an automated way and provide a robust access control mechanism within healthcare organizations. The feasibility of the proposed access control mechanism is demonstrated through proof-of-concept and test performance measurements. Finally, the solution is validated as a realistic solution

*Corresponding author.
Email addresses: antonio.lopez41@um.es (Antonio López Martínez), montassar-bellah_naghmouchi@telecom-sudparis.eu (Montassar Naghmouchi), maryline.laurent@telecom-sudparis.eu (Maryline Laurent), joaquin.garcia_alfaro@telecom-sudparis.eu (Joaquín García Alfaro), mgilperez@um.es (Manuel Gil Pérez), arm@um.es (Antonio Ruiz Martínez)

adapted to the scale of a country based on health statistics.

Keywords: Self-Sovereign Identity, SSI, clinical environment, blockchain, access Control, FHIR, health data protection, security, privacy

1. Introduction

The digital transformation of healthcare has brought significant advancements in patient care, medical research, and administrative efficiency. Electronic Health Records (EHRs) have become an integral part of modern healthcare systems. They enable the seamless sharing of patient information among authorized professionals and institutions. This accessibility improves the quality of care by providing clinicians with a comprehensive medical history, facilitating informed decision-making, and promoting continuity of care across different settings [1].

However, the increasing digitization of health data exposes the sector to many challenges related to data security, privacy, and interoperability [2, 3, 4, 5, 6]. High-profile cyberattacks and data breaches have underscored the vulnerabilities inherent in centralized data repositories, where unauthorized access can lead to the compromise of sensitive personal information on a massive scale. For instance, the largest cyberattack in 2024 affected Change Healthcare, one of the largest healthcare payment processing companies. The attack was due to the non-use of Multi-Factor Authentication (MFA), which caused cash flow problems in doctors' offices and hospitals [7]. These incidents not only undermine patient confidence but also impose significant financial and reputational costs on healthcare organizations. Organizations estimate that the average cost of each data breach is approximately ten million dollars [8]. Moreover, stringent regulations governing health data, such as the General Data Protection Regulation (GDPR) in Europe [9] and the Health Insurance Portability and Accountability Act (HIPAA) in the United States [10], require robust measures to protect patient privacy and ensure compliance.

In response to these challenges, there has been a growing interest in decentralized identity management solutions that give individuals greater control over their personal data. Self-Sovereign Identity (SSI) frameworks offer a promising approach to addressing security and privacy concerns while improving interoperability between disparate healthcare systems. SSI enables users to own and manage their digital identities without relying on centralized

authorities, reducing the risk of single points of failure and unauthorized data sharing [11]. Additionally, SSI enhances patient privacy by allowing individuals to selectively disclose personal information, ensuring that only necessary data is shared with healthcare providers, thereby fostering greater trust in digital healthcare services [12].

To implement the SSI principles, Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) are integral to the practical implementation of the SSI framework. DIDs provide unique, user-controlled identifiers that remove reliance on centralized registries, thereby enhancing privacy and limiting single points of failure [13]. VCs provide cryptographically secure attestations of user attributes, enabling reliable and tamper-proof credential verification [12]. Distributed ledger (DL) technology underpins these components by providing an immutable and transparent ledger for recording DIDs and VCs, ensuring data integrity and facilitating interoperability among disparate healthcare systems. In addition, smart contracts (SCs) automate critical processes such as data access control, consent management, and emergency access protocols, ensuring these operations are executed securely and consistently without manual intervention.

To address these challenges with innovative technologies, we proposed a comprehensive framework in our previous work [13] that leveraged SSI technologies to enhance healthcare data security and interoperability. The framework detailed the essential infrastructure, application layers, and data management strategies for robust healthcare data ecosystems. However, while the framework provided a solid foundation, it did not address the specifics of access control mechanisms between different healthcare actors. For this, we go one step further and make novel contributions in this article: the identification of the different healthcare actors that can interact in the framework, the definition of the patient's EHR to identify the data available on it with the selection of the FHIR format, and the novel role-based access control model, hereafter referred to as the role model, that specifies which FHIR resources an actor is allowed to access. Moreover, we develop three different use cases to validate the proposed role model. Thus, this article proposes a complete smart contract model to materialize the established role model and to address the different requirements identified in our previous article. Moreover, we present the Proof of Concept (PoC) implemented to study the performance of the smart contract model created [14], analyzing the feasibility of our solution under realistic conditions.

The remainder of the article is organized as follows. Section 2 reviews

the literature and presents related work. Section 3 defines the role model, identifies the healthcare participants, creates the patient's EHR, and maps each participant to the specific EHR parts. Then, Section 4 summarizes our previous work and the access control mechanism developed through a novel smart contract model. Section 5 implements and validates the smart contract model defined, and tests the performance of our solution. Finally, Section 6 summarizes the work and explores some future works.

2. Related Work

In this section, we review the literature to check the state-of-the-art studies related to our work. Table 1 summarizes the research works that are considered and explained in this section. The features defined to compare the works appear in Table 1 as columns: i) the platform used; ii) proposed role model; iii) proposed access control mechanism; iv) provided data sharing mechanism; iv) definition of patient EHR (format, data, etc.); and v) adherence to the SSI paradigm.

To begin with, Pham et al. [15] implemented a remote system for healthcare. Its purpose was to retrieve information from sensors placed on patients, generate logs and alert doctors to abnormal situations. As a special feature, the authors proposed using a GPS sensor to retrieve the physical location of the patient in case of an emergency. They used SCs to collect data and Externally Owned Accounts (EOAs) in Ethereum to anonymize the patient's identity. They proposed two SCs for registering patients and doctors, one for assigning patients to a doctor, and one for uploading telemetry data from the patient's devices to the platform. However, they lacked offline data storage and secure communications, which are key aspects of a DL-based healthcare remote system. Dagher et al. [16] proposed a privacy-preserving framework for EHR access control and interoperability. The primary characteristic of this framework was the proxy re-encryption technique, where a private key is distributed among the proxies in pieces, allowing them to encrypt messages but not decrypt them without the full key. These proxies interact with the DL through the proxy re-encryption contract, which manages the process of encrypting and decrypting the information using the re-encryption scheme designed. They implemented smart contracts to provide patient data registration and access control. Besides, they developed a smart contract to classify the DL nodes according to patients, providers, or third parties, which

Table 1: Summary of the state-of-the-art studies revised.

Ref.	year	Platform	Role model	Access control	Data sharing	Patient EHR	SSI paradigm
[15]	2018	Ethereum	○	●	●	○	◒
[16]	2018	Ethereum	◒	●	●	○	○
[17]	2019	Ethereum	●	●	●	○	○
[18]	2020	Hyperledger Fabric	●	●	◒	●	○
[19]	2020	Ethereum	○	●	●	○	○
[20]	2021	Ethereum	○	●	●	○	○
[21]	2021	Hyperledger Fabric	◒	●	●	○	○
[22]	2022	Hyperledger Indy/Aries	○	●	●	●	●
[4]	2023	Ethereum	○	●	●	○	○
[6]	2024	Ethereum	●	●	●	○	○
Ours	2025	**Hyperledger Fabric**	●	●	●	●	●

we believe is an unrealistic approach for patients who cannot implement a node.

Daraghmi et al. [17] developed MedChain, a DL-based system for EHR access and permissions management. They provided specific SCs for different functionalities: node consensus, history of health activities performed by healthcare providers for patients, placement of health records, tracking logs, access control, and re-encryption. This solution also implemented a novel incentive mechanism for EHR providers to maintain and protect health records. They measured the efficiency and performance of their solution, and concluded that DL-based solutions can achieve similar results to traditional systems. Tanwar et al. [18] implemented a DL-based patient-centered approach to provide a permission-based EHR sharing system. Their system has four actors: Patient, Clinician, Lab, and System Admin. Although they presented a role model, they do not cover all possible actors that may appear in the healthcare domain, for example, nurses, insurance, and pharmacies. Finally, the authors proposed different SCs to upload EHRs and grant and revoke access to them by patients. Khatoon [19] proposed a healthcare management system. The author created a Decentralized App (DApp) to manage data sharing for lab results, communication between patients and service providers, healthcare reimbursement, clinical trials, and outpatient surgical procedures. Also, Khatoon used a dataset of health data to feed and test DApp. However, transactional and operational costs were associated with using the Ethereum platform.

To continue, Omar et al. [20] developed a smart contract-based policy management system for EHRs in smart cities. The authors wanted to keep patient's insurance policies transparent to them. They provided a module-based architecture with encryption, verification, and policy manage-

ment modules. However, they did not clearly explain which SCs were implemented. Chelladurai et al. [21] created another DL-based EHR management system. They proposed SCs for patient registration, patient assigning and updating health data, data sharing, and viewership permissions. The innovation of this work is that they collected real health records for the tests, and evaluated the performance of the delivered smart contracts. However, they stored the patient data in the DL, which present a strong privacy issue given regulations such as GDPR and HIPAA. Harrel et al. [22] implemented a novel identity wallet through the SSI paradigm and VCs and DIDs technologies. The authors defined the VC schemas, including the data fields each credential should have and the relevant patient health data. Nevertheless, they did not manage the different healthcare actors and their permissions in health data management.

Ghani et al. [4] presented a DL-based system for access management in telemedicine. As an interesting detail, the authors used an Interplanetary File System (IPFS) to store the patient data. They proposed a set of requirements and explained how each requirement and its implementation covered the GDPR. Finally, Kalita et al. [6] developed an SC to provide access control to patient data for associated parties. They focused on the gas consumption of this SC in Ethereum platform to provide efficient management of usage cost. The proposed SC limits participation to legitimate members approved by the owner, and stores health data with selective disclosure of information. The authors also identified four categories of participants: Patient, Doctor, Hospital, and Insurer.

After presenting all the revised studies and considering Table 1, we reach different conclusions. All works manage access control and health data sharing, but identification of healthcare participants, patient EHR, and use of the SSI paradigm are still an unexplored area. Therefore, we identify specific gaps to cover in the literature: i) a comprehensive identification and definition of the participants that can appear in the healthcare domain, as many works consider only patients, doctors, and hospitals, but there are other actors like researchers and IT specialists who interact with the patient health data. As general protection regulations like GDPR state, all participants working with sensitive patient data must be identified and have the consent of the patient to use their data; ii) there is no specific definition of the data or resources that make up a patient EHR. This combination of roles with the definition of patient EHR is still not covered in the literature; only Harrel et al.'s work [22] defines the data fields for the health VCs, but without con-

sidering the standards presented in the literature; iii) the potential of SSI to empower patients as true owners of their data, combined with technologies such as smart contracts (SCs) to implement these solutions; iv) the implementation of Proof of Concepts (PoCs) and feasibility analyses to develop new healthcare solutions.

