



Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific
Research
University of Mohamed Khider - BISKRA
Faculty of Exact Sciences, Natural Sciences and Life
Computer Science Department

Order Number:

Thesis

Presented to obtain the diploma of academic Master in

Computer Science

Option: **Networks and Technologies of Information and
Communication**

Secure and Automated Algerian Hospitals

By:

DJOUAMA Salah Eddine

KHADRAOUI Sabrine

LAMOURI Ahmed Abdesselam

Defended the 17/06/2025, in front of the jury composed of:

Dr.ZERNADJI Tarek	...	President
Dr. SAHRAOUI Somia	MCA	Supervisor
Dr.DAKHIA Abdelhafid	...	Examiner

College Year: 2024 - 2025



République Algérienne Démocratique
et Populaire
Ministère de L'Enseignement Supérieur et de La
Recherche Scientifique
Université Mohamed Khider - BISKRA
Faculté des Sciences Exactes, des Sciences de La
Nature et de La Vie
Département d'Informatique

numéro d'Ordre:

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours: **Réseaux et technologies de l'information et
Communication**

Les Hôpitaux Algériens Sécurisés Et Automatisés

Par:

DJOUAMA Salah Eddine

KHADRAOUI Sabrine

LAMOURI Ahmed Abdesselam

Soutenu le 17/06/2025, devant le jury composé de:

Dr.ZERNADJI Tarek	...	Président
Dr. SAHRAOUI Somia	MCA	Rapporteur
Dr.DAKHIA Abdelhafid	...	Examineur

Année Universitaire: 2024 - 2025

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Acknowledgements

First and foremost, we thank **God** for granting us the courage and determination to complete this project.

We would like to express our sincere gratitude to our thesis supervisor, **Dr.Sahraoui Samia**, for her guidance, advice, assistance, and follow-up throughout the project.

We also thank the members of the jury for their evaluation of our work.

Finally, we extend our appreciation to the computer science professors for their efforts.

Dedication

We dedicate this work with deep gratitude to our families, for their unconditional love, unwavering support, and constant encouragement throughout our journey.

To our friends, who motivated us, inspired us, and shared both the challenges and the moments of joy.

To every single person who stood by us, offered help, guidance, or even a kind word during difficult times your presence made a real difference.

This project is the result of collective support, and we are sincerely thankful to all of you.

Abstract

In recent years, the integration of advanced technologies in the healthcare sector has opened new horizons for secure, efficient, and intelligent patient monitoring systems. This project proposes the integration of Blockchain technology and Wireless Body Area Networks (WBANs) to bring innovative solutions to the healthcare domain.

The system uses Blockchain to ensure the security, integrity, and traceability of medical data, while also eliminating issues such as identity theft, denial of truth, and enabling legal action based on verifiable evidence. And WBAN technology is employed for continuous, permanent, and real-time patient monitoring, enabling rapid medical intervention when necessary.

The thesis provides a comprehensive background on eHealth systems, Blockchain architecture (with a focus on Hyperledger Fabric), and WBANs, highlighting their roles and challenges in the medical domain. Furthermore, the proposed solution is detailed through system design diagrams, including use case, class, and sequence diagrams. Implementation aspects are discussed, along with the tools and environment used. Evaluation is conducted using specialized metrics and tools to validate the effectiveness of the system in detecting and mitigating threats during medical data streaming. The results demonstrate the potential of combining these technologies to significantly improve healthcare data security and trustworthiness.

Keywords: eHealth, Blockchain, WBAN (Wireless Body Area Network), Medical Data Security, Identity Theft Prevention, Real Time Patient Monitoring, Hyperledger Fabric, Data Integrity.

Résumé

Au cours des dernières années, l'intégration des technologies avancées dans le secteur de la santé a ouvert de nouvelles perspectives pour des systèmes de surveillance des patients plus sûrs, efficaces et intelligents. Ce projet propose l'intégration de la technologie Blockchain et des réseaux corporels sans fil (WBAN) afin d'apporter des solutions innovantes dans le domaine de la santé.

Le système utilise la Blockchain pour garantir la sécurité, l'intégrité et la traçabilité des données médicales, tout en éliminant des problèmes tels que le vol d'identité, le déni de vérité, et en permettant des poursuites judiciaires basées sur des preuves vérifiables. Et la technologie WBAN est utilisée pour une surveillance continue, permanente et en temps réel des patients, permettant ainsi une intervention médicale rapide lorsque cela est nécessaire.

Le rapport présente un aperçu complet des systèmes de santé électronique, de l'architecture de la Blockchain (en mettant l'accent sur Hyperledger Fabric) et des WBANs, en soulignant leurs rôles et défis dans le domaine médical. La solution proposée est ensuite illustrée à travers des diagrammes de conception : cas d'utilisation, classes et séquences. Les aspects liés à l'implémentation, aux outils et à l'environnement de développement sont également discutés. Une évaluation est menée à l'aide de métriques et d'outils spécialisés afin de valider l'efficacité du système dans la détection et l'atténuation des menaces pendant le flux des données médicales. Les résultats démontrent le potentiel de ces technologies à améliorer significativement la sécurité et la fiabilité des données de santé.

Mot clés: e-Santé, Blockchain, WBAN (Réseau corporel sans fil), Sécurité des données médicales, Prévention du vol d'identité, Surveillance en temps réel, Hyperledger Fabric, Intégrité des données.

الملخص

في السنوات الأخيرة، أدى دمج التقنيات المتقدمة في قطاع الرعاية الصحية إلى فتح آفاق جديدة نحو أنظمة مراقبة المرضى بشكل آمن وفعال وذكي. يقترح هذا المشروع دمج تكنولوجيا البلوكشين مع شبكات الجسم اللاسلكية (WBAN) لتقديم حلول مبتكرة في المجال الصحي. يُستخدم نظام البلوكشين لضمان أمن البيانات الطبية وسلامتها وإمكانية تتبعها، بالإضافة إلى القضاء على مشاكل مثل سرقة الهوية وإنكار الحقيقة، مع إتاحة اتخاذ إجراءات قانونية بناءً على أدلة قابلة للتحقق. من ناحية أخرى، تُستخدم تكنولوجيا WBAN في المراقبة المستمرة والدائمة وفي الوقت الحقيقي لحالة المرضى، مما يسمح بتدخل طبي سريع عند الحاجة. يعرض التقرير خلفية شاملة حول أنظمة الصحة الإلكترونية، وبنية البلوكشين مع التركيز على Hyperledger Fabric، وشبكات WBAN، مع تسليط الضوء على أدوارها وتحدياتها في المجال الطبي. كما يتم تقديم الحل المقترح من خلال مخططات تصميم النظام مثل مخطط الاستخدام، ومخطط الأصناف، ومخطط التسلسل. كما يُناقش الجوانب المتعلقة بالتنفيذ، والأدوات، وبيئة التطوير. وتم إجراء تقييم باستخدام أدوات ومقاييس متخصصة للتحقق من فعالية النظام في اكتشاف التهديدات والتخفيف من آثارها أثناء بث البيانات الطبية. وتُظهر النتائج الإمكانيات الكبيرة لدمج هذه التقنيات في تعزيز أمن وموثوقية البيانات الصحية.

الكلمات المفتاحية: الصحة الإلكترونية، البلوكشين، شبكات WBAN (شبكات الجسم اللاسلكية)، أمن البيانات الطبية، منع سرقة الهوية، المراقبة اللحظية لحالة المريض، Hyperledger Fabric، سلامة البيانات.

Contents

Abstract

Résumé

Abstract in Arabic

List of Figures V

List of Tables VI

1 General Introduction 1

1.1 Context 1

1.2 Problematic and Motivation 2

1.3 The Project's Purpose 2

1.4 Structure of the Dissertation 2

List of Abbreviation 1

2 Background 4

2.1 Introduction 4

2.2 Electronic Health (E-health) 4

2.2.1 Definition 4

2.2.2 Benefits and Importance 5

2.2.3 Real World Applications 5

2.2.3.1 Mhealth Platform for Diabetes Management 5

2.2.3.2 KRY and Doktor.se 5

2.2.3.3 Ehealth Nigeria 6

2.2.4 Future of E-health 6

2.2.4.1 Integration of Artificial Intelligence (AI) and Machine Learning 6

2.2.4.2 Expansion of Telemedicine and Remote Care 6

2.2.4.3 Smart Hospitals and Digital Twins 6

2.2.4.4 Blockchain Integration 6

2.3 Blockchain Technology 7

2.3.1 Definition 7

2.3.2	History and Evolution	7
2.3.3	Types	8
2.3.4	Blockchain Vs Traditional Databases	10
2.3.5	How It Works	11
2.3.6	Architecture	12
2.3.6.1	Data Layer	12
2.3.6.2	Network Layer	12
2.3.6.3	Consensus Layer	13
2.3.6.4	Contract Layer	13
2.3.6.5	Application Layer	13
2.3.7	Applications	14
2.3.7.1	In Healthcare	14
2.3.7.2	In Communication	15
2.3.7.3	In Cloud Gaming System	15
2.3.7.4	In Energy Sector	15
2.3.7.5	In Banking	15
2.3.7.6	In Real Estate	15
2.4	Hyperledger Project	15
2.4.1	Hyperledger Fabric	16
2.4.1.1	Hyperledger Fabric Architecture	16
2.4.1.2	How Hyperledger Fabric Works	20
2.5	Wireless Body Area Network	21
2.5.1	Definition	21
2.5.2	The Body Sensor Network Function	22
2.5.3	Wireless Communication Technologies	23
2.5.3.1	ZigBee Technology	23
2.5.3.2	Bluetooth Technology	23
2.5.3.3	Ultrawideband (UWB) Technology	23
2.5.3.4	Wi-Fi—IEEE 802.11	23
2.5.4	WBAN Challenges	23
2.6	Conclusion	24
3	Blockchain Technology in eHealth	25
3.1	Introduction	25
3.2	Relationship between Blockchain Technology and Ehealth	25
3.2.1	Data Security and Privacy	25
3.2.2	Interoperability and Data Sharing	25
3.2.3	Patient Centric Data Control	26
3.2.4	Supports AI and Analytics	26
3.2.5	Legal and Regulatory Compliance	26
3.3	Related works	26