3. Role Model for the Healthcare domain

This section defines a novel role model for accessing and sharing patient data. For this, our contribution is divided into specific parts: i) identification of the actors involved in the healthcare environment; ii) definition of the patient's EHR, considering the format and the types of health data included (appointments, medications, allergies, etc.); and iii) creation of role-based access for the patient's EHR model, defining how identified actors can access the EHR and which parts they are permitted to view.

3.1. Healthcare actors identification

Many different actors are involved in the healthcare domain. Defining a role model requires a complete understanding of these entities, their relevance, and the role that each one plays in the healthcare environment. As an important aspect, we consider the patient as the center of the use case, and the roles are the different entities that interact with the patient in the clinical procedures, medical appointments, etc. To discover such a list of healthcare participants, we have explored the literature and general information on medical procedures, with a focus on the interacting actors [23, 24]. Figure 1 shows the complete list of identified actors, defined as follows:

- **Patient family**: This role includes people who have a direct relationship with the patient. Family and friends are here. This role includes both family and friends because special cases can arise, such as poor parental relationships or the absence of friends or relatives.
- **Primary care provider**: This role includes general practitioners and family doctors. In many cases, they are the starting point of healthcare workflow.
- **Specialist Provider**: After meeting the primary care professionals, patients may be referred to specialists, who focus on specific areas of medicine, such as cardiology or oncology.

- **Nurse**: This role includes all members of the nursing profession, from technical assistants to nurses. They are involved in physical care and management of patients.

- **Laboratory Staff**: These are the clinical laboratory professionals who manage the different patient samples received for analysis.

- **Pharmacist**: This role represents the pharmacies, in charge of supplying the patients with the prescriptions established by the practitioners. Moreover, there may be other entities that provide medications, such as supermarkets or retail stores. They may provide medications of general use if the government allows it.

- **Public Health Official**: This role works on community health, disease prevention, and health promotion at the population level. For instance, the regulation of COVID-19 quarantine in 2020 was defined by this role.

- **Healthcare Administrators**: This role includes the people who manage the operations of healthcare facilities, including hospitals and clinics, to ensure efficient and effective delivery of services.

- **Health IT Specialist**: This role is in charge of the health systems from the data and technology perspective. For instance, a network administrator who is in charge of a hospital's various private network configurations.

- **Medical Researcher**: This role conducts clinical and biomedical research to advance medical knowledge and develop new treatments. Furthermore, companies working on medical technologies (X-ray machines, blood analyzers, etc.) may also appear in this role.

- **Insurance**: This role includes the companies that manage health insurance plans, develop policies, and ensure the financial aspects of patient care. For public areas, this role can be eliminated.

- **Regulatory and Compliance Officer**: This role ensures that healthcare organizations comply with laws, regulations, and standards (GDPR, HIPAA, etc.), and maintain quality and safety. In European Union countries, it also includes the Data Protection Officer (DPO), defined in the GDPR, who ensures compliance with the GDPR.

- **Pharmaceuticals:** These companies develop and distribute drugs and medical devices, focusing on innovation and safety.
- **Community Health Worker**: This role works with local communities to improve health outcomes through education (workshops, community events), outreach, and support services (social awareness, smoking cessation groups, weight loss support groups).

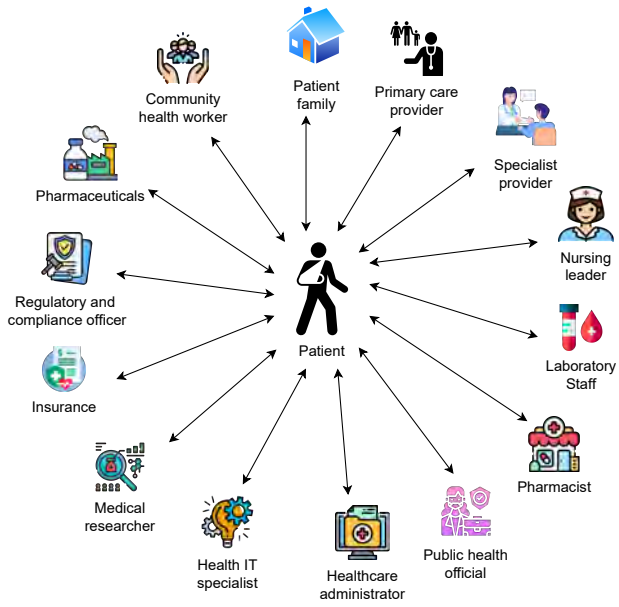

Figure 1: Actors involved in the healthcare sector.

3.2. Patient's EHR as a collection of FHIR resources

In Section 2, many works have been reviewed, and some of them present access control mechanisms for health data. However, none analyzes what information is stored in a patient's EHR. These findings are critical to the healthcare interoperability challenge. With this purpose identified, we undertook a specific effort to define the EHR. First, we searched protocols and languages used in the healthcare domain. In this context, López Martínez

et al. [25] listed the main protocols that appeared in the clinical environment, a subdomain of healthcare where the lifecycle of the patient sample is managed. This starting point allowed us to analyze and study interesting data representation standards: HL7 v2 [26], HL7 v3 [27], Fast Healthcare Interoperability Resources (FHIR) [28], and Clinical Document Architecture (CDA) [29].

Essentially, HL7 v2 and HL7 v3 are messaging standards for the electronic exchange of clinical and administrative healthcare data. FHIR is also a standard designed to facilitate the exchange of health information across disparate systems in a consistent, simple, and secure manner. Finally, CDA is an XML-based standard of HL7 v3 that specifies the structure and semantics of clinical documents to be exchanged between healthcare providers. Analysis of these protocols reveal the following key aspects:

- FHIR incorporates modern web technologies like APIs, JSON, and XML, as opposed to HL7 v2 and v3, which are more complex and sometimes require custom interfaces for integration, increasing development time and cost. CDA is document-based, less flexible, and more difficult to manipulate for real-time data exchange.

- FHIR is modular, consisting of resources that can be combined in various ways to meet specific use cases. This makes it to adapt FHIR to different scenarios. HL7 v2, v3, and CDA are more rigid and less adaptable, often leading to incomplete or inconsistent implementations [30].

- FHIR is gaining traction in the healthcare industry. While HL7 v2 and v3 have been widely used in the past, their adoption is declining in favor of FHIR [31].

Therefore, **FHIR** is the standard chosen to represent our patient's EHR. Digging deeper into this standard, FHIR defines all possible healthcare data into five resource families: Clinical, Diagnostic, Medication, Workflow, and Financial. For instance, the Clinical family includes resources such as AllergyIntollerance, CarePlan, Condition (problem), etc.

For our proposal, we envision a subset of all resources that FHIR provides, taking into account those requested for a patient's daily use. Moreover, our FHIR resource selection is supported in a realistic use case presented in [25]. The clinical use case included in such work is based on a typical interaction

between patients and healthcare professionals, where: i) the patient requests a doctor's appointment; ii) the doctor initiates a procedure to collect a patient sample; iii) a nurse extracts the sample and sends it to a clinical laboratory; iv) the sample is analyzed, and the result is obtained; and v) the doctor writes a prescription for the patient's condition.

In Figure 2, we present a diagram with the selected FHIR resources and the existing connections between them. The nomenclature for naming the resources is taken from the officially available FHIR ontology [28]. The various FHIR resources are defined as follows:

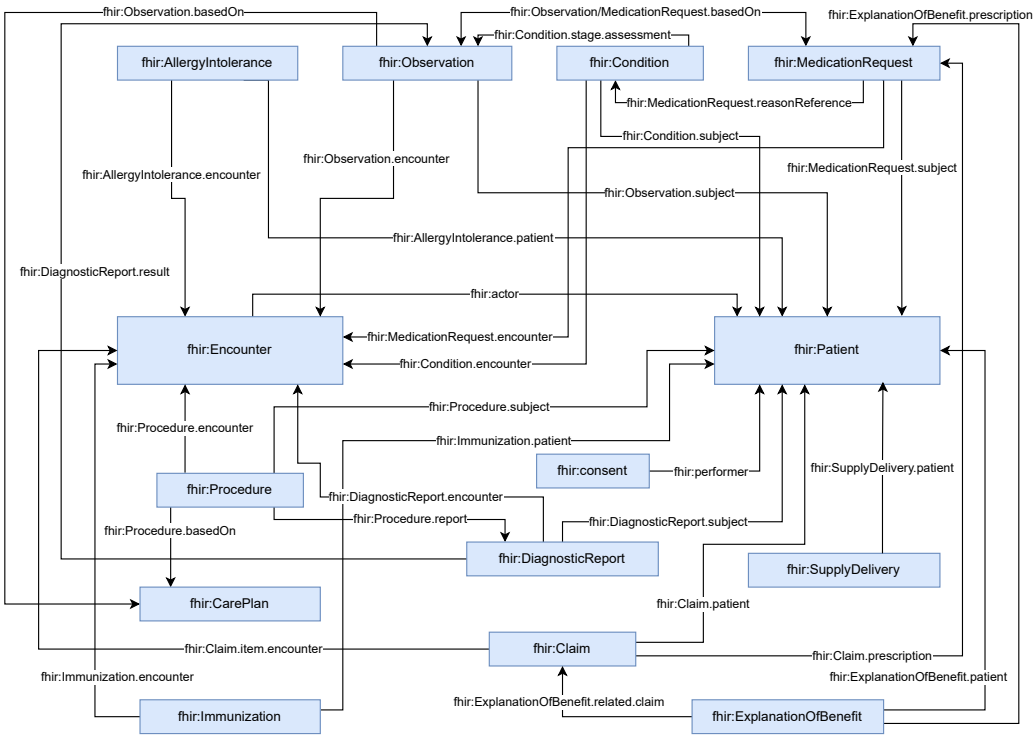

Figure 2: Patient's EHR composed of FHIR resources.

- **Patient**: It includes all the demographic information about the patient.
- **Encounter**: This type represents an interaction between a patient and a healthcare professional to provide a service or assess health status.

- **Observation**: This type includes the clinical records. Here, we can find vital signs, laboratory data, imaging results, clinical findings, device measurements, clinical assessment tools, personal characteristics, and social history like tobacco use.
- **Condition**: This type represents an identified problem, situation, or clinical concept that may affect a patient. For instance, a problem detected by a laboratory result.
- **MedicationRequest**: This type covers all types of medication orders for a patient.
- **Procedure**: This type includes actions that are or have been performed on or for a patient, such as surgical procedures, diagnostic procedures, and biopsies.
- **AllergyIntolerance**: This type represents allergies or intolerances of the patient.
- **Immunization**: It provides the current and historical administration of vaccines to the patient.
- **SupplyDelivery**: This type represents the record of medication delivered and administered.
- **DiagnosticReport**: This type provides the information typically received by a diagnostic service upon completion of investigations. It can contain atomic results, text reports, images, and codes.
- **CarePlan**: This type explains how one or more practitioners intend to care for a particular patient.
- **Claim**: This type contains the list of professional services and products provided to a patient and sent to insurance for reimbursement.
- **ExplanationOfBenefit**: This type represents the official information about the claim, adjudication details, and optional account balance information.
- **Consent**: We envision the patient consent as an FHIR resource, where different consents can be created, such as emergency consent, research consent, etc.