3.3.1	Review on Blockchain Solutions for Healthcare Data Security	26
3.3.2	Review on Security Solutions in WBAN	28
3.4	Conclusion	30
4	Conception and Implementation	31
4.1	Introduction	31
4.2	Proposed Solution	31
4.3	System Diagrams	33
4.3.1	Use Case Diagram	33
4.3.2	Class Diagram	34
4.3.3	Sequence Diagrams	36
4.4	Development Environment	39
4.5	Conclusion	40
5	System Presentation and Evaluation	41
5.1	Introduction	41
5.2	System Presentation	41
5.3	System evaluation	48
5.3.1	Evaluator Tools	48
5.3.1.1	Postman	48
5.3.1.2	Hyperledger Caliper	48
5.3.2	Interpretation Results	48
5.3.2.1	Performance of API	48
5.3.2.2	Performance of Network	50
5.4	Conclusion	53
6	Conclusion and Perspectives	54
	Bibliography	61

List of Figures

2.1	Block structure in a blockchain [18].	7
2.2	Key events in blockchain technology [18].	8
2.3	Types of Blockchain [20].	8
2.4	Difference between types of blockchain [24].	10
2.5	Client server Vs Peer to peer architecture[28].	12
2.6	CAP and FLP theorms [29, 30].	13
2.7	Blockchain Architecture [27].	14
2.8	Hyperledger projects [36].	16
2.9	Transaction flow in Hyperledger Fabric [42].	20
2.10	Wireless body area network scheme [45].	22
2.11	Various medical sensors deployed on the human body [47], [48], [49], [50], [51], [52], [53], [54], [55].	22
4.1	System architecture.	32
4.2	Use case system diagram.	33
4.3	Class system diagram.	35
4.4	Sequence diagram: Admin scenario: Registration scenario.	36
4.5	Sequence diagram: Login scenario.	37
4.6	Sequence diagram: Doctor scenario: Show patients and create prescription.	37
4.7	Sequence diagram: Nurse scenario: Create patient account and search patient.	38
4.8	Sequence diagram: Patient scenario: Search doctors by specialization and show pre- scriptions.	38
4.9	Ubuntu.	39
4.10	VSCode.	39
4.11	Docker.	39
4.12	Node js.	40
4.13	HTML CSS and JavaScript.	40
5.1	Main interface (top).	41
5.2	Main interface (bottom).	42
5.3	Login interface.	42
5.4	Admin interface: Hospital information.	43

5.5	Admin interface: Create doctor account.	43
5.6	Doctor interface: Personal information.	44
5.7	Doctor interface: Update patient medical file.	44
5.8	Doctor Interface: Show Patient Data	45
5.9	Doctor interface: Notifications.	45
5.10	Nurse interface: Personal information.	46
5.11	Nurse interface: Create patient account.	46
5.12	Patient interface: Personal information.	47
5.13	Patient interface: Edit data.	47
5.14	Patient interface: Show options.	47
5.15	Postman.	48
5.16	Hyperledger Caliper.	48
5.17	Update admin details.	49
5.18	Retrieve doctor list.	49
5.19	Hyperledger Caliper results 1.	51
5.20	Hyperledger Caliper results 2.	51
5.21	Hyperledger Caliper results 3.	52
5.22	Hyperledger Caliper Results Graphically.	52

List of Tables

2.1	Comparison between Blockchain and Traditional databases.	10
2.2	Consensus algorithms [18, 41].	19
2.3	WBAN challenges.	24
3.1	Related work on Blockchain applications in Healthcare data security.	28
3.2	Related work on security mechanisms in sensor networks.	29

List of Abbreviation

ICT: Information and Communication Technology.

WBAN: Wireless Body Area Network.

WHO: World Health Organization.

NFT: Non Fungible Token.

CAP: Consistency Availability Partitioning.

FLP: Fisher Lynch Paterson.

MSP: Membership Service Provider.

SNS: Simple Notification Service.

Chapter 1

General Introduction

1.1 Context

The rapid advancement of IoT (Internet of Things) technologies has led to successful and automated applications in several domains [1]. In our country, there is a serious desire to keep up with the development of digital technologies to automate operational processes and ensure data security. Among the sectors, health was particularly sensitive considering the volumes of individual data and healthcare records involved along with the necessity to possess functioning, responsive, and secure patient service in exigent times. The conventional operation of the majority of Algerian hospitals still heavily relies on manual processes, paper records, and limited automation, which have a propensity to lead to inefficiencies, mistakes, and potential security violations. In light of this backdrop, modernization of Algerian hospitals is no longer a choice but imperative.

This project, entitled **”Secure and Automated Algerian Hospitals”** attempts to address these pressing issues by proposing a comprehensive system that enhances data security and operational automation in hospitals. The system relies on two main pillars of technology: blockchain technology for secure management of medical data, and robot automation to assist healthcare professionals as well as enhance monitoring and delivery of services.

Security wise, the system uses a private blockchain network to record and maintain electronic medical records (EMRs) and hospital operations. Blockchain’s decentralized as well as immutable properties ensure assurances that medical data are tamper proof, traceable, and can be accessed only by authorized personnel. By making use of smart contracts, the platform allows a secure identity management and access control mechanism such that only the legitimate parties administrative personnel, doctors, and nurses are allowed to deal with sensitive information as per their function.

Through a focus on the unique needs of Algerian hospitals, this framework establishes a malleable and scalable model of digital transformation for healthcare, making possible a more intelligent, more secure, and safer medical infrastructure.

1.2 Problematic and Motivation

Algerian healthcare facilities are plagued with major data security, efficiency, and quality of care concerns due to legacy systems and manual effort. Sensitive medical data is at risk of being lost, disclosed, and accessed without permission, and staff are bogged down in unnecessary work. Something must be done a safe, automated solution with a defined solution to the local Algerian environment to protect data, automate processes, and support medical staff in real time.

This project is motivated by the goal of revolutionizing Algerian hospitals with secure and intelligent technologies. Through the implementation of blockchain for data protection and autonomous robots for automating tasks, we aim to improve patient care, reduce the workload on medical staff, and ensure data integrity. The project is aligned with the vision of developing smart, efficient, and secure healthcare facilities in accordance with international standards.

1.3 The Project's Purpose

The objective of this work is to design and implement an integrated framework that enhances the security and automation of Algerian hospitals. Specifically, the project aims to:

1. **Secure patient medical data** using a private blockchain system that ensures confidentiality, integrity, and traceability.
2. **Automate routine hospital tasks** through autonomous care assistant robots to support healthcare staff and improve operational efficiency.
3. **Improve access control and surveillance** of hospital facilities using intelligent systems and biometric technologies.
4. **Getting rid of the problem of accusing a doctor and impersonating him** and the ability to follow up legally.
5. **Ensure real time communication and coordination** between systems and agents using IoT.

Through this approach, the project seeks to contribute to the digitalization of the Algerian healthcare system by providing a smart, secure, and scalable solution adjusted to local needs.

1.4 Structure of the Dissertation

The dissertation comprises five chapters each addressing a distinct research component. Here's a chapter breakdown:

- **Chapter 2: Background** establishes a foundation by examining the research and relevant literature and theories, and outlining the technological principles that serve as the supporting framework for the research.
- **Chapter 3: Blockchain Technology in HealthCare** which is the project focus on, and there relationship with each other, also this parte contain some related real examples.

- **Chapter 4: Conception and Implementation** that guarantee a full explanation of the proposed solution using a different diagrams of Unified Modeling Language also Pointing the tools and steps Necessary to achieve it.
- **Chapter 5: System Presentation and Evaluation** presents the findings from the research and some screenshot of system, also results of our system evaluation.
- **Chapter 6: Conclusion and Perspectives** this chapter summarizes the most important research findings and contributions and concludes the main goal of this article, and gives a point of view for future work.

Chapter 2

Background

2.1 Introduction

This chapter provides the foundational knowledge required to understand the core components of our system. It begins by exploring the concept of electronic health (eHealth), its significance, and its potential for transforming healthcare services. The chapter then delves into blockchain technology, highlighting its mechanisms, architecture, and applicability in securing healthcare systems. Furthermore, it introduces the Hyperledger project with a focus on Hyperledger Fabric as a blockchain framework. Lastly, it examines Wireless Body Area Networks (WBANs), their role in real-time health monitoring, and the challenges associated with their implementation.

2.2 Electronic Health (E-health)

2.2.1 Definition

E-health is a broad concept with several definitions, reflecting the many ways digital technologies are used in healthcare. It can refer to electronic health records, telemedicine, mobile health apps, and more. Some define it technically as the use of ICT in health while others see it as a broader transformation in how care is delivered and managed [2]. This variety shows that E-health is both a technological and cultural shift in modern healthcare, but it can be defined as:

- E-health interventions are defined as health services provided through internet-based technologies to improve patient engagement, monitoring, and outcomes [3].
- E-health is broadly considered as the use of information and communication technologies (ICT) in healthcare [4].
- E-health refers to health services and information delivered or enhanced through the internet and related technologies [5].

So we can definitely understand the word 'E-health' with the definition of **World Health Organization** (WHO), the regional office for the eastern Mediterranean, that defines eHealth as

”The cost effective and secure use of information and communications technologies in support of health and health related fields, including health care services, health surveillance, health literature, and health education, knowledge and research” [6].

2.2.2 Benefits and Importance

The benefits and importance of E-health can be summarized in the following points [7, 8, 3]:

- Increases accessibility to healthcare, especially in remote or underserved areas.
- Enables continuous health monitoring through mobile and wearable tech.
- Supports early detection and intervention, reducing hospital admissions.
- Improves workflow efficiency in health systems through digital records.
- Demonstrates cost effectiveness of eHealth compared to traditional care.
- Improves self care and patient autonomy.
- Facilitates real time feedback between patient and doctor.
- Enhances quality of life for chronic patients through proactive management.

2.2.3 Real World Applications

E-Health is transforming healthcare by providing innovative solutions, in the following we mention some of real world solutions:

2.2.3.1 Mhealth Platform for Diabetes Management

The mHealth Platform for Diabetes Management is a mobile application designed to help diabetes patients monitor their health metrics, such as blood glucose levels, activity, and diet. By providing personalized feedback and reminders, it encourages users to make healthier lifestyle choices. The platform has led to improved health behaviours, such as increased physical activity and better dietary habits, which have resulted in better glycaemic control and reduced complications for users, particularly in regions with limited healthcare access (Asia) [9].