3.3. Role-based access control model

This paper defines what part of the patient's EHR can be shared with each participant. Essentially, in the model we propose, healthcare actors are understood as roles, and each role should have access to only the minimum information (principle of minimization) necessary to do their job correctly.

To enforce this role-centric approach, we evaluated several access control models [32]. Discretionary Access Control (DAC) becomes extremely complex in large and dynamic environments, even with potential conflicts that can arise in these domains. Mandatory Access Control (MAC) is a rigid and centralized approach set by a central authority, which does not apply to our decentralized framework and patient-centric approach. While Attribute-Based Access Control (ABAC) provides fine-grained context-awareness, the complexity of defining and managing comprehensive policies can be prohibitive. Therefore, Role-Based Access Control (RBAC) is selected as the most suitable model. RBAC naturally aligns permissions with the functional healthcare actors already identified, effectively implements the principle of minimization by design, and provides a proven balance of security, manageability, and scalability for healthcare information systems.

Figure 3 graphically represents the RBAC model, including the healthcare actors and FHIR resources, and indicates which resources participants can access or have visibility of. To enforce Figure 3, we present the fundamentals behind each relationship:

- Patient Family: They have access to *Patient*, *Condition (Opt.)*, and *CarePlan (Opt.)* resources. We envision the patient selecting who they want to access their demographic information for possible emergencies and configuring their contact list. We mark *Condition* and *CarePlan* as optional since patients should decide if they want to share their diseases/care plans with their trusted people.

- Primary Care Provider: This group visualizes *Condition*, *Observation*, *Encounter*, *CarePlan*, *MedicationRequest*, *AllergyIntolerance*, *Immunization*, *Procedure*, and *DiagnosticReport* resources. They are the general and family practitioners, and, in many cases, they need to have a complete view of the patient's EHR because the patient presents with generic symptoms that requires the contextual information of the EHR.

- Specialist Provider: This actor accesses *Condition*, *Encounter*, *DiagnosticReport*, *MedicationRequest*, *Observation*, and *Procedure* re-

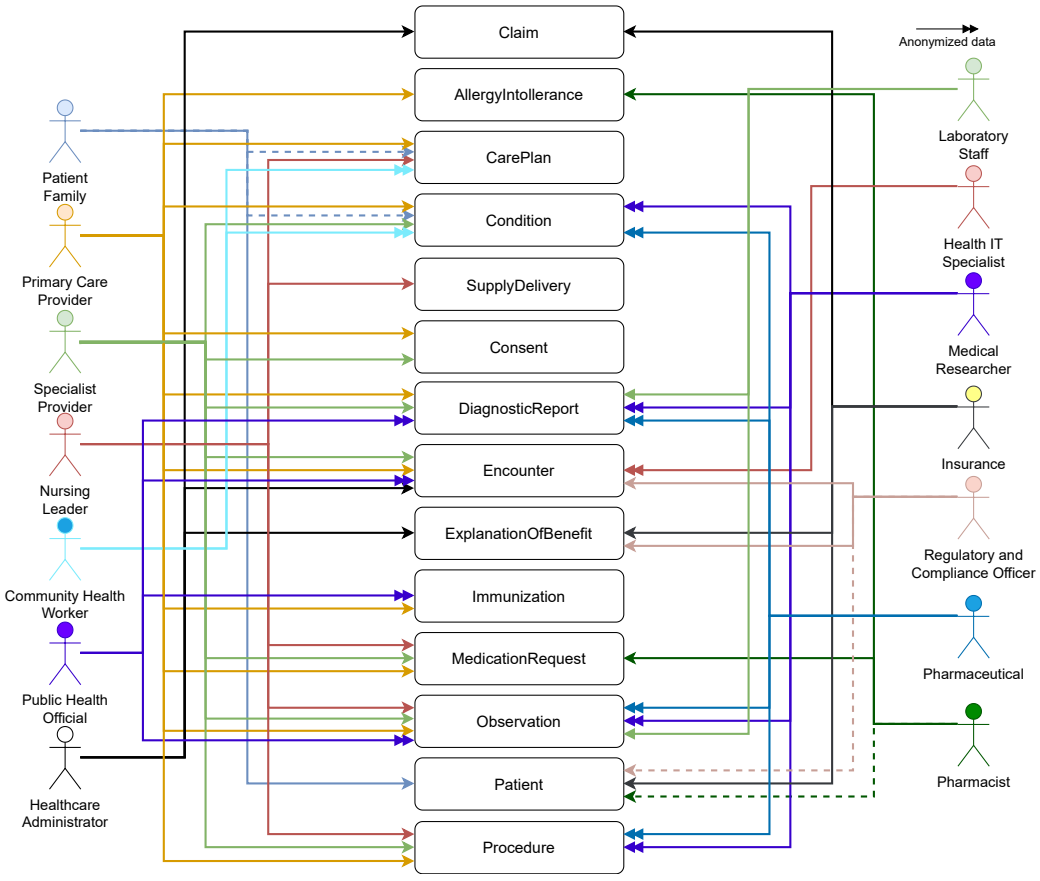

Figure 3: Role-based access control model for patient EHR.

sources. Specialists request the appropriate patient EHR for their specialty. An important aspect is that they do not have access to the *CarePlan* by default, but they can be creators of such information. Therefore, they could request patient access to this data under certain circumstances.

- Nursing Leader: They inspect *CarePlan*, *SupplyDelivery*, *MedicationRequest*, *Observation*, and *Procedure* resources. This group acts after the practitioners. They dispense medications or collect patient samples for laboratory analysis. To do this, they need access to the data created

by the practitioner about the steps to be taken for the patient.

- Community Health Worker: This group accesses *Condition* and *CarePlan* resources. This role works primarily through education and awareness of community groups. For this, they need to know the trends in conditions and care plans available for health issues.

- Public Health Official: They have access to *Observation*, *Immunization*, *Encounter*, and *DiagnosticReport* resources. To create effective public health regulations, professionals working in this direction should have access to global and general EHR data. However, this data should be shared anonymously, using data privacy mechanisms, since they do not require knowing any information about the people's identities.

- Healthcare Administrator: This role inspects *Claim*, *Encounter*, and *ExplanationOfBenefit* resources. This administrative role controls the wellness of the service provided to the patient in the healthcare facilities. This wellness is represented through encounters between patients, healthcare professionals, and financial staff.

- Laboratory Staff: These actors have access to *DiagnosticReport* and *Observation* resources. They are specific to the clinical laboratories and manage information about patient samples.

- Health IT Specialist: They have access to *Encounter* resources. This role is in charge of managing and protecting the healthcare technological infrastructure. Therefore, they do not need EHR data except for the services provided by healthcare professionals to patients, and they coordinate the correct delivery of these services from an IT perspective.

- Medical Researcher: This actor visualizes *Condition*, *DiagnosticReport*, *Observation*, and *Procedure* resources. When searching for new treatments, diseases, or other medical conditions, researchers need information about patients' diagnostic results and health issues. They do not need patient identity, so the data is anonymized. However, if the data are from a clinical trial and the volunteers are experiencing adverse situations, the reidentification of such data may be permitted.

- Insurance: This group requires *Claim*, *ExplanationOfBenefit*, and *Patient* resources. In this case, the insurance companies need the activities

at the healthcare facilities and the patient's financial information for payment collection.

- Regulatory and Compliance Officer: They collect *Encounter*, *ExplanationOfBenefit*, and *Patient (Opt.)* resources. This group ensures that healthcare organizations comply with laws, regulations, and standards to maintain quality and safety. They may request specific patient demographic information when a specific problem is identified.

- Pharmaceutical: This actor requires *Condition*, *DiagnosticReport*, *Procedure*, and *Observation* resources. They focus on the development of new medications or health procedures. To do this, they need to know how diagnostics are performed and the procedures raised from them.

- Pharmacists: This group visualizes *MedicationRequest*, *AllergyIntolerance*, and *Patient (Opt.)* resources. Pharmacies dispense medications prescribed by practitioners. They may optionally request patient demographic information, such as address, in order to ship the medication.

In conclusion, this role model definition provides a comprehensive view of the different participants interacting with the patient. Since all possible medical actors have been identified, this contribution facilitates the creation of a novel and complete SSI access control mechanism for patient data, as we present in the following sections.

3.4. Healthcare use cases

To validate the presented role model, we select three different healthcare use cases. Each use case will demonstrate the interactions between the identified healthcare participants and the different FHIR resources of the patient's EHRs that are accessed.

3.4.1. Use Case 1 - Patient sample lifecycle

Considering the work of López Martínez et al. [25], we develop this patient sample lifecycle use case to study and analyze how the role model covers these daily interactions in the healthcare domain. In this context, Figure 4 shows the steps of the use case.

First, the patient has a general appointment (*step 1*) with the primary care provider to explain their symptoms and concerns. This actor accesses the patient's past encounters, observations, and conditions, and completes

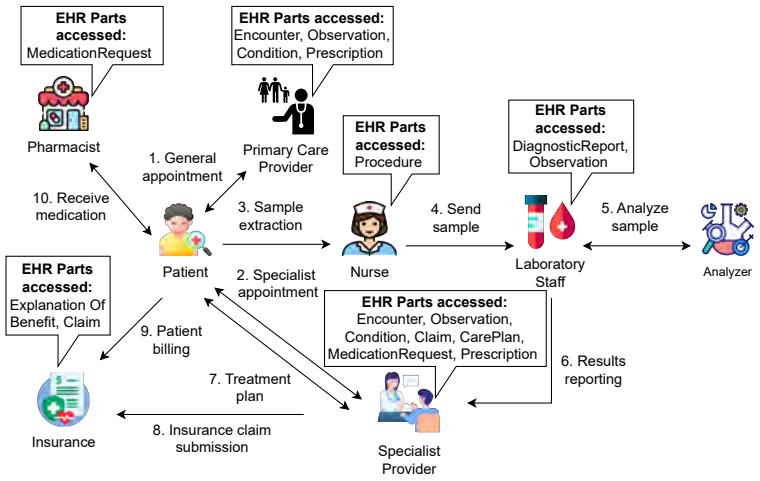

Figure 4: Use case 1 - Patient sample lifecycle.

the creation of a specific appointment with a specialist provider. On the selected date, the patient visits the specialist provider (*step 2*), who begins the process of taking blood samples to investigate the disease. To continue, the patient undergoes a blood sample (step 3), where the nurse follows the specialist provider's established protocol to collect the sample and send it to the laboratory (*step 4*). Once the sample arrives at the lab, the lab staff begins the analysis (*step 5*), by placing the sample in the analyzer, which generates the test results.