2.2.3.2 KRY and Doktor.se

KRY and Doktor.se are Swedish mobile apps that offer virtual consultations with healthcare professionals, including prescription services and referrals. These platforms improve accessibility to medical care, reduce pressure on physical healthcare facilities, and provide timely care to patients, enhancing overall healthcare efficiency (Sweden) [10].

2.2.3.3 Ehealth Nigeria

eHealth Nigeria uses the open-source OpenMRS platform to digitize health records, especially in resource-limited areas. The system improves data management for maternal health, polio, and HIV programs, making healthcare delivery more efficient and accurate while being accessible to healthcare workers with varying technical skills [11].

2.2.4 Future of E-health

2.2.4.1 Integration of Artificial Intelligence (AI) and Machine Learning

The integration of Artificial Intelligence (AI) and Machine Learning (ML) into eHealth is revolutionizing healthcare delivery, particularly through Remote Patient Monitoring (RPM). These technologies enable continuous, real time tracking of patient health metrics, facilitating early detection of potential health issues and allowing for timely interventions [12].

2.2.4.2 Expansion of Telemedicine and Remote Care

The adoption of telemedicine has accelerated, offering patients remote consultations and reducing the burden on healthcare facilities. This trend is expected to continue, with advancements in communication technologies enhancing the quality and accessibility of virtual care [13].

2.2.4.3 Smart Hospitals and Digital Twins

Smart hospitals are integrating advanced technologies like AI, IoT, and robotics to enhance patient care and operational efficiency. For instance, some hospitals employ AI to predict sepsis risks and use robots for surgeries and supply deliveries. The smart hospital market is projected to grow significantly in the coming years [14].

Digital twins virtual replicas of physical systems are being utilized in healthcare to simulate patient specific scenarios. These models can predict treatment outcomes, optimize hospital operations, and personalize patient care. For example, digital twins of the heart allow doctors to test treatments before actual procedures [15].

2.2.4.4 Blockchain Integration

Blockchain technology offers a decentralized and secure method for managing healthcare data. It ensures data integrity, enhances privacy, and facilitates interoperability among different healthcare systems. The global blockchain in healthcare market is expected to experience substantial growth, reflecting its increasing adoption [16].

2.3 Blockchain Technology

2.3.1 Definition

Blockchain is a secure technology and revolutionary, originally attached to the terms cryptocurrency and NFTs but now used in various industries such as healthcare, gaming, logistics, and education. This allows for the sharing of data in a decentralized, transparent, and tamper-resistant manner using the principles of distributed ledger technology (DLT), where each transaction is validated by several nodes of the network and added to a block chain. It is a system that reduces fraud, increases transparency, and eliminates intermediaries [17].

So Blockchain is a digital distributed ledger that stores data in the form of blocks that are linked together using a cryptographic function. Blocks are the containers of data that define a piece of digital information consisting of transactions along with the timestamp and cryptographic functions. A block consists of a header and a list of transactions. The block header contains metadata that includes the hash of the previous block, timestamp, nonce, and Merkle tree root [18]. The figure 2.1 summaries that.

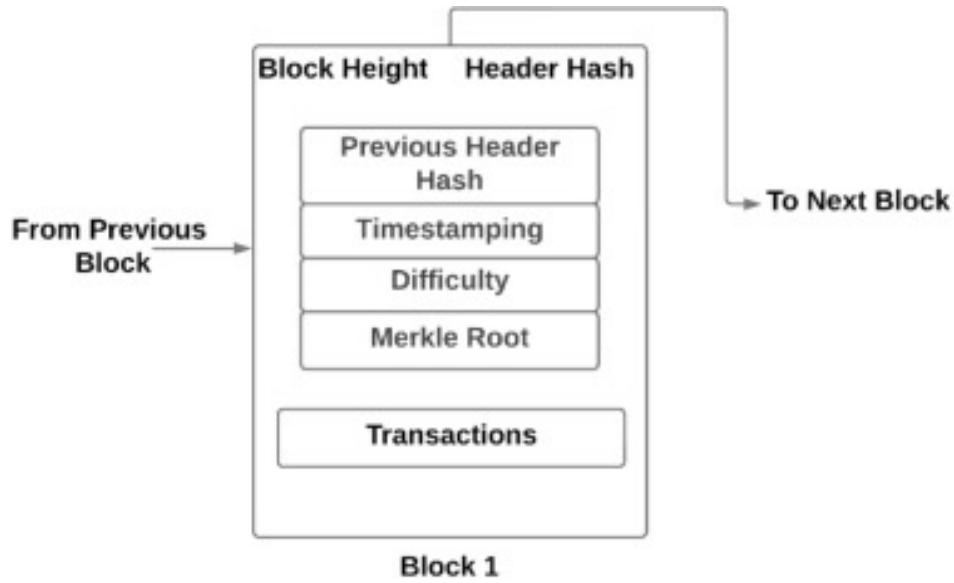


Figure 2.1: Block structure in a blockchain [18].

2.3.2 History and Evolution

Ever since its inception in the year 2008, blockchain technology has gained attention from across the globe. Today, many nations are adopting this technology to bring reforms across various sectors including finance, healthcare, education, governance, supply chain, agriculture, and energy. The following figure 2.2 gives a comprehensive overview of its historical background and evolution journey.

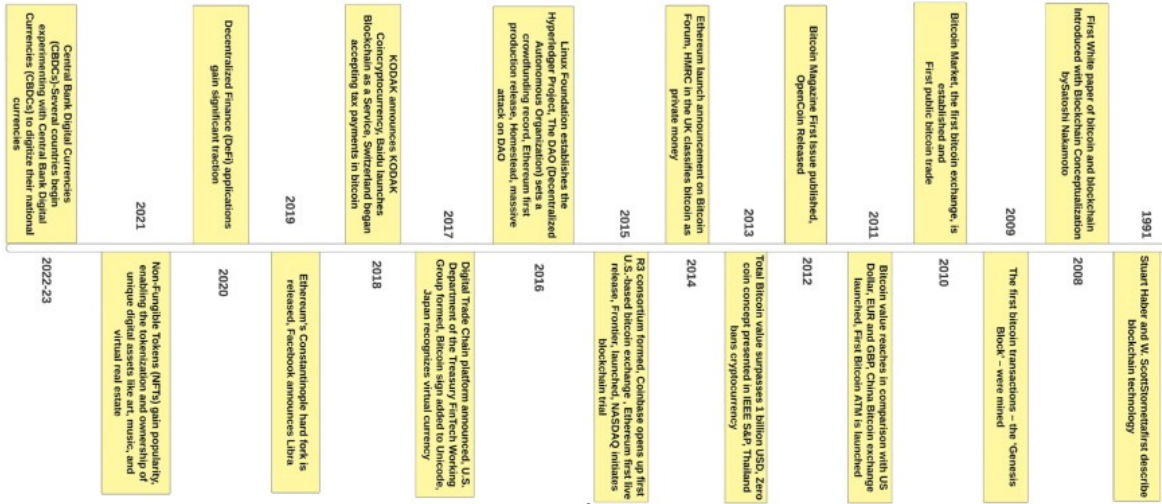


Figure 2.2: Key events in blockchain technology [18].

2.3.3 Types

The blockchain architecture depends on two major aspects. Firstly, the ownership of the data infrastructure citing whether the blockchain is public or private, and secondly, the permissions associated with the joining, read, write, and commit operations. Based on these aspects, the blockchain architecture and solutions are developed for the users. Open and closed blockchains are based on the write permissions while the public and private blockchains are based on the read permissions [19]. There are four basic sorts of blockchain technology. As shown in the picture 2.3.

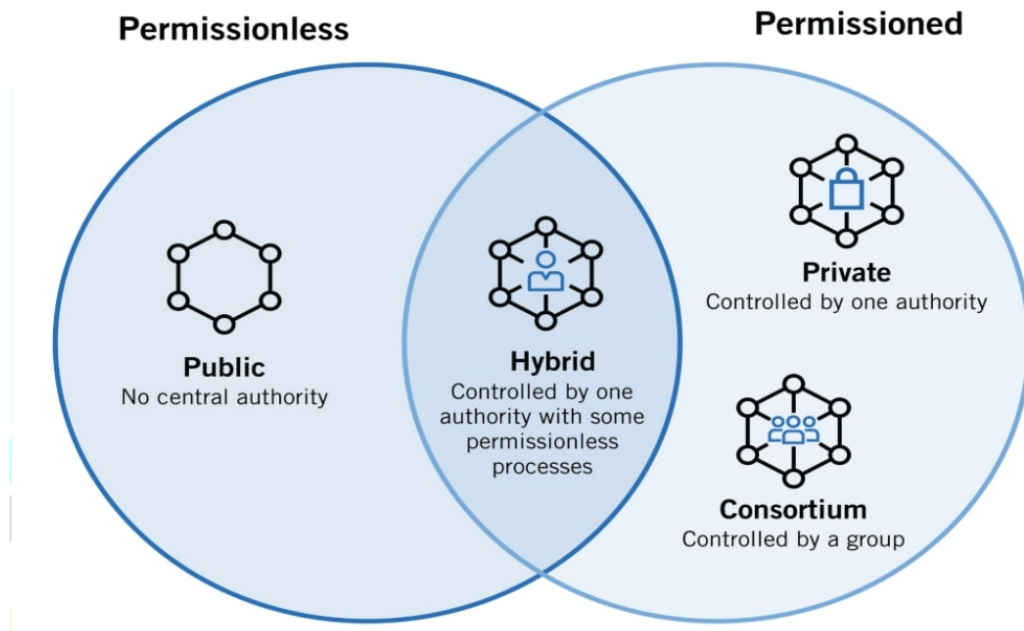


Figure 2.3: Types of Blockchain [20].

1. **Public Blockchain**

To be considered public, a blockchain has to be open and accessible to all, therefore permissionless. This means that anyone can use their computer to become a network's node. Once the software is downloaded and installed on the computer, such a blockchain node can then perform mining, verify transactions, or access the entire network's record. It operates using Proof of Work (PoW) and Proof of Stake (PoS) mechanisms to validate transactions, rewarding participants accordingly. Public blockchains are widely used for cryptocurrency activities like mining and trading, with Bitcoin, Ethereum, and Litecoin being notable examples [21].