With the results obtained, the laboratory staff reports them to the specialist provider (*step 6*), who prepares a new appointment with the patient. The specialist provider develops the treatment plan (*step 7*) after discussing the blood test results with the patient. This treatment plan includes a prescription for medication to be taken in the pharmacy. This step completes the medical workflow, and for private domains, the insurance company must be updated with the medical procedure performed (*step 8*). For summary purposes, we have included this step at the end; however, in the real world, insurance claims are likely updated at each appointment, extraction, and test analysis. Therefore, the patient must meet with the claims and pay the generated billing (*step 9*). Finally, the patient goes to the pharmacy, provides the medication order generated by the specialist provider, and the pharmacy dispenses the medication.

3.4.2. Use Case 2 - Clinical Trial

We propose this clinical trial use case inspired by various works [33, 34]. Essentially, a pharmaceutical company has developed a new medication to improve blood glucose control in patients with Type 2 Diabetes. A clinical trial is planned to assess the effectiveness and efficacy of such a medication. Figure 5 presents the complete use case.

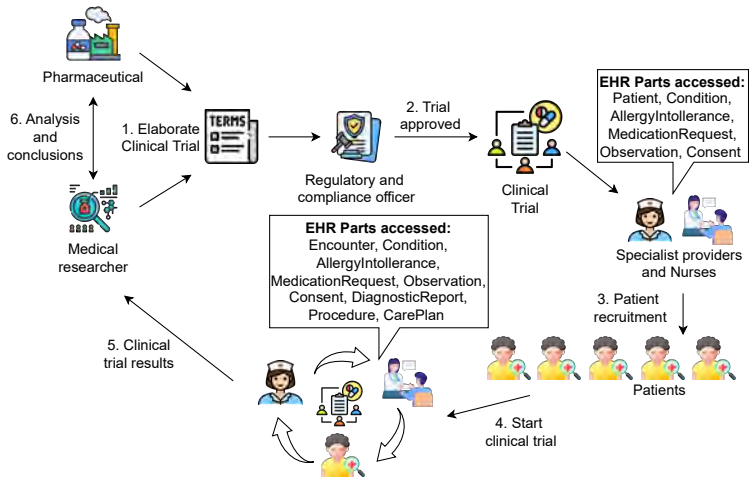

Figure 5: Use Case 2 - Clinical trial.

The use case starts with study design and protocol development, where the pharmaceutical company and medical researchers design (*step 1*) the clinical trial objectives, protocol, methodology, patient eligibility criteria, etc. The final document is sent to the regulatory and compliance officer for approval. After its approval (*step 2*), there is a site selection and preparation, as shown in Figure 5. The next procedure is patient recruitment (*step 3*), where specialist providers and nurses need different parts of the patient's EHR to decide which patients can participate in the trial. Consent is obtained from the patient. The professionals verify the diagnosis of Type 2 Diabetes by checking the conditions, assessing allergy intolerances for possible contraindications, reviewing medication requests to determine current medications, and observations to note vital signs and recent lab results.

The next step encompasses the execution of the clinical trial (*step 4*). During the clinical trial, the specialist providers and nurses need to access the encounters to document the different visits, the observations to read and

store different tests performed, the procedures as they document the different techniques considered, the diagnostic reports to include detailed results from diagnostic imaging or tests, the care plans to update them as needed regarding the clinical trial, and, finally, the conditions to document possible issues. After the clinical trial is completed, the results are sent, without correlation of patient identity (*step 5*), to the original pharmaceutical company and medical researchers, who analyze the results and extract the specific conclusions (*step 6*).

3.4.3. Use Case 3 - Emergency case

Our final use case involves a serious car accident where a patient is found unconscious. In this case, emergency medical services are dispatched to provide to the scene to provide immediate care. Figure 6 lists all the steps of this use case.

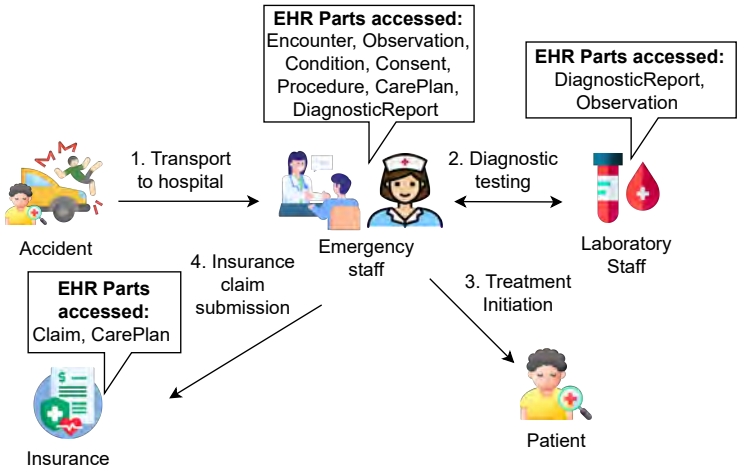

Figure 6: Use Case 3 - Emergency.

Firstly, the patient is transported to the hospital by ambulance (*step 1*). Upon arrival, the patient is stabilized and admitted to the emergency department. In this step, some observations and procedures can be performed to provide rapid first-aid. In the emergency department, the specialist provider enrolls the patient in an encounter and orders immediate diagnostic tests (*step 2*). After such diagnostic tests have been performed, the specialist

provider initiates a treatment plan (*step 3*), creating specific medication orders, procedures, and care plans. Finally, the insurance company is contacted with information about the healthcare plan implemented for the patient (*step 4*). Other procedures may appear here, such as surgery, ongoing monitoring, and care, etc.

4. Proposal: SSI and Role-Based Access Control to patient EHR

In Section 3.1, we introduced the wide variety of actors involved in the healthcare domain. This presents a real challenge in managing access to patient EHRs, which we have also defined in Section 3.2. Moreover, there is a need to protect and secure the patient's EHR, preserving the privacy of the patient's data and providing the patient with the sovereignty to decide and select who and why the actor accesses their data, as modeled in the role model proposed in Section 3.3. In our previous work [13], we introduced a comprehensive framework for secure healthcare data management using SSI technologies. That framework detailed the infrastructure, application agent-based layer, and data handling processes essential for empowering patients as true data owners. Building on this foundation, in this section, we present a novel privacy-preserving and role-based access control mechanism within this framework, thus enhancing the security and efficiency of patient data access. This mechanism is based on the fundamentals and model presented in Section 3.

4.1. SSI framework

Figure 7 is an overview of the platform. Firstly, we have normal processes and extra features defined in the first version of the framework. On the one hand, the normal processes are user certificate creation, user authorization, mutual authentication of users, and information sharing. These procedures ensure a normal working of the patient's daily interactions with practitioners and laboratories. On the other hand, the extra features include patient wallet recovery, health data access revocation, VC revocation, and patient data sharing in emergencies. Secondly, the framework architecture consists of two parts: a user wallet and a blockchain platform.

On the one hand, the user wallet serves to store and share patient health data, manage DIDs, VCs, and cryptographic keys. For that, the *user wallet* component follows an agent-based approach, implementing an edge agent hosted in a mobile App on the user's device and a cloud agent in a cloud

infrastructure. The cloud agent intervenes whenever an edge agent wants to connect to another agent, for instance, when patients share health data with healthcare actors, or with the blockchain platform, for instance, to store and read DIDs from the ledger. Moreover, the cloud agent is used to securely maintain a backup of patient data and publish the specific part of the patient data in read-only mode when the patient decides to share it with a healthcare actor. In the previous work, patient data was stored locally on their devices. However, we propose now to store patient data on the cloud agent, where we already have a backup, facilitating data-sharing processes. On the cloud agent, we can facilitate data sharing between the patient and healthcare participants by using the consent resource proposed in Section 3.2, which is part of the patient's EHR, and the smart contract model presented in Section 4.2.

On the other hand, the *blockchain platform* provides a relationship of trust between the patient and healthcare actors, and ensures data integrity, as we implement a DL composed of different organizations (DL nodes). The platform of choice is Hyperledger Fabric [13], a permissioned DL that comes with pluggable Certificate Authorities (CAs) and Membership Service Providers (MSPs) that allow organizations to issue X.509 certificates to their clients and collaborators. The MSP uses these X.509 certificates to authorize users to write or read data on the ledger. Another key feature of the Hyperledger Fabric is that we can create private channels, where private transactions occur between selected nodes. This allows for better data segregation on the ledger. In fact, the certificates a user obtains are required to access these private channels, so we have effective access control over who can be on which channels.

Hyperledger Fabric also enables the deployment of Smart Contracts (SCs), called chaincodes, which can be deployed on both private and public channels. SCs running on private channels only require the participation of peers or nodes that are part of the channel, and only users authorized to the channel can invoke the SCs. The ledger private channels will store the public keys of registered users of different organizations (each organization will have a private channel) and the role model and permissions defined by each organization. The user wallets will also have access to the private channels to identify, authenticate, and authorize data access based on the role model we defined above. All of this is specified and implemented in SCs, which are described in the Smart Contract Model in Section 4.2.

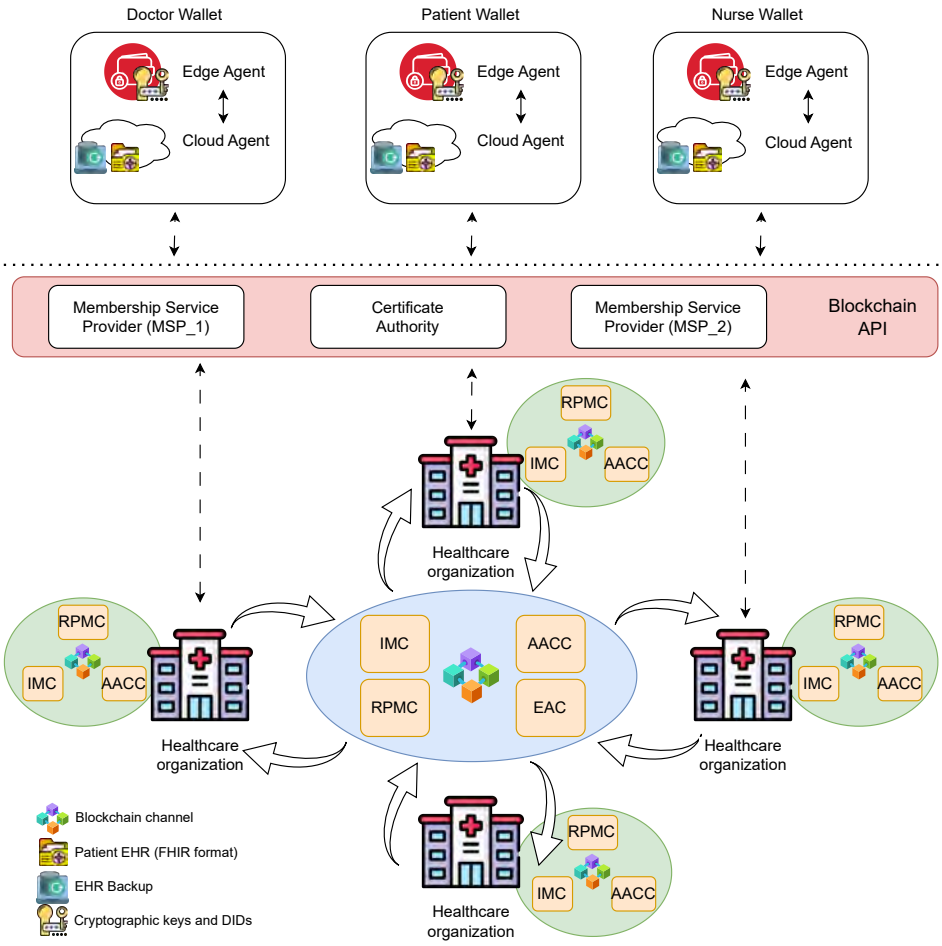

Figure 7: Platform architecture.