2. **Private Blockchain**

This type of blockchain operates within closed systems and private networks, making it especially suitable for organizations and enterprises where access is limited to selected members. It emphasizes strong security, controlled access, proper authorization, and permission management [21]. Correspondingly, private blockchains are permissioned distributed ledgers. Well known examples include Multichain, Hyperledger projects, and Corda [22].

3. **Hybrid Blockchain**

Hybrid blockchains combine elements from both private and public networks. Companies can set up private, permission based systems alongside a public system. In this way, they control access to specific data stored in the blockchain while keeping the rest of the data public. They use smart contracts to allow public members to check if private transactions have been completed. For example, hybrid blockchains can grant public access to digital currency while keeping bank-owned currency private [23].

4. **Consortium Blockchain**

Otherwise known as a federated blockchain, a consortium network is a type of hybrid blockchain, but with multiple organizations in charge of its semi-closed ecosystem instead of just one. This gives the network a greater degree of decentralization while retaining the advantages of both private and public blockchains [22]. Some of the examples of this type of consortium are Energy Web Foundation, R3 [21].

The following picture 2.4 compares between four main types of blockchain, and the aspects of comparison are: advantages, disadvantages and use cases.

	Public (permissionless)	Private (permissioned)	Hybrid	Consortium
ADVANTAGES	<ul style="list-style-type: none"> + Independence + Open access + Transparency + Trust 	<ul style="list-style-type: none"> + Access control + Performance + Simpler consensus 	<ul style="list-style-type: none"> + Access control + Performance + Scalability + Third-party communications 	<ul style="list-style-type: none"> + Access control + Efficiency + Scalability + Security
DISADVANTAGES	<ul style="list-style-type: none"> - Environmental impact - Performance - Scalability - Security 	<ul style="list-style-type: none"> - Auditability - Environmental impact - Transparency - Trust 	<ul style="list-style-type: none"> - Environmental impact - Transparency - Upgrading 	<ul style="list-style-type: none"> - Complex setup - Environmental impact - Transparency - Trust
USE CASES	<ul style="list-style-type: none"> • Cryptocurrency • Document validation • NFTs 	<ul style="list-style-type: none"> • Asset ownership • Internal voting • Supply chain 	<ul style="list-style-type: none"> • Medical records • Real estate • Retail 	<ul style="list-style-type: none"> • Banking • Research • Supply chain

Figure 2.4: Difference between types of blockchain [24].

2.3.4 Blockchain Vs Traditional Databases

Both blockchain and traditional databases are used to store and manage data, but they differ significantly in structure, control, and use cases. The table 2.1 explains how do traditional databases differ from blockchain [25],[26]:

Table 2.1: Comparison between Blockchain and Traditional databases.

Characteristics	Traditional databases	Blockchain
Architecture	Client Server	Peer to peer
Authority	Centralized	Distributed/Decentralized
Control	Administrator	Consensus (e.g. PoW, PoS)
Permissions	Permissioned	Public, Permissioned, Hybrid
Disintermediation	Data is centrally administered by a DBA	Trustless and decentralized: No need of intermediaries to validate and authorize the stored data
Robustness	Single point of failure	No single point of failure or control
Audit trail of all data instances	History of records and ownership of digital record not available. Data can be updated and modified	History of records and ownership of digital record. Data is traceable to its source. A new block added for any new addition or modification

Data Handling	CRUD	RW
Data Storage	Tables	Blocks
Public Verifiability	Data integrity is low	Data Integrity and Transparency is maintained
Transparency	Database administer decides what is visible to the public	Public Blockchains Supports data transparency
Integrity	Data can be subjected to malicious attacks	Supports data Integrity
Data Persistence	Non-Persistent	Immutable

2.3.5 How It Works

Blockchain technology has gained significant attention as a disruptive innovation capable of transforming data security, transaction transparency, and trust mechanisms in digital systems. Unlike traditional centralized architectures, blockchain relies on a distributed and consensus driven infrastructure that eliminates intermediaries. To fully understand its value and impact, it is essential to examine how blockchain operates at a technical level. The following steps provide a detailed breakdown of its fundamental processes [18]:

1. Transaction Initiation

A transaction is created by a user and digitally signed using cryptographic keys, this step ensures authenticity and privacy of the operation.

2. Transaction Broadcast

The transaction is broadcast to a decentralized peer to peer network. All nodes (participants) receive this transaction.

3. Validation via Consensus

Network nodes validate the transaction using consensus mechanisms like Proof of Work (PoW) or Proof of Stake (PoS). This step ensures trust without a central authority.

4. Block Creation

Once validated, transactions are grouped into a new block. Each block contains a set of transactions, a timestamp, a nonce (in PoW), and the hash of the previous block.

5. Block Linking (Chaining)

The new block is added to the chain using cryptographic hashes, forming an immutable and chronological ledger of all transactions.

6. Ledger Replication

The updated blockchain is replicated across all network nodes, ensuring all participants share the same history.

7. Immutability & Security

Due to cryptographic linking and distributed consensus, altering any block would require massive computational power or majority control of the network, making the system highly secure and tamper-proof.

2.3.6 Architecture

The design of blockchain systems is based on a layered architecture, which we show in Figure 2.7, below we explain the five layers from bottom to top:

2.3.6.1 Data Layer

This is the bottommost layer of a blockchain system. The nature of decentralization in blockchain can lead to issues like data tampering, untraceable node actions, and slow transaction verification, which affect data trust. The data layer addresses these concerns by using a connected block structure where each block stores transaction data and is linked via hash pointers. Any change to block data alters its hash, preventing tampering and ensuring data integrity. This model of decentralized trust is built on the data structure, storage structure, and ledger pattern [27].

2.3.6.2 Network Layer

The network layer is crucial for implementing decentralization at the physical level in blockchain systems. Blockchain nodes communicate directly with each other in a decentralized peer-to-peer (P2P) network (as shown in 2.5), bypassing a central authority. This P2P structure allows nodes to join or leave the system quickly and supports decentralized communication. Initially designed for distributing large files over unreliable networks, P2P networks involve multiple computers acting as both requesters and responders. Recent research has focused on enhancing system performance and security, with studies evaluating techniques to improve file-sharing systems, analyzing design issues, and addressing security threats like peer trust and secure traffic routing [27].

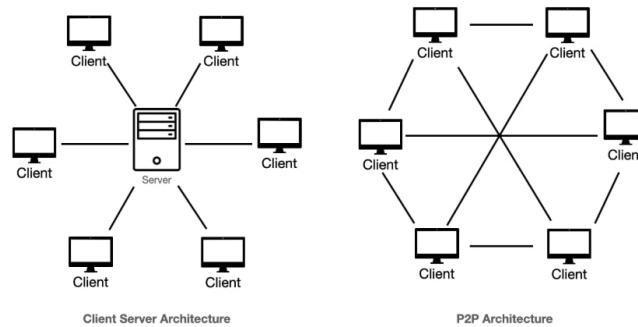


Figure 2.5: Client server Vs Peer to peer architecture[28].

2.3.6.3 Consensus Layer

The consensus layer manages the consensus algorithm, which coordinates the decentralized blockchain system to ensure secure and stable operation. In blockchain, nodes communicate and agree on events without a central authority, using a consensus algorithm to maintain data consistency across nodes. However, due to challenges like network latency, node failures, and bandwidth limitations, distributed systems face limitations outlined by the FLP (Fischer, Lynch, and Paterson) impossibility principle means that in a system containing multiple deterministic processes, as long as one process may fail, no protocol can guarantee a finite time for all processes to agree, and CAP theory (picture 2.6). These theories suggest that a perfect system cannot simultaneously achieve consistency, availability, and fault tolerance. As a result, consensus algorithms must balance trade offs. Blockchain consensus determines who can create blocks and package transactions, and these algorithms are generally categorized into proof based and voting based types [27].

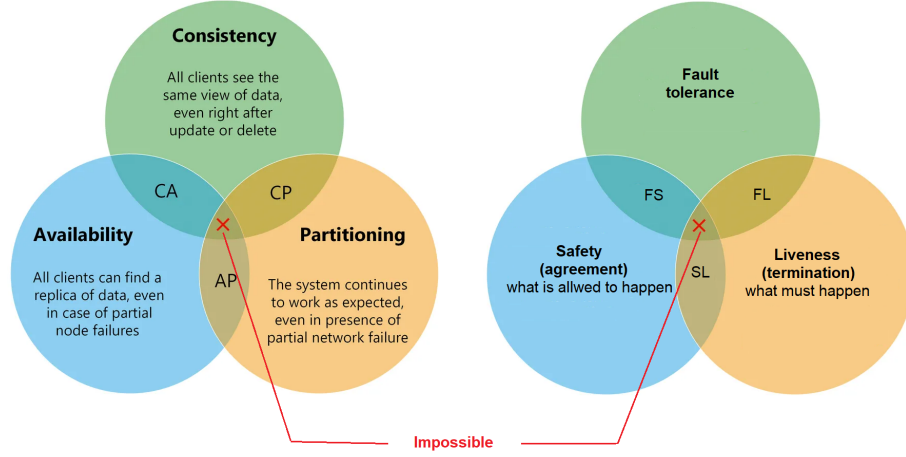


Figure 2.6: CAP and FLP theorems [29, 30].

2.3.6.4 Contract Layer

The contract layer enables smart contracts, which are predefined, immutable agreements that execute automatically when triggered. As an extension of the blockchain, smart contracts allow the blockchain to manage complex transactions [27].

2.3.6.5 Application Layer

This is an abstraction of the application built atop a blockchain system, that offers API interfaces that allow users to easily build decentralized applications (Dapps) using blockchain services. As blockchain technology advances, numerous Dapps have been developed to provide decentralized trust solutions for various traditional industry challenges [27].

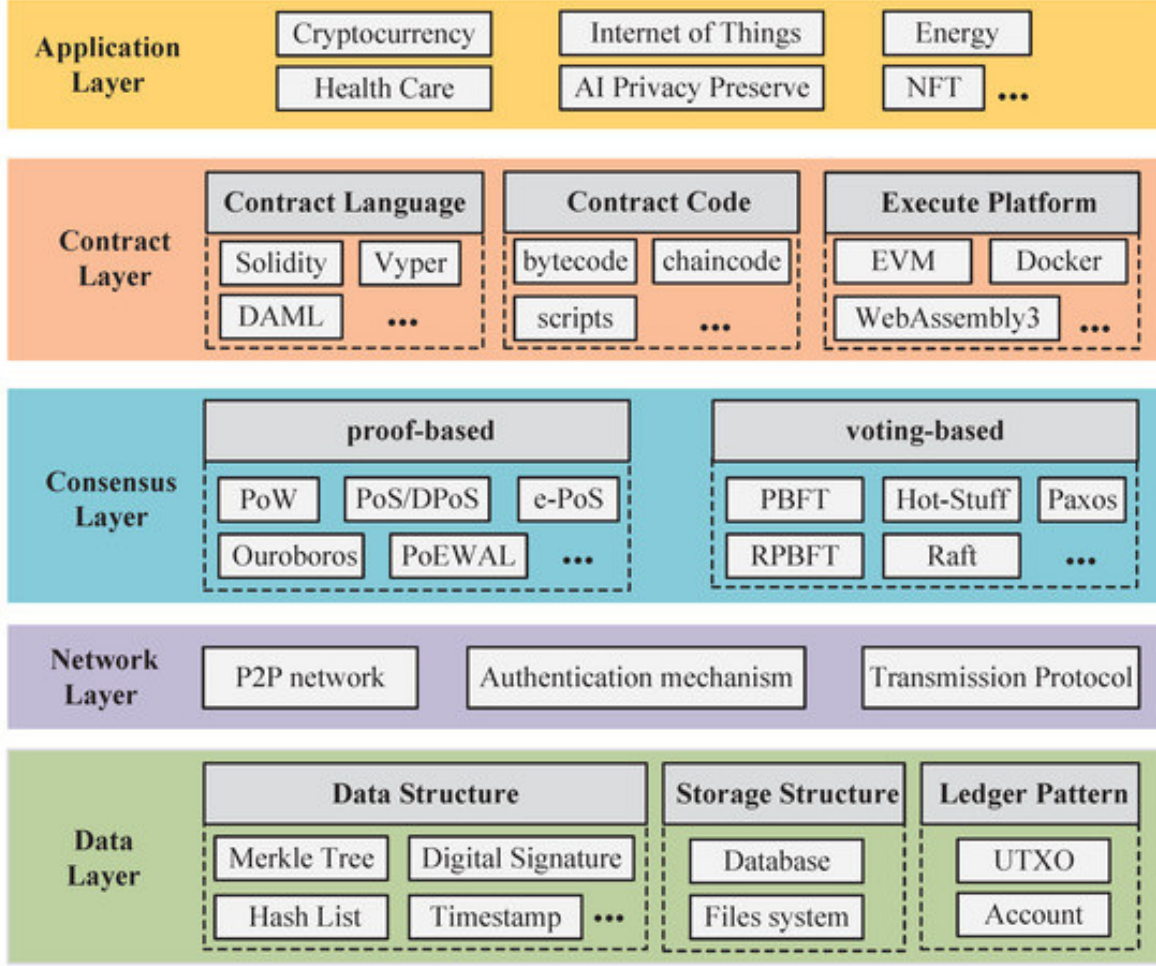


Figure 2.7: Blockchain Architecture [27].

2.3.7 Applications

Blockchain technology has emerged as a revolutionary innovation with the potential to transform a wide range of industries beyond its original application in cryptocurrencies [31]. Thanks to its core features decentralization, transparency, security, and immutability blockchain provides new ways to handle data and transactions in a trustless environment. As a result, it is being increasingly adopted in various sectors such as [32, 33, 34]:

2.3.7.1 In Healthcare

The need for secure and interoperable healthcare systems has grown with the rise of IoT devices and mobile health apps that generate large amounts of medical data daily. Blockchain technology offers a solution by enabling secure recording and sharing of medical records while preserving patient privacy. However, while it simplifies healthcare management, it also brings certain challenges that must be addressed.

2.3.7.2 In Communication

The rapid advancement of information and communication technologies has led to increasingly complex and heterogeneous communication systems, with a growing number of end devices and vast amounts of data. This complexity introduces significant challenges in security, privacy, service delivery, and network management. Integrating blockchain with machine learning offers promising solutions to these challenges and has attracted strong interest from both academia and industry.

2.3.7.3 In Cloud Gaming System

Cloud gaming enables game scenes to be rendered in the cloud and streamed as real time video to end devices, allowing users to play games anytime, anywhere, regardless of their hardware limitations. However, as a commercial service, its payment models are still in the early stages of development.

2.3.7.4 In Energy Sector

For governments, effective energy distribution, allocation, and production are crucial for economic growth. Without a well developed energy strategy, governments cannot foster necessary economic development. Similarly, private entities also play a significant role in contributing to blockchain, both materially and immaterially.

2.3.7.5 In Banking

A Corda-based implementation is developed through a collaboration of financial institutions, following industry standards. Corda is a blockchain ledger platform that has been widely adopted, while Hyperledger Fabric is recognized for its security and privacy features, making it ideal for banks. Additionally, Ethereum is a platform that allows the development of decentralized applications.

2.3.7.6 In Real Estate

In recent years, the real estate sector has faced challenges, particularly with rising house prices making property ownership more difficult. However, blockchain technology offers a new perspective, enabling the issuance of tokens tied to specific actions. This could allow properties to be leased through predefined programming, while tokens also help businesses incorporate necessary logic, such as fraud prevention.

2.4 Hyperledger Project

The Hyperledger Project, initiated by the Linux Foundation, is an open source collaborative effort designed to advance cross industry blockchain technologies, launched in 2015. It aims to provide modular, scalable, and enterprise-grade distributed ledger frameworks that can be tailored for various business use cases [35].

The broader Hyperledger project consists of multiple frameworks 2.8, including:

- Hyperledger Fabric 2.4.1: for modular permissioned blockchains.

- Hyperledger Sawtooth: supporting parallel transaction execution and smart contract flexibility.
- Hyperledger Indy: focused on decentralized identity.
- Hyperledger Besu, Iroha, etc.

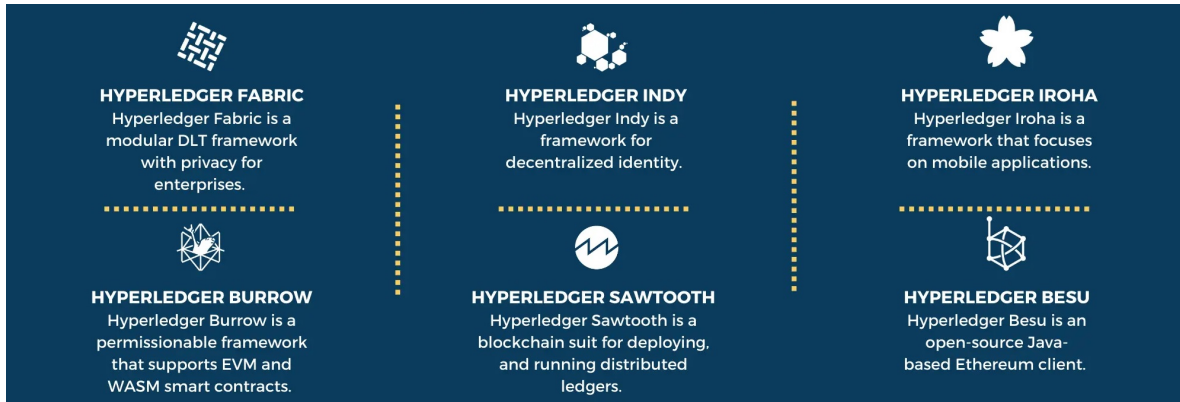


Figure 2.8: Hyperledger projects [36].

2.4.1 Hyperledger Fabric

There are numerous popular enterprise blockchain projects in the blockchain space, and Hyperledger seems to be one of them [37], is an open source permissioned blockchain system established under the Linux foundation and the first blockchain system that supports the creation of smart contracts in general purpose languages that allows clients to submit transactions to a blockchain system which offers decentralized control of a shared, distributed state [38]. Hyperledger isn't a single framework, it is an umbrella project that includes several other individual projects beneath it [36].

2.4.1.1 Hyperledger Fabric Architecture

The architecture of Hyperledger Fabric is modular and highly customizable, allowing businesses to tailor their blockchain networks to meet specific needs. It uses a permissioned model, where participants must be authenticated through a Membership Service Provider (MSP), ensuring secure and private transactions. Key components like peers, orderers, and channels facilitate efficient and secure transaction processing, while chaincode (smart contracts) automates business logic execution. The consensus mechanism ensures that the network operates in a consistent and fault-tolerant manner [39, 40]. Here is an overview of Hyperledger Fabric Architecture:

- **Peers[40]:**

This is the main component of Hyperledger Fabric. It is responsible for managing the ledger, executing smart contracts, and participating in the consensus process. A peer node in Hyperledger Fabric is a server that runs the Hyperledger Fabric software and is part of a network of peer nodes that make up a Hyperledger Fabric blockchain. Each peer node stores a copy of the

ledger and participates in the consensus process to validate and endorse transactions, as well as maintain the state of the ledger. Peer nodes can also run smart contracts, known as chaincode in Hyperledger Fabric, which define the business logic of the blockchain network.

1. **Endorser Peers:** Responsible for endorsing transactions. They execute chaincode and simulate transactions to verify if they meet the endorsement policy before a transaction is submitted for ordering.
2. **Committing Peers:** Maintain a copy of the ledger and state database. They commit transactions and blocks to the ledger after they are validated and ordered.

- **Orderers**[40, 39]:

In Hyperledger Fabric, an orderer is a component that is responsible for ensuring the delivery of transactions to the appropriate peer nodes for validation and endorsement. The orderer maintains an ordered log of all transactions that have occurred on the network and provides a communication channel for the peer nodes to reach a consensus on the order in which transactions should be processed. The orderer does not validate or endorse transactions but rather acts as a mediator to facilitate communication between the different peer nodes.