4.2. Smart Contracts Definition

We define a decentralized access control mechanism based on smart contracts, using the defined role model of Section 3.3, thereby enhancing security and privacy while granting patients greater control over their health records. Moreover, we intend to allow organizations to customize this role model to fit their specific structure, roles, and permissions in each organization's private channel. We also provide clients (patients) and collaborators (doctors, nurses, care-providers, etc.) of an organization with the opportunity to register their identifiers within the organization, and build an effective identity and access management protocol. This protocol enables users to identify, authenticate, and authorize other users to access their resources (EHR records, lab results, medical data) based on the roles assigned to them by the organization, using the identifiers and authentication keys they have chosen. We also design an emergency access protocol, that enables emergency doctors to access the EHR of an unconscious patient. All operations within our framework are ledger-enabled, and we provide audit and compliance mechanisms for auditability and traceability when needed.

We propose designing four distinct smart contracts. The first SC is the **Identity Management Contract (IMC)**, which allows the DL platform to act as a Verifiable Data Registry (VDR), by storing the DIDs and public keys used to identify and authenticate users. IMC is deployed on each private channel of a given organization (hospital, laboratory) to identify its collaborators and patients. They are also deployed on other shared channels, such as the emergency channel (used for our emergency access protocol). If the IMC is deployed on a single organization's private channel, then it is under the control and responsibility of that organization. In cases where the IMC is deployed on shared channels, it is the subject of consensus among all organizations. Users (patients, doctors, health-care professionals) generate their own DIDs and associated keys on their wallets, and use the IMC of an organization they are associated with, to register the DID and the public keys needed to authenticate the DID owner. However, before joining the private channel and invoking a contract, the user must obtain an X.509 certificate from the organization's MSP. This step allows the organizations to correctly identify and verify the user - outside of the ledger in the real world - before allowing them into the channels, which helps establish a relationship between the user identity within the organization and their general real-world identity (social security number, qualifications as a doctor or a healthcare professional, etc.). Note that a user may use the same DID across multiple

channels with many organizations or they may use different identifiers for different organizations.

The second SC is the **Role and Permission Management Contract (RPMC)**. This SC is critical to the access control model in the framework. Given the role model defined in Section 3, the organization is responsible for assigning roles to users and specifying the access policies for those roles through the RPMC it deploys on its private channel or agrees to deploy on channels to which the organization belongs. The organization invokes the RPMC to assign roles to users who are already registered through the IMC. The organization has already verified the user with the X.509 certificate, which makes it easier to assign roles to users and associate those roles with the registered DIDs.

The RPMC is deployed in the same way as the IMC, and it is customizable for each organization regarding roles and policies. In an access request, the resource owner (the patient owning an EHR, a laboratory owning a lab result, etc.), calls the RPMC to verify the roles and the policies associated with that role, and to grant access to the appropriate FHIR resources. For example, a nurse can have policies for only reading the Consent part of the FHIR EHR.

In addition, we include the **Audit And Compliance Contract (AACC)**. The AACC records all logs of the various procedures performed within the framework. For example, the registration of a public key by the IMC, the assignment of a role by the RPMC, or the updates of access policies or role models. The AACC can optionally log identity authentication requests (by listening to DID resolution events from the IMC), authorization requests for emergency access, and access operations (by listening to the events from the RPMC).

Figure 8 illustrates the processes explained so far. It shows a sequence diagram including the IMC, RPMC, and AACC, where the doctor is granted access to a hospital's private channel via the MSP, the doctor retrieves a certificate from the MSP (steps 1 and 2), he registers his DID using the IMC (step 3), and the hospital assigns the doctor - using that DID - to a role using the RPMC (step 4). Later, to access a patient's data (the patient has already done the same procedure as the doctor and has been assigned to a patient role), we see how the flow works using the three smart contracts. The doctor sends an access request to the patient wallet specifying his role and DID and including a signature using the private key associated to this DID (step 5). This request is redirected to the cloud agent (the online part of the wallet, as described in step 6) so that the authentication

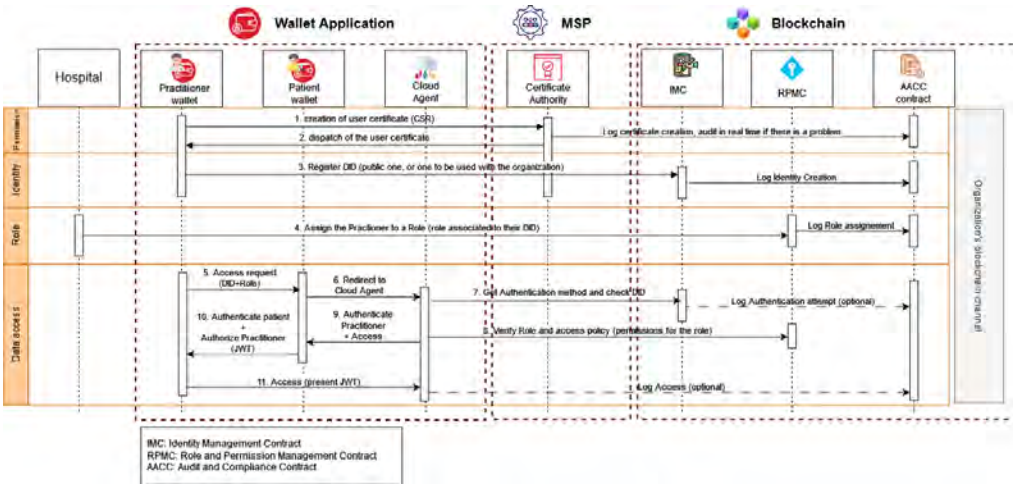

Figure 8: Normal EHR sharing workflow.

method (public key) can be retrieved from the ledger via the IMC (step 7). The cloud agent retrieves the role and permissions from the ledger via the RPMC (step 8) and authenticates that the request is coming from the doctor using that public key. After verifying the roles and permissions against the request in step 9, the cloud agent indicates to the patient's wallet whether the doctor has been authenticated and whether the request matches the roles and permissions. The patient now sends an authenticated access token (JWT) that allows the doctor to access data on the cloud agent (step 10). This JWT expresses the access rules and permissions, the authenticity of the patient, and the patient's consent. The doctor presents the JWT to the cloud agent for access (step 11) and may store it if it can be reused multiple times and has a specific validity period. The AACC can be personalized to log as many events as an organization needs or wants, and as required by law. In fact, Hyperledger Fabric even provides the ability to log read transactions or queries on the ledger itself, which means that we can even optionally log read events required for authentication or authorization. This could prove useful for accountability, especially for the emergency case we describe next.

In addition to the three main SCs, we include an **Emergency Access Contract (EAC)** that implements an emergency access procedure that allows emergency doctors or similar authorized roles to access the patient's data on the cloud agent in some emergency situations where a patient is unconscious and unable to perform the normal EHR sharing process. Fig-

ure 9 shows this workflow. We define an **Emergency Channel** (EC) that gathers several hospitals and emergency services, where the IMC and RPMC are used to register users and assign roles to them, e.g. a registering patient who provides emergency-access consent to emergency services, and a hospital that assigns a doctor the role of emergency-doctor, etc.. The EAC provided in this channel is used to generate **Emergency Tokens** (ETs) for a role authorized to access EHR in an emergency (emergency-doctor, for example). Basically, Figure 9 illustrates how the doctor can obtain an ET from the EAC to access the data on the cloud agent of an unconscious patient who has already registered for this emergency protocol (step 1) and provided emergency consent (step 2). Assume a doctor is registered with DID-B (step 3) and the emergency-doctor role (step 4), he has found an unconscious patient and collected his DID-A (step 5). He then invokes the EAC (step 6) to access the EHR of DID-A patient. The IMC and RPMC transactions on the EC, triggered by the EAC, indicate that the patient with DID-A has consented to emergency access (step 7) and that the doctor with DID-B has an emergency-doctor role (step 8). The EAC then writes a transaction on the EC that states that *"doctor with DID-B is authorized to access data of patient with DID-A"* (step 9). The doctor with DID-B takes this ET payload, puts it into a JWT, and adds a signature with the private key associated with their DID-B. This creates an ET that is used by the doctor to the cloud agent (step 11) for accessing EHR data of the patient (step 12). Note that several events are registered by the AACC, from the patient signup to emergency protocol, to the log ET generation and access granted by the cloud agent to the EHR (step 13), for audit and accountability.

To avoid making the figure too long, several sub-steps that are part of the normal EHR sharing workflow (Figure 8) are omitted. For example, before step 12, the doctor's wallet must resolve, thanks to the IMC, the patient's DID-A in the patient's endpoint (which is the cloud agent) where the patient's EHR data is stored, and the patient's public key for later authenticating the cloud agent. Another example is the verification of the ET by the cloud agent, which requires (1) verifying the doctor's signature in the ET by resolving DID-B through the IMC, and obtaining the related public key; (2) verifying that the ET payload matches the payload found on the EC as a transaction written by the EAC.