There are 5 types of Orderer[40]:

1. **Solo Orderer:** is a single node that is responsible for maintaining the ledger and ordering transactions in the network. This type of orderer is typically used in small or test networks.
2. **Kafka Orderer:** is a cluster of orderer nodes that use the Apache Kafka distributed messaging system to order transactions and maintain the ledger. This type of orderer is typically used in large or complex networks.
3. **Raft Orderer:** is a cluster of orderer nodes that use the Raft consensus algorithm to order transactions and maintain the ledger. This type of orderer is typically used in networks that require high performance and low latency.
4. **SNS Orderer:** is a cluster of orderer nodes that use the Amazon Simple Notification Service (SNS) to order transactions and maintain the ledger. This type of orderer is typically used in networks that are hosted on Amazon Web Services (AWS).
5. **CouchDB Orderer:** is a cluster of orderer nodes that use the Apache CouchDB database to store and order transactions in the network. This type of orderer is typically used in networks that require high availability and scalability.

- **Membership Service Provider (MSP):**

MSP is the component responsible for managing identities and ensuring that participants in the network are authenticated and authorized to interact with the blockchain, and defines the rules for identity validation and membership verification within the Hyperledger Fabric network [39]. There are several types of membership service providers (MSPs) in Hyperledger, including [40]:

1. **Local MSP:** This type of MSP is designed for development and testing purposes and is used to manage the identities of local users and applications.

2. **File-based MSP:** This type of MSP uses a file system to store the identity information of users and applications.
3. **Certificate Authority based MSP:** This type of MSP uses a certificate authority (CA) to issue and manage the digital certificates that are used to identify users and applications.
4. **Database based MSP:** This type of MSP uses a database to store the identity information of users and applications.
5. **External Identity Provider based MSP:** This type of MSP uses an external identity provider, such as a corporate LDAP server or an external CA, to manage the identities of users and applications.

- **Ledger**[39, 40]:

In Hyperledger Fabric, the ledger is a distributed database that records all of the transactions that occur on the network and stores the entire history of transactions. Each peer node maintains a copy of the ledger, and the ledger is updated whenever a new transaction is endorsed and committed to the network. The ledger is composed of two parts: the world state, which stores the current state of all assets on the network, and the transaction log, which stores a record of all transactions that have occurred on the network.

It consists of two parts:

1. **Blockchain:** the core data structure that records all transactions in a sequence of blocks. Each block contains a batch of transactions, and blocks are linked together to form a chain.
2. **State Database:** an off-chain database that stores the current state of the ledger. It is used to quickly query and retrieve the current values of assets and other data.

- **Chaincode (Smart Contracts)**[39, 40]:

In Hyperledger Fabric, chaincode refers to smart contracts that define the business logic and rules for updating the ledger. Written in Go, chaincode is executed by endorsing peers to validate transactions before they are ordered and committed to the ledger. There are two types of chaincode:

1. **System Chaincode:** Managed by network administrators for controlling network operations. It can be written in various programming languages such as Go, JavaScript, or Java.
2. **User Chaincode:** Managed by users to implement application specific logic and runs on endorsing peers, its function is to validate and execute transactions.

- **Consensus Algorithm:**

In Hyperledger Fabric, the consensus algorithm ensures that all peers agree on the order and validity of transactions, maintaining a consistent and secure ledger. The ordering service handles this by organizing transactions into blocks and distributing them to peers. Fabric supports multiple consensus mechanisms, with Raft as the default[40].

Hyperledger Fabric allows for the use of various consensus algorithms, such as (show table 2.2):

Table 2.2: Consensus algorithms [18, 41].

Consensus Algorithm	Advantage	Disadvantage	Example
PoW (Proof of Work)	Secure, reduces the chances for Denial-of-service attack	Computationally intensive, slow, requires lots of energy, potential for 51% attack	Bitcoin, Ethereum
PoS (Proof of Stake)	Computationally less intensive, energy efficient, difficult to attack	Nothing at stake problem, stakeholders can join power to form centralized authority, 51% attack	Peercoin
PBFT (Practical Byzantine Fault Tolerance)	Scalable, fast	Used in private and permissioned blockchains	Stellar, Ripple, Hyperledger Fabric
PoB (Proof of Burn)	Expensive computer equipment not needed	Resources are wasted needlessly as mining power is given to those who burn money	Slimcoin, TGCoin
PoC (Proof of Capacity/ Space)	Energy efficient, prevents Denial of Service attack	Incentive issue	Spacemint, Burstcoin
PoET (Proof of Elapsed Time)	Low cost of participation	Not apt for public blockchains	Hyperledger Sawtooth

- **Channels** [40]:

In Hyperledger Fabric, channels are used to provide additional privacy and security by creating private sub-networks within the larger network. These channels allow specific groups of participants to execute transactions and share data confidentially, with each channel having its own separate ledger. Participants can only see transactions submitted to their respective channel, ensuring that sensitive information is not exposed to others on the network. there are three types of channels:

1. **Application Channels:** These channels are created by the organizations that are part of a consortium on the network. They are used for conducting transactions and sharing data among the members of the consortium.
2. **System Channels:** These channels are created by the network administrator and are used for deploying and updating the network's shared ledger and other system-level components.
3. **Private Channels:** These channels are created by members of the consortium and are

used for conducting private transactions between two or more specific organizations. Private channels allow organizations to maintain the confidentiality of their transactions while still being able to take advantage of the security and immutability of the shared ledger.

- **Certificate Authority (CA) [39]:**

The CA is responsible for issuing certificates to participants, ensuring their identity in the network, and interacts with the Membership Service Provider to validate identities before participants can join the network.

- **Client [39]:**

The client is an application or service that interacts with the Hyperledger Fabric network, it submit transactions to endorsing peers and request information from the ledger.

2.4.1.2 How Hyperledger Fabric Works

The transaction flow in Fabric follows three phases: execution, ordering and validation. This is referred to as the Execute Order Validate (E-O-V) model [42], and is visualized in Figure 2.9:

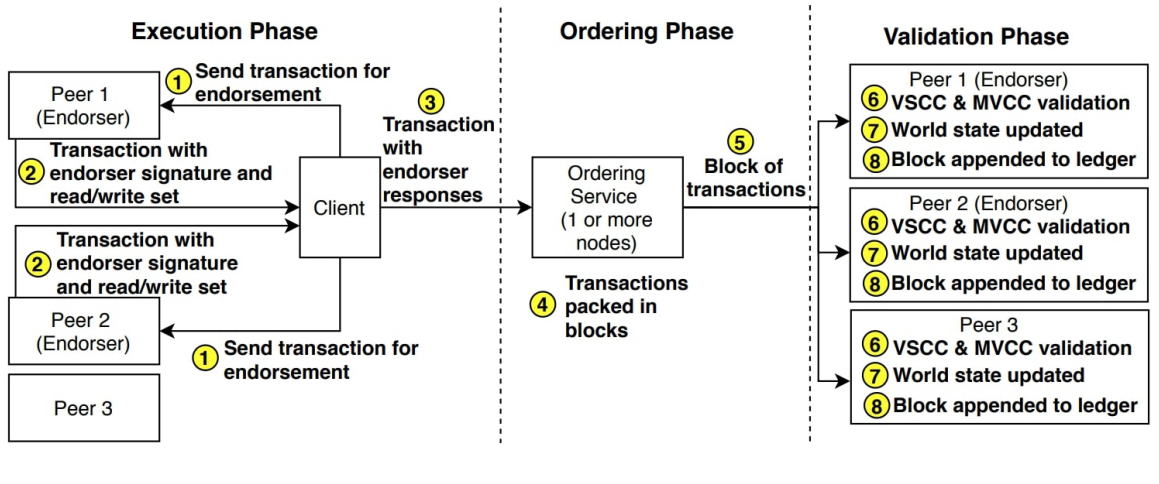


Figure 2.9: Transaction flow in Hyperledger Fabric [42].

- **Execution Phase**

- **Step 1:** The client sends a transaction to all the endorsers. This transaction can include multiple reads and writes to one or more keys in the world state.
- **Step 2:** The endorsers simulate the execution of the transaction on the world state and generate a read/write set that corresponds to the current world state of every key involved in the transaction. Then, the endorsers send a response back to the client containing their signature and the read/write set. This distributed execution helps to maintain trust without a centralized authority.
- **Step 3:** The client collects the endorsing peers' responses and sends them to the ordering service nodes. Optionally, the client may check the validity of the endorsing peers' signatures

and the consistency of the read/write set received from different peers. These checks are mandatory in the validation phase. By doing this, the client can detect transaction failures early to reduce overhead.

- **Ordering Phase**

- **Step 4:** (Ordering Phase) The ordering service orders the transactions received from the client using a consensus protocol. A transaction block is created based on three conditions:
 - * If a fixed duration of time has elapsed (block timeout),
 - * If a fixed number of transactions have been received (block size),
 - * Or if the total size of transactions reaches a fixed limit (block max bytes).
- **Step 5:** The block of transactions is then sent to all the peers

- **Validation Phase**

- **Step 6:** Every peer, upon receiving a block of transactions from the ordering service, validates each transaction in the block independently. A peer checks if a sufficient number of valid endorsing peer signatures, based on the endorsement policy, have been collected (Validation System Chaincode (VSCC) validation). Then, the peer verifies if the version of every key in the read set of each transaction matches the version of the same key in the current world state (MVCC validation).
- **Step 7:** If both VSCC and MVCC validation checks pass, the write sets of the transactions are applied to the world state. If any validation checks fail, the client is notified that the transaction was aborted, and the world state remains unchanged.
- **Step 8:** The validated block containing both aborted and committed transactions is appended to the ledger. The commit or abort status of every transaction is logged.

2.5 Wireless Body Area Network

2.5.1 Definition

Body Area Networks (BANs), particularly Wireless Body Area Networks (WBANs), consist of multiple sensors placed on or in the human body to monitor vital signs and physical activities. These networks face challenges due to dynamic body movements and must adapt to changing link conditions to maintain performance and energy efficiency [43].

It is also known as wearable or implantable networks of interconnected smart sensors used to continuously collect physiological and physical data from a person's body for medical or fitness related purposes. These networks wirelessly transmit the collected data to a remote healthcare provider for diagnosis or monitoring [44].

Figure 2.10 visually explains the concept of WBAN.

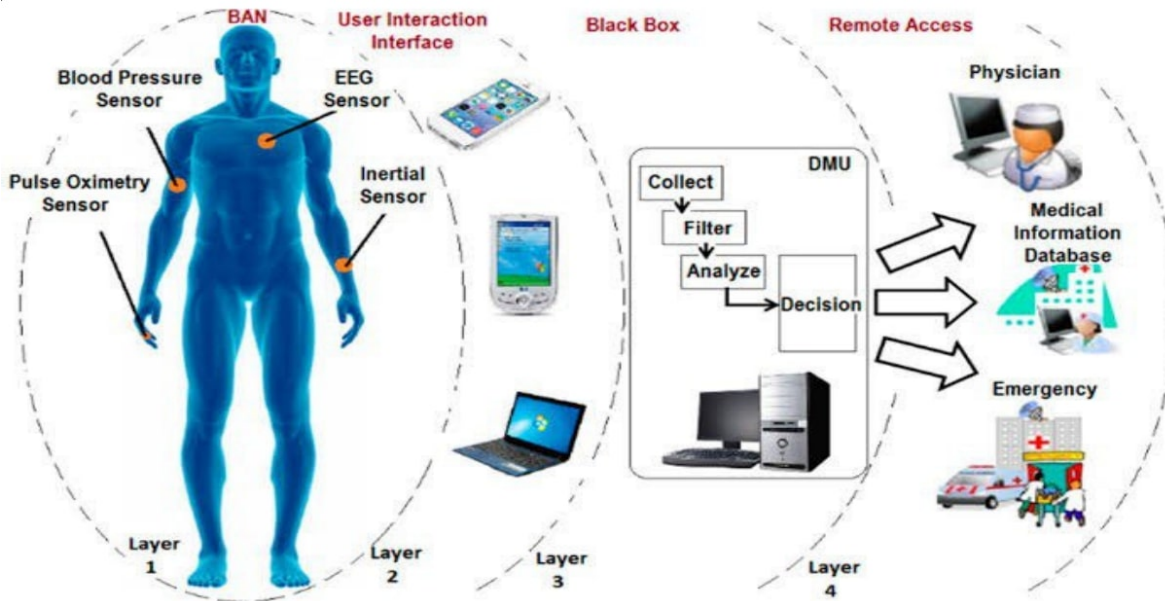


Figure 2.10: Wireless body area network scheme [45].

2.5.2 The Body Sensor Network Function

Sensor nodes in body sensor networks monitor key health indicators, aiding in accurate medical diagnosis and decision making [46].

Here some examples of different types of sensors used in various ways showing in figure 2.11 [46]:



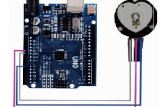






ECG The heart's electrical activity 	Blood flow Measurement of accelerating forces in three-dimensional space of the body 	Heart rate The frequency of the cardiac cycle 
Blood pressure The force applied by the circulation of blood on the walls of the blood vessels 	Body temperature An indicator of the body's ability to create and release heat 	Respiration rate Number of inhale and exhale movements per unit time 
Oxygen level Indicates the oxygen that is flowing in the patient's blood 	Blood sugar Measures the amount of sugar (type, source, energy) in the blood 	Electroencephalography Measures automatic brain activity and other brain capacities 

Figure 2.11: Various medical sensors deployed on the human body [47], [48], [49], [50], [51], [52], [53], [54], [55].

2.5.3 Wireless Communication Technologies

Wireless networks are widely studied due to their potential benefits and applications. However, their success largely depends on aligning the technology’s capabilities with user needs and expectations. In body sensor networks, where sensitive health data is transmitted, reliable and interference-free communication is essential [46]. This section introduces wireless technologies used in such networks [46]:

2.5.3.1 ZigBee Technology

A low power, intelligent network protocol based on IEEE 802.15.4, used in personal area networks for low rate data transmission. In body sensor networks, it provides 128 bit encryption and supports star, tree, and mesh topologies. It operates at 2.4 GHz with a transmission speed of 252 kbps, requiring each device to have a coordinator.

2.5.3.2 Bluetooth Technology

A low cost, low energy wireless communication protocol designed for short range data exchange using radio waves. It is well suited for body sensor networks, enabling real time monitoring through small embedded chips that create personal wireless networks for nearby device communication

2.5.3.3 Ultrawideband (UWB) Technology

A low energy radio technology designed for short range, high-bandwidth communication across a broad frequency spectrum. It can transmit data at rates between 20 kbps and 850 Mbps, making it suitable for continuous monitoring of physiological signals like ECG/EEG. Although promising for body sensor networks, UWB components are not yet widely available for practical implementation.

2.5.3.4 Wi-Fi—IEEE 802.11

Based on the IEEE 802.11 standard, supports high-speed data transfer over 2.4 and 5 GHz bands with a range of about 100 meters. Suitable for large data transfers like video streaming and conferencing, it is widely supported by most devices. However, its high energy consumption makes it less ideal for body sensor networks.

2.5.4 WBAN Challenges

As an emerging technology, Wireless Body Area Networks (WBANs) face numerous technical and ethical challenges that need to be addressed, including concerns about privacy. Some of the most critical technical issues are summarized in Table 2.3 [46].

Table 2.3: WBAN challenges.

Challenges Related to	Description
Range	The WBAN range is limited, a few meters away from the body. Hence, reliable wireless communication is performed inside or close to the human body.
Energy	Consumption WBAN requires constant energy to work properly, which necessitates a constant power supply.
Security	Due to the low power supply, it is difficult to add complex security mechanisms.
Service quality	One of the most important challenges in WBAN is to improve service quality.
Placement	It is difficult to put many nodes in a limited space.
Mac Layer	Minimizing delays in sending critical information is essential. To address this, quality control services should be used to prioritize and expedite urgent data. Additionally, direct transmission of sensitive information is recommended to further reduce delays.
Network Layer	Improper routing can lead to high energy use and traffic bottlenecks. To prevent this, load-balancing routing protocols, along with traffic prioritization and quality control, should be used to optimize performance and reduce congestion.
Transportation Layer	WBANs require a highly reliable transport mechanism to ensure accurate, timely delivery of critical health data, as loss or damage can be life-threatening. To save energy, periodic reporting can be used to limit unnecessary data transmission.
Application Layer	The user interface layer in WBANs should use AI and machine learning to deliver meaningful, intelligent insights from medical data, enhancing decision-making and user interaction.

2.6 Conclusion

The first chapter outlined the essential technologies and concepts that underpin our system. eHealth showcases the digital transformation of healthcare, while blockchain offers a secure and transparent data management solution. Hyperledger Fabric presents a robust framework for implementing private, permissioned blockchains tailored to healthcare needs. WBANs enable continuous patient monitoring through sensor networks. Understanding these components is crucial to grasp the architecture and operation of the proposed system in subsequent chapters. The next chapter discusses **Blockchain Technology in eHealth**.

Chapter 3

Blockchain Technology in eHealth

3.1 Introduction

This chapter explores the integration of blockchain technology within the field of electronic health (eHealth). It begins by examining the relationship between **Blockchain** and **eHealth**, highlighting how blockchain's key features such as decentralization, immutability, and transparency can address critical challenges in healthcare data management. The chapter also reviews existing literature and related works to provide insight into current advancements, applications, and limitations of using blockchain in healthcare systems, also the security mechanisms employed to protect sensor networks.

3.2 Relationship between Blockchain Technology and Ehealth

Blockchain technology is increasingly being integrated into eHealth systems to address long-standing challenges in healthcare such as data security, interoperability, patient empowerment, and regulatory compliance. As healthcare becomes more digitized, blockchain provides a decentralized, tamper-resistant framework that supports trustworthy and efficient health services, this last led to a relationship between the two, we can explain the nature of this relation through different aspects:

3.2.1 Data Security and Privacy

One of the primary contributions of blockchain in eHealth is its ability to enhance data security and privacy. By design, blockchain stores information in encrypted, immutable blocks distributed across a peer to peer network [56]. This ensures that patient records cannot be altered without consensus, protecting sensitive health data from unauthorized access and breaches [57]. and secure architecture is crucial for protecting electronic health records (EHRs), especially in cloud based and telehealth platforms [58].

3.2.2 Interoperability and Data Sharing

Blockchain also plays a key role in solving interoperability challenges across different health systems. Since data on a blockchain can be accessed and verified by multiple authorized parties without

central control, it facilitates secure and real-time information exchange between healthcare providers. This feature enhances care coordination and reduces redundancy [57], and blockchain enabled interoperability can unify fragmented health systems and simplify cross-institutional data access [59].

3.2.3 Patient Centric Data Control

In traditional healthcare systems, patients often lack control over how their health data is used. Blockchain introduces a shift toward a patient centric model by allowing individuals to manage data access through smart contracts. Patients can grant, revoke, or monitor access to their data, increasing transparency and trust, this approach empowers users and supports data ownership, privacy, and compliance with modern ethical standards [60].

3.2.4 Supports AI and Analytics

Blockchain complements artificial intelligence (AI) in eHealth by ensuring the integrity and traceability of datasets used for machine learning and predictive analytics. Inaccurate or tampered data can lead to poor clinical decisions, but blockchain's immutability guarantees that AI models train on reliable data, and the synergy can support early diagnosis, continuous monitoring, and treatment personalization in a secure and accountable way [56].

3.2.5 Legal and Regulatory Compliance

Blockchain helps healthcare organizations meet regulatory requirements such as HIPAA and GDPR by maintaining detailed, auditable logs of all data transactions. Each access or modification to a patient record can be transparently tracked, supporting accountability and compliance, and improve regulatory reporting and risk management in digital healthcare environments [57, 59].

So we can say that: **Blockchain technology** enhances **eHealth systems** by securing data, enabling interoperability, empowering patients with data ownership, and supporting AI based healthcare delivery all while ensuring legal compliance.

3.3 Related works

We will conduct two types of reviews, as detailed below:

3.3.1 Review on Blockchain Solutions for Healthcare Data Security

Blockchain technology has evolved beyond its initial application in cryptocurrencies to become a foundational innovation across various domains. Its decentralized, transparent, and immutable nature makes it suitable for systems requiring trust and data integrity, example in sector of **Education**, Primebook discusses blockchain's potential in making educational resources more accessible and simplifying student loan and scholarship management in India, [61] and in Slovenia EduCTX provides a global platform for issuing blockchain-based academic credits [62], also its applications in **Agriculture and food supply chains** filed, in Haïti nation AgriLedger helps farmers in developing countries get

fair market value and verify the origin of produce [63].