We also include an objection mechanism where the EAC notifies the patient when the ET payload is generated (step 10). A conscious patient would be able to see such a notification from their wallet, and could object to it if

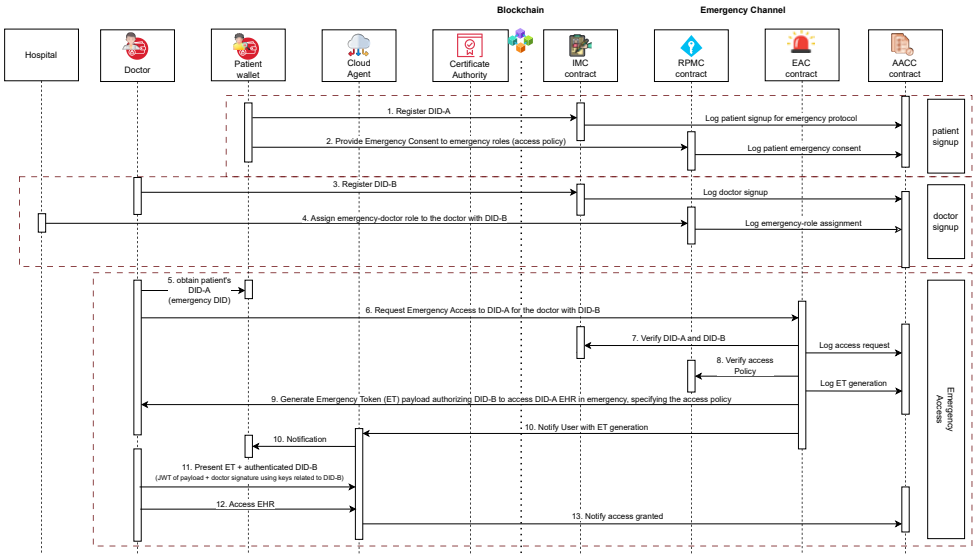

Figure 9: Emergency workflow.

they wanted to. An unconscious patient would not be able to opt-out immediately, so access will happen. However, even afterward, the patient will still be able to see it in their wallet notifications (step 11) and Emergency Channel (AACC) logs and can still pursue it in cases of fraud or unauthorized access.

4.3. Challenges and limitations

While the proposed architecture offers significant benefits for secure and interoperable healthcare data management, its practical implementation, particularly at scale, faces certain challenges and limitations inherent in the foundational technologies employed. These relate primarily to the current maturity and performance characteristics of SSI frameworks and the operational and governance aspects of smart contracts.

Starting with SSI frameworks, scalability can be a bottleneck, especially when it comes to high-volume credential issuance. For instance, performance tests conducted on the widely used SSI stack Hyperledger Aries Cloud Agent Python (ACA-Py) with AnonCreds credentials and Hyperledger Indy as the blockchain, reported significant challenges under load. One documented test took approximately 4 hours to attempt issuing 18,000 credentials, with over

2,500 failures, and observed a credential issuance rate that dropped from an initial three credentials per second to less than one per second after the first hour [35]. While performance improvements are actively being pursued through developments like Aries Askar, which aims to enhance cryptographic operations [36], these findings suggest that current SSI stacks may require further optimization and maturation to efficiently handle the demands of large-scale healthcare systems issuing credentials to potentially millions of users and devices.

The use of smart contracts, while enabling automation and trust, also introduces considerations. Depending on the underlying blockchain platform, factors such as energy consumption due to data redundancy between nodes, transaction costs (gas fees), and potential latency can impact operational efficiency and sustainability at scale [37]. Furthermore, the legal status and enforceability of smart contracts are still an evolving area globally, and requires careful navigation. The implementation of such systems requires clear governance frameworks and compliance with new or evolving regulations surrounding blockchain technology and automated agreements, particularly in the highly regulated healthcare sector [38]. Addressing these technical, legal, and regulatory challenges is essential for the successful large-scale deployment of smart contract-based access control mechanisms.

5. Implementation and Validation

In this section, we implement and validate our proposed secure smart contract-based access control mechanism. To do that, we explain the implementation, the performance tests, and the results obtained.

5.1. Implementation

First, we work with the foundation defined in our previous article [13]. Therefore, we have Hyperledger Fabric as the permissioned DL network that acts as the verifiable data registry. With the newly defined model, we leverage the channels available in Fabric, creating channels with varying access granularities, including inter-organizational and intra-organizational access, as well as emergency cases. Moreover, we utilize the Veramo library for implementing the wallet application, which includes the edge agent and the cloud agent. In this context, we highlight that this article focuses on the smart contract-based access control mechanism and the defined role model.

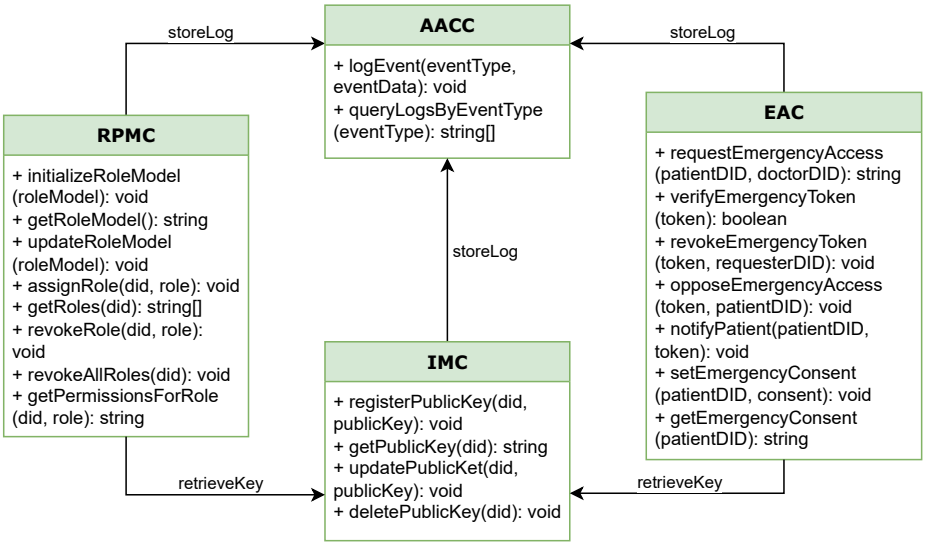

Figure 10: Smart contracts specifications.

Following the design presented in Section 4.2, Figure 10 shows the specification of each contract and the specific connections created between them. Starting with IMC, this contract implements the methods required to manage public keys and DIDs. It contains the functionality to register, receive, update, and delete public keys. The IMC is used in cases where the user identity needs to be verified, for instance, to see the signer of a VC. This contract is connected to the AACC to store the logs of the new identities added. Next, the RPMC contains all the logic for role model management. This contract allows the upload of the role model used in each organization. By default, the role model used is the one defined in this article. In addition, the RPMC implements the methods used for registering new healthcare professionals in the hospitals or clinical organizations, and for querying such roles and permissions by patients to grant access to their EHR. Additionally, the RPMC implements revocation procedures to eliminate the role when the professional leaves the healthcare organization. Finally, this contract is connected to the IMC to check the identities accepted on the platform, and to the AACC to register logs about the processes performed.

The EAC contract implements the emergency functionality. For this, it includes methods to register patient consent for accessing their EHR in an

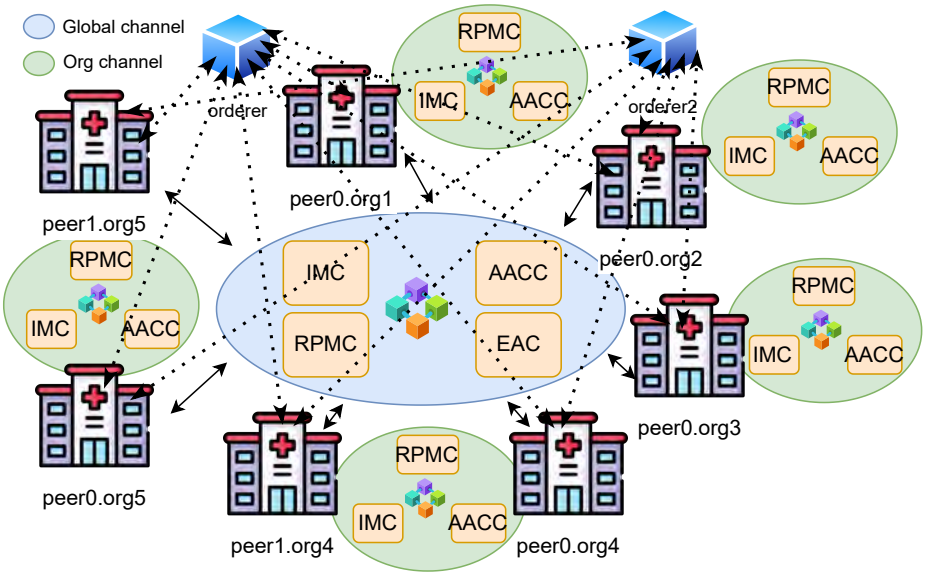

Figure 11: Test network deployed.

emergency, to provide authorized healthcare professionals, and to create an emergency token that the healthcare professional sends to the patient's cloud agent to access the necessary information. This contract also establishes contact with the IMC to obtain the patient's identity information and with the AACC to document all steps taken with the patient's data. Finally, we will present the AACC. It manages all the logs produced in the rest of the SCs. This contract allows the write and read requests for auditing purposes.

5.2. Performance Testbed

After explaining the smart contract specification, we establish a multi-organization deployment for realistic testing and feasibility analysis of the results. In Figure 11, we show the testbed we have composed with five different organizations, alternating single and multi-peer organizations, and two orderers (nodes found in Hyperledger Fabric that order the transactions in blocks and distribute them among DL peers) to balance the traffic and requests from all the organizations.

For the feasibility analysis, we set up two different test experiments:

- Stress test: The efforts are intended to determine the system threshold.

Table 2: Summary of the performance tests.

Tools	Environment	Deployment	Tests	Channels	Metrics
• Hyperledger Fabric V2.5.10 • Hyperledger Caliper	Ubuntu Server • 12CPUs • 32GB RAM • 120GB Disk	• 5 Orgs • 7 Nodes • 2 Orderers	Stress (10,000T) • 1,000TPS Linear (350,000 Transactions) • 50 to 500TPS	• Multi-org • Single-org	• Success rate • Failure rate • Min Latency • Max Latency • Avg Latency • Throughput

The idea is to determine the maximum number of Transactions Per Second (TPS) supported by the solution with a 100% success rate. The total number of transactions we test is 10,000.

- Linear test: This test examines the behavior of the system under normal conditions. In this case, we perform the execution at a linear rate of TPS, from a lower to a higher number of transactions. The total number of transactions is 350,000.

The tool used for testing is Hyperledger Caliper [39], a DL performance benchmarking tool that measures and evaluates the performance of ledger implementations using customizable test cases. It provides metrics such as transaction throughput, latency, resource utilization, and success rate for comparative analysis. For our study, we define several metrics: Successful transactions, Failed transactions, Maximum Latency (seconds), Minimum Latency (seconds), Average Latency (seconds), and Transactions per Second (TPS). We perform both stress and linear tests for the multi-org channel (consisting of 7 DL nodes) and the private channel established in Org5 (2 DL nodes), as we can see in Figure 11. Finally, an Ubuntu Server VM with 12 CPUs, 32GB RAM, and 120GB storage is used to deploy the network and run the tests. Table 2 summarizes the tools, environment, deployment, tests, etc. of the performance tests.

5.3. Results analysis

First, we run the stress and linear tests for the multi-org channel. Figure 12 shows the results obtained. The left-hand chart shows the average latency during all executions and operations, and the right-hand chart shows the

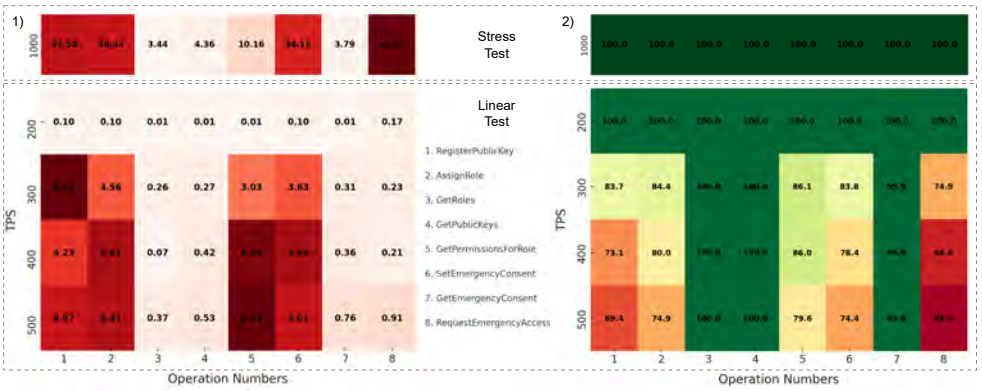

Figure 12: Multi-org channel tests. 1) Average Latency (ms), 2) Success Rates (%).

success rate experienced in the test. As an important detail, the first row of both charts in Figure 12 represents the stress test, which involves 10,000 transactions. The following rows represent different linear tests we ran using different TPS. With a total number of 350,000 transactions, we ran tests with 100 to 200TPS, 100 to 300TPS, 100 to 400TPS, and 100 to 500TPS.

To begin with, the read operations executed are "GetRoles", "GetPublicKey", "GetPermissionsForRole", and "GetEmergencyConsent". Aditionally, the write operations are "RegisterPublicKey", "AssignRole", "SetEmergencyConsent", and "RequestEmergencyAccess". These operations represent the most frequent operations triggered in the platform.

As a general conclusion, we perform both write and read operations, revealing significant differences between them. For write operations, when we use a 300TPS send rate, the success rates and latency are worse. These operations store data in the ledger and require more resources to execute. However, this is not a problem because write operations are rarely triggered, e.g. when a user is first registered, or when a public key/role needs to be updated. In contrast, read operations are executed correctly and with excellent latency at high send rates. This fact indicates that the platform will correctly support daily transactions, as most read operations are performed between patients and healthcare participants. There is one exception to the "GetPermissionsForRole" operation. It requires two queries to the ledger, one to read the user role and another to check the permissions of the role.

Analyzing the stress test, we got a success rate of 100% for the 10,000 transactions and 1,000TPS with three Caliper workers in parallel sending

them. Nevertheless, if we increase the number of transactions or the number of Caliper workers, the network becomes saturated, and the transactions are dropped. If we also increase the TPS, the network crashes, and 1,000TPS is set as the maximum send rate. The latency results are high because the network cannot process all transactions in parallel. However, this situation is practically impossible in a realistic environment, only for a catastrophic situation where many different users need health data. Even so, the read operations are performed in 3 seconds, which is a reasonable value for a petition response.

Regarding the linear tests performed, we can check that the send rate is proportional to the number of successful transactions. As we increase the send rate, the network becomes more saturated, and the number of dropped transactions increases. By default, the Hyperledger Fabric network has a 60-second time limit for transaction processing. If the transaction is not processed within the specified time, it is dropped, and the transaction fails. This is the reason the transactions fail, not that the transaction was processed incorrectly. For a linear rate of 100 to 200TPS, we can conclude that the platform works successfully with 100% success rate and very low latency.

For the local channels, we do not provide the results to avoid redundant information. The results are better since there is only one organization in the channel, so consensus is not necessary. Finally, we have different alternatives to improve these results when the platform is implemented in a production environment. Currently, each operation performed calls the AACC to create a log. This increases the complexity of the chaincode logic and the time to wait for the successful storing of the created log. There is an event listener component that can be included in the network architecture. This listener receives every event generated by the rest of the chaincodes and can call the AACC to generate that log. This component eliminates complexity and allows the logs to be created asynchronously, in a different procedure from the operation being executed. For the scope of this research article, the event listener has not been implemented, but will be included in future work.

5.4. Validating the solution as a realistic solution for a country scale

After presenting the results of the various tests performed, we need to validate that the solution can be scaled to a realistic environment. To do that, we use the definition of use cases provided in Section 3.4, which was used to validate the proposed role model. Thanks to such a description, we can extract the frequency of execution for the operations defined in the smart

contracts. Moreover, we get important input from the workflows presented in Figures 8 and 9. In this context, we classify the SC operations according to their execution frequency:

- Registrations: registerPublicKey (IMC), deletePublicKey (IMC), initializeRoleModel (RPMC), assignRole (RPMC), setEmergencyConsent (EAC).
- Updates/Revocations: updatePublicKey (IMC), updateRoleModel (RP MC), revokeRole (RPMC), revokeAllRoles (RPMC), revokeEmergencyToken (EAC), and queryLogsByEventType (AACC).
- Emergencies: requestEmergencyAccess (EAC), verifyEmergencyToken (EAC), opposeEmergencyToken (EAC), opposeEmergencyAccess (EAC), getEmergencyConsent (EAC).
- Daily: getPublicKey (IMC), getRoles (RPMC), getPermissionsForRole (RPMC), and logEvent (AACC).

In addition, we research a realistic use case. We chose the Spanish healthcare domain to validate our solution. Spain has approximately 49 million people and 845 hospitals, with an average of 58,333 people per hospital [40]. Moreover, according to studies conducted in 2017 [41], each person has about seven annual medical consultations. The note is a bit outdated, and the COVID-19 pandemic has likely increased this number. Therefore, we can assume that each person will have 14 yearly medical consultations for this study. Regarding emergencies, a study indicates that 3.7% of Spanish workers suffered an accident in 2021 [42]. For the study, we can extrapolate these accident numbers to the entire Spanish population. Finally, the clinical trials produced in 2024 were 843 in Spain [43].

With these numbers, we can estimate the total number of transactions and the TPS associated:

- Registrations: For this study, we can establish 1 time for the registration. With forty-nine million people, we have 49,000,000 people $\times$ 5 operations = 245,000,000 transactions, which translates to 7.76TPS, considering that all transactions are executed in one year. Still, we know that these transactions will be triggered within a timeframe longer than one year.

- Updates/Revocations: In this case, we stipulate that updates and revocation procedures are performed annually. Having 49,000,000 people $\times$ 6 operations = 294,000,000 transactions, translated to 9.32TPS.

- Emergencies: These transactions are performed annually to address emergencies. We extract four actors interacting in these procedures from the emergency use case presented in Section 3.4. Therefore, we calculate 49.000.000 people $\times$ 3.7% = 1,813,000 people suffering an accident $\times$ 4 actors $\times$ 5 operations assigned for emergencies = 43,512,000 transactions, which are 1.14TPS.

- Daily: These transactions include the daily interactions between patients and healthcare professionals. Here, we have the local channels associated with each healthcare organization. Consequently, the numbers are 58,333 people per hospital $\times$ 14 medical consultations per year = 816.666 appointments. From the patient sample lifecycle use case presented in Section 3.4, we have six actors interacting in medical consultations and four transactions (daily transactions) per actor: 816,666 appointments $\times$ 6 actors $\times$ 4 operations = 19,600,000 transactions, establishing 0.62TPS in the local-org channel.

- Clinical trials: As we have investigated, Spain had 843 clinical trials in 2025. These trials have different phases, in which people participate. For our study, we can assume that all trials were in the final phase and involved 1,000 participants. Moreover, we conclude from the clinical trial use case (Section 3.4) that 6 actors interact in this flow. Then, we have 843 trials $\times$ 1000 people $\times$ 6 actors = 5,058,000 interactions. Regarding transactions, this use case works with daily transactions so that 5,058,000 $\times$ 4 operations = 20,232,000 transactions, translated to 0.64TPS in the global channel.

With these calculations, we obtain about 18,86TPS for the multi-org channel and 17,72TPS for the local-org channel, knowing that registrations and updates/revocations transactions can be distributed among both channels. These results demonstrate that our solution meets the requirements for working in a realistic country healthcare domain, as we achieved 100% of successes for a send rate of 200TPS. Furthermore, we have researched other works in the literature working with DL technologies [18, 44, 45, 46], and

they performed smaller tests reaching 100/200TPS with simpler smart contract models and DL networks. Therefore, we can successfully demonstrate that our solution can fulfill the purpose for which it was designed.

6. Conclusions

This proposal successfully extends our previously proposed health data management framework [13] by integrating an advanced access control mechanism based on a comprehensive definition of a role model for the patient's EHR. By implementing smart contracts within the Hyperledger Fabric environment, we have enhanced the system's ability to effectively enforce role-based access policies. This advancement not only strengthens the overall security and auditability of the framework, but also gives patients greater control over their health data and automates the selective disclosure of EHR data based on defined roles and thus the needs of different verifiers, improving the privacy and data minimization aspects of health data sharing. Despite the current challenges of scalability of certain SSI stacks and blockchains, we demonstrate through our performance measures that a large scale blockchain-based SSI is possible. As such, the proposed approach is a realistic and promising solution for a more transparent, auditable, and patient-centric healthcare system satisfying several use cases, from the most common to emergencies.

Future works will focus on improving the integration between the SSI wallet exchanges and the blockchain-defined access control model. We believe that a seamless integration between both would eventually lead to a direct wallet-to-wallet exchange of health data governed by blockchain-defined policies that are enforced through smart contracts. In addition, with the rapid evolution of AI models, we believe we can make use of a local AI agent implemented at the wallet level that acts as a recommendation and privacy advisor, helping wallet owners structure their data presentations and select the data to share with other actors based on many rules and policies defined on the blockchain (such as our RBAC model) and outside of it.

Acknowledgments

This work has been partially funded by the strategic project CDL-TALEN TUM from the Spanish National Institute of Cybersecurity (INCIBE) by the

Recovery, Transformation and Resilience Plan, Next Generation EU. Furthermore, this work benefited from State aid managed by the Agence Nationale de la Recherche under the France 2030 programme, reference ANR-22-PESN-0006, project TracIA. It is also partly supported by the chair Values and Policies of Personal Information, Institut Mines-Telecom, France, and International Alliance for Strengthening Cybersecurity and Privacy in Healthcare (CybAlliance, Project no. 337316).

Author's Contribution

All authors contributed to the study conception and design. Software development, investigation, and analysis were performed by Antonio López Martínez and Montassar Naghmouchi. The first draft of the manuscript was written by Antonio López Martínez and Montassar Naghmouchi. Manuel Gil Pérez, Antonio Ruiz Martínez, Maryline Laurent, and Joaquín García rewrote some parts of the manuscript and made several revisions to the entire manuscript. All authors read and approved the final manuscript.

References

[1] B. Al-Haimi, F. Ali, F. Hujainah, Digital transformation in healthcare: Impact on organizations' strategies, future landscape, and required skills, Springer Nature Singapore, 2023, pp. 61–74. doi:10.1007/978-981-99-8572-2_3.

[2] T. McGhin, K.-K. R. Choo, C. Z. Liu, D. He, Blockchain in healthcare applications: Research challenges and opportunities, Journal of Network and Computer Applications 135 (2019) 62–75. doi:10.1016/j.jnca.2019.02.027.

[3] A. López Martínez, M. Gil Pérez, A. Ruiz-Martínez, A comprehensive review of the state-of-the-art on security and privacy issues in healthcare, ACM Computing Surveys 55 (12) (2023) 1–38. doi:10.1145/3571156.

[4] A. Ghani, A. Zinedine, M. E. Mohajir, A patient-centric blockchain-based system for access management in telehealth and telemedicine domain, in: 2023 7th IEEE Congress on Information Science and Technology, 2023, pp. 669–674. doi:10.1109/CiSt56084.2023.10409917.

[5] S. S. Mahadik, P. M. Pawar, R. Muthalagu, N. R. Prasad, S.-K. Hawkins, D. Stripelis, S. Rao, P. Ejim, B. Hecht, Digital privacy in healthcare: State-of-the-art and future vision, IEEE Access 12 (2024) 84273–84291. doi:10.1109/ACCESS.2024.3410035.

[6] K. P. Kalita, D. Boro, D. Kumar Bhattacharyya, Designing efficient patient-centric smart contracts for healthcare ecosystems with access control capabilities, Security and Privacy 7 (6) (2024) e427. doi:10.1002/spy2.427.

[7] Energy, Commerce, What we learned: Change healthcare cyber attack, accessed on 04/05/2025 (2024).
URL https://energycommerce.house.gov/posts/what-we-learned-change-healthcare-cyber-attack

[8] IntrapriseHealth, Cybersecurity nightmares: The cost of healthcare cyberattacks in 2024, accessed on 04/05/2025 (2024).
URL https://intraprisehealth.com/the-cost-of-cyberattacks-in-healthcare/

[9] European Union, Regulation (EU) 2016/679 of the European Parliament and of the Council, accessed on 04/05/2025 (2016).
URL https://eur-lex.europa.eu/eli/reg/2016/679/oj

[10] U.S. Department of Health and Human Services, Health insurance portability and accountability act of 1996, accessed on 04/05/2025 (1996).
URL https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations

[11] S. K. Roy, Decentralized identity and access management (IAM) and self-sovereign identity, International Journal of Research in Engineering, Science and Management 6 (12) (2023) 201–210.

[12] C. Mazzocca, A. Acar, S. Uluagac, R. Montanari, P. Bellavista, M. Conti, A survey on decentralized identifiers and verifiable credentials, IEEE Communications Surveys & Tutorials (2025) 1–32doi:10.1109/COMST.2025.3543197.

[13] A. L. Martínez, M. Naghmouchi, M. Laurent, J. Garcia-Alfaro, M. G. Pérez, A. R. Martínez, P. Nespoli, Empower healthcare through a self-sovereign identity infrastructure for secure electronic health data access (2025). arXiv:2501.12229.
URL https://arxiv.org/abs/2501.12229

[14] Antonio López, SSI Smart Contracts for Healthcare, Accessed on 04/05/2025 (2025).
URL https://github.com/LopeeezOne/ssi-smart-contracts

[15] H. L. Pham, T. H. Tran, Y. Nakashima, A secure remote healthcare system for hospital using blockchain smart contract, in: 2018 IEEE Globecom Workshops, 2018, pp. 1–6. doi:10.1109/GLOCOMW.2018.8644164.

[16] G. G. Dagher, J. Mohler, M. Milojkovic, P. B. Marella, Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology, Sustainable Cities and Society 39 (2018) 283–297. doi:10.1016/j.scs.2018.02.014.

[17] E.-Y. Daraghmi, Y.-A. Daraghmi, S.-M. Yuan, MedChain: A design of blockchain-based system for medical records access and permissions management, IEEE Access 7 (2019) 164595–164613. doi:10.1109/ACCESS.2019.2952942.

[18] S. Tanwar, K. Parekh, R. Evans, Blockchain-based electronic healthcare record system for healthcare 4.0 applications, Journal of Information Security and Applications 50 (2020) 102407. doi:10.1016/j.jisa.2019.102407.

[19] A. Khatoon, A blockchain-based smart contract system for healthcare management, Electronics 9 (1) (2020) 94. doi:10.3390/electronics9010094.

[20] A. A. Omar, A. K. Jamil, A. Khandakar, A. R. Uzzal, R. Bosri, N. Mansoor, M. S. Rahman, A transparent and privacy-preserving healthcare platform with novel smart contract for smart cities, IEEE Access 9 (2021) 90738–90749. doi:10.1109/ACCESS.2021.3089601.

[21] M. U. Chedallurai, D. S. Pandian, D. K. Ramasamy, A blockchain based patient centric electronic health record storage and integrity manage-

ment for e-health systems, Health Policy and Technology 10 (4) (2021) 100513. doi:10.1016/j.hlpt.2021.100513.

[22] D. T. Harrell, M. Usman, L. Hanson, M. Abdul-Moheeth, I. Desai, J. Shriram, E. de Oliveira, J. R. Bautista, E. T. Meyer, A. Khurshid, Technical design and development of a self-sovereign identity management platform for patient-centric health care using blockchain technology, Blockchain in Healthcare Today 5 (2022) 1–12.

[23] V. Mantzana, M. Themistocleous, Z. Irani, V. Morabito, Identifying healthcare actors involved in the adoption of information systems, European Journal of Information Systems 16 (1) (2007) 91–102.

[24] ShelterForce, The U.S. health care system: Players and incentives, accessed on 04/05/2025 (2020).
URL https://shelterforce.org/2020/10/23/the-u-s-health-care-system-players-and-incentives/

[25] A. López Martínez, M. Gil Pérez, A. Ruiz-Martínez, A comprehensive model for securing sensitive patient data in a clinical scenario, IEEE Access 11 (2023) 137083–137098. doi:10.1109/ACCESS.2023.3338170.

[26] Health Level Seven International, HL7 version 2.9 standard, Accessed on 04/05/2025 (2019).
URL http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185

[27] Health Level Seven International, HL7 version 3 standard: Introduction to the HL7 version 3 RIM, Accessed on 04/05/2025 (2013).
URL http://www.hl7.org/implement/standards/product_brief.cfm?product_id=186

[28] Health Level Seven International, Fast healthcare interoperability resources (FHIR), Accessed on 04/05/2025 (2024).
URL https://hl7.org/fhir/

[29] Health Level Seven International, HL7 version 3 standard: Clinical document architecture, release 2, Accessed on 04/05/2025 (2005).
URL http://www.hl7.org/implement/standards/product_brief.cfm? product_id=7

[30] J. C. Mandel, D. A. Kreda, K. D. Mandl, I. S. Kohane, R. B. Ramoni, SMART on FHIR: a standards-based, interoperable apps platform for electronic health records, Journal of the American Medical Informatics Association 23 (5) (2016) 899–908. doi:10.1093/jamia/ocv189.

[31] M. Ayaz, M. F. Pasha, M. Y. Alzahrani, R. Budiarto, D. Stiawan, The fast health interoperability resources (FHIR) standard: Systematic literature review of implementations, applications, challenges and opportunities, JMIR Medical Informatics 9 (7) (2021) e21929. doi:10.2196/21929.

[32] A. Ubale Swapnaja, G. Modani Dattatray, S. Apte Sulabha, Analysis of DAC MAC RBAC Access Control based Models for Security, International Journal of Computer Applications 104 (5) (2014) 6–13.

[33] I. A. Omar, R. Jayaraman, K. Salah, I. Yaqoob, S. Ellahham, Applications of blockchain technology in clinical trials: review and open challenges, Arabian Journal for Science and Engineering 46 (4) (2021) 3001–3015.

[34] L. Hang, C. Chen, L. Zhang, J. Yang, Blockchain for applications of clinical trials: Taxonomy, challenges, and future directions, IET Communications 16 (20) (2022) 2371–2393.

[35] P. Wen, Test results performed with hyperledger aries indy, accessed on 02/05/2025 (2022).
URL https://github.com/lissi-id/acapy-load-test-results/blob/main/AcaPy_0-7-4/Endurance_Test/04-0_7_3_indy-200rpm/report-test-results-0-END.pdf

[36] H. Foundation, Askar, accessed on 02/05/2025 (2025).
URL https://github.com/openwallet-foundation/askar

[37] A. O. Bada, A. Damianou, C. M. Angelopoulos, V. Katos, Towards a green blockchain: A review of consensus mechanisms and their energy consumption, in: 2021 17th international conference on distributed computing in sensor systems (DCOSS), IEEE, 2021, pp. 503–511.

[38] M. Finck, Blockchain regulation and governance in Europe, Cambridge University Press, 2018.

[39] Hyperledger caliper, the blockchain benchmarking tool., https://www.lfdecentralizedtrust.org/projects/caliper.

[40] M. of Healthcare, Hospitals, accessed on 04/05/2025 (2024).
URL https://www.sanidad.gob.es/ciudadanos/hospitales.do?tipo=hospital

[41] M. Pacientes, Spain above the european average in doctor visits, according to eurostat, accessed on 04/05/2025 (2017).
URL https://www.medicosypacientes.com/articulo/espana-por-encima-de-la-media-europea-en-visitas-al-medico-segun-eurostat/

[42] N. I. of Statistic, Active population survey, accessed on 04/05/2025 (2021).
URL https://ine.es/prensa/epa_2020_m.pdf

[43] S. Millan, The pharmaceutical industry increases clinical trials by 10% and surpasses pre-pandemic figures., accessed on 04/05/2025 (2025).
URL https://cincodias.elpais.com/companias/2025-01-15/la-industria-farmaceutica-eleva-un-10-los-ensayos-clinicos-y-supera-las-cifras-pre-pandemia.html

[44] G. Al-Sumaidaee, R. Alkhudary, Z. Zilic, A. Swidan, Performance analysis of a private blockchain network built on hyperledger fabric for healthcare, Information Processing & Management 60 (2) (2023) 103160.

[45] M. Hasnain, F. R. Albogamy, S. S. Alamri, I. Ghani, B. Mehboob, The hyperledger fabric as a blockchain framework preserves the security of electronic health records, Frontiers in Public Health 11 (2023) 1272787.

[46] A. Nedaković, A. Hasselgren, K. Kralevska, D. Gligoroski, Hyperledger fabric platform for healthcare trust relations—proof-of-concept, Blockchain: Research and Applications 4 (4) (2023) 100156.