There are other domains that exploit the power of blockchain from the security side, such as:

- Finance and banking:
 - Citi, JPMorgan, and SIX Expand Blockchain Ventures to overhaul traditional finance systems. JPMorgan’s Onyx digital assets unit handles \$1bn daily transactions with its JPM Coin, and SIX Digital Exchange has created bridges between digital and traditional bond markets [64].
 - Mastercard’s Blockchain Initiatives developing a blockchain based network to facilitate digital asset transactions between consumers, merchants, and financial institutions, aiming to replicate its card network’s scale in the crypto space [65].
 - Janus Henderson Embraces Securities Tokenization asset manager Janus Henderson is exploring blockchain for converting fund units into digital tokens, aiming to reduce costs and increase efficiency in financial services[66].
 - JPM Coin is JPMorgan’s take on digital money, designed to help big institutions move funds faster and more efficiently. It’s not like Bitcoin this coin runs on a private blockchain and is backed one to one by the U.S. dollar. The main goal is to speed up payments, especially across borders, while keeping things secure and within the rules [64].
- Government and public services:
 - E-Voting and secure voting systems for transparent, and tamper proof e-voting systems to ensuring the integrity of elections [67]. In West Virginia piloted a blockchain based mobile voting system to facilitate secure voting for overseas military personnel [68].
 - Public records management using blockchain to store public records such as land titles, tax records, and identity documents securely. It reduces fraud and ensures transparency [69].
 - Estonia has implemented blockchain technology across various public services, including national ID systems, e-voting, and secure data management, setting a global benchmark for digital governance [70]. It’s e-Residency program and KSI blockchain infrastructure enable secure digital identities and services [71].

The integration of **Blockchain technology** in **Healthcare** has gained traction as hospitals seek secure, efficient, and interoperable systems for managing digital data. Numerous studies have explored how blockchain supports the digitalization of health records and ensures robust data security mechanisms (Table 3.1):

Table 3.1: Related work on Blockchain applications in Healthcare data security.

Year	Study Title	Technology Used	Purpose	Dataset/ Platform	Key Contributions
2025 [72]	Blockchain for Healthcare Management	Blockchain (PoA, ZKP)	Secure, transparent health data handling	Not specified	Increased speed (35%), accuracy (22%), and integrity (98.5%)
2024 [73]	Hospital Data Security Protocol	Blockchain	Secure sharing and backup of hospital data	Simulated blockchain protocol	Secure info recovery; encryption and access key management
2024 [74]	Hyperledger Fabric for EHR	Hyperledger Fabric	Decentralized EHR management	Frere Provincial Hospital	Case study implementation and benefit analysis
2023 [75]	Secure Data Transmission of EHR	Blockchain IPFS SPAKE protocol	Decentralized patient data access and control	Public hepatitis dataset	Smart contract based access, analysed overhead and block time
2023 [76]	Medical History Card Using Blockchain	Blockchain Web App	Unified, portable patient history display	Prototype Web App	Decentralized, secure storage, access by QR code
2021 [77]	Survey on Blockchain in Healthcare	Blockchain Survey	Comprehensive review of blockchain in health	Literature based	Identifies gaps and potential future research areas
2021 [78]	Security and Privacy for Healthcare Blockchains	ZKP, ABE, Anonymous Signatures	Secure and private data sharing	Conceptual framework	Cryptographic tools for privacy preserving blockchain

3.3.2 Review on Security Solutions in WBAN

Sensor networks are widely deployed across various domains such as smart cities, industrial monitoring, environmental sensing, and transportation systems. These networks often operate in open and

untrusted environments, making them vulnerable to a range of security threats including eavesdropping, data tampering, spoofing, and denial of service attacks [46]. The following table 3.2 summarizes security mechanisms employed to protect sensor networks. These mechanisms include cryptography for ensuring data confidentiality and integrity, key management protocols to securely handle cryptographic keys, secure routing techniques that defend against routing attacks, and trust management frameworks that evaluate node reliability. While effective, these approaches face challenges in scalability and resource constraints, which motivates the exploration of **Blockchain** based solutions as a new alternative [46].

Table 3.2: Related work on security mechanisms in sensor networks.

Year	Security Mechanisms	Study Title	Techniques
2021 [79]	key Management	Trust key management scheme for wireless body area networks	uses ECG signals to securely generate and distribute symmetric keys in WBANs, ensuring both data protection and user privacy without traditional authentication methods.
2020 [80]	Trust Management	Naïve Bayes based trust management model for wireless body area networks	A trust model using the Naïve Bayes classifier identifies sensor nodes as trustworthy or malicious. Trained in MATLAB, it enables secure data exchange by selecting only reliable nodes, with accurate classification results.
2018 [81]	Cryptography	Hybrid encryption algorithm in wireless body area networks (WBANs)	The Hybrid Encryption Algorithm (HEA) secures both ad hoc and wired sensor networks while addressing key sensor constraints like limited battery, bandwidth, processing power, and dynamic topology.
2017 [82]	Secure Routing	Secure routing of WBAN with monarchy butterfly optimization	secure routing in WBANs as a multi-objective optimization problem and uses Monarchy Butterfly Optimization to improve routing efficiency. It adapts to dynamic network conditions and ensures secure data transfer within a mobile ad hoc network setup.

3.4 Conclusion

The chapter demonstrated how blockchain enhances the efficiency, security, and trust in eHealth systems. By exploring their relationship and reviewing related works, it provided a clear view of current research and implementations, laying the groundwork for designing secure and decentralized healthcare solutions. The next chapter focuses on the **Conception and Implementation** of our system.

Chapter 4

Conception and Implementation

4.1 Introduction

In this chapter we will explain our proposed solution through the architecture of our system, and detail the design phase that modeled with use case, class and different sequence diagrams scenarios. Finally, we will present the tools and environment used for development.

4.2 Proposed Solution

The architecture of the proposed eHealth system (show figure [4.1](#)) is based on **Hyperledger Fabric**, a permissioned blockchain framework. The system is designed to ensure **privacy, scalability, and interoperability** across multiple healthcare institutions. The core architectural components and their interactions are described below:

- **Fabric Network Overview:**

The Fabric network consists of three hospital organizations: **Hosp1**, **Hosp2**, and **Hosp3**, each acting as an independent entity. These organizations are interconnected through a common channel named **hospitalchannel**, which facilitates secure and permissioned communication and data sharing.

Each organization hosts:

- **Peer Node (peer0)**: maintains the ledger (denoted L1) and executes smart contracts (SC). Each peer has its own copy of the ledger and participates in transaction endorsement.
- **Smart Contracts (Chaincode)**: installed and instantiated on peers. Define business logic for patient, doctor, nurse, and admin roles.
- **Certificate Authorities (CA1, CA2, CA3)**: CA1 for Hosp1, CA2 for Hosp2, CA3 for Hosp3. Each CA is responsible for:
 - * Registering and enrolling identities (admins, doctors, nurses, patients).
 - * Issuing X.509 digital certificates.
 - * Enabling secure authentication and identity verification via MSP.

* Crucial for ensuring trust and access control in the network.

- **Channel Configuration:**

The `hospitalchannel` is a private channel shared among the hospitals. It includes chaincode definitions, endorsement policies, and anchor peer configurations. The channel ensures secure collaboration while preserving data confidentiality.

- **Ordering Service:**

The **Orderer** node handles transaction ordering and block creation. It guarantees consistent block generation, transaction ordering, and synchronizes ledgers across all peers.

- **Network Configuration:**

A global configuration defines the consortium, Membership Service Providers (MSPs), Certificate Authorities (CAs), and identity management policies. Only authenticated entities can access the network.

- **External Interfaces:**

- **Fabric SDK (Node.js) Server:** serves as middleware between the frontend and blockchain. Handles transaction submission, querying, identity authentication, and network access via the SDK.
- **Frontend: HTML, CSS, JavaScript:** a responsive web interface for patients, doctors, nurses, and admins.
- **Dockerized Deployment:** all Fabric components (peers, orderer, CA, SDK server) are containerized for portability, scalability, and environment isolation.

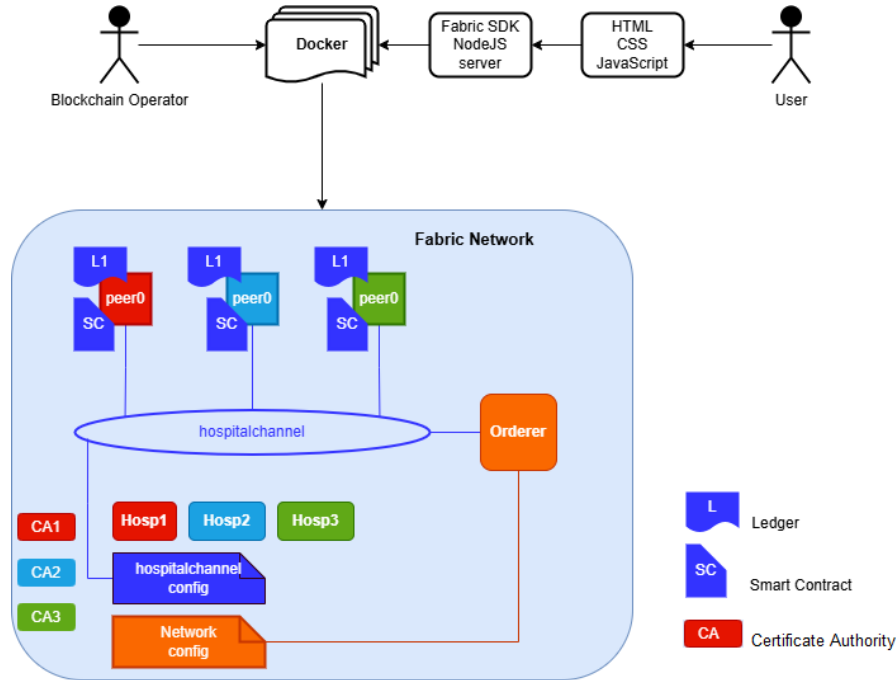


Figure 4.1: System architecture.

In summary, the system architecture 4.1 ensures secure and efficient communication between health-care institutions using blockchain technology. It provides robust data integrity, selective access, and traceability for all participants in the eHealth network.

4.3 System Diagrams

To explain our system in a visual way to facilitate understanding, we chose **Unified Modeling Language** (UML) which is a pictogram based graphical modeling language designed as a standardized method of visualization in the fields of software development and object oriented design [83].

4.3.1 Use Case Diagram

In this section, we present the different functional needs of our system that we will express through the use case diagram illustrated in Figure 4.2.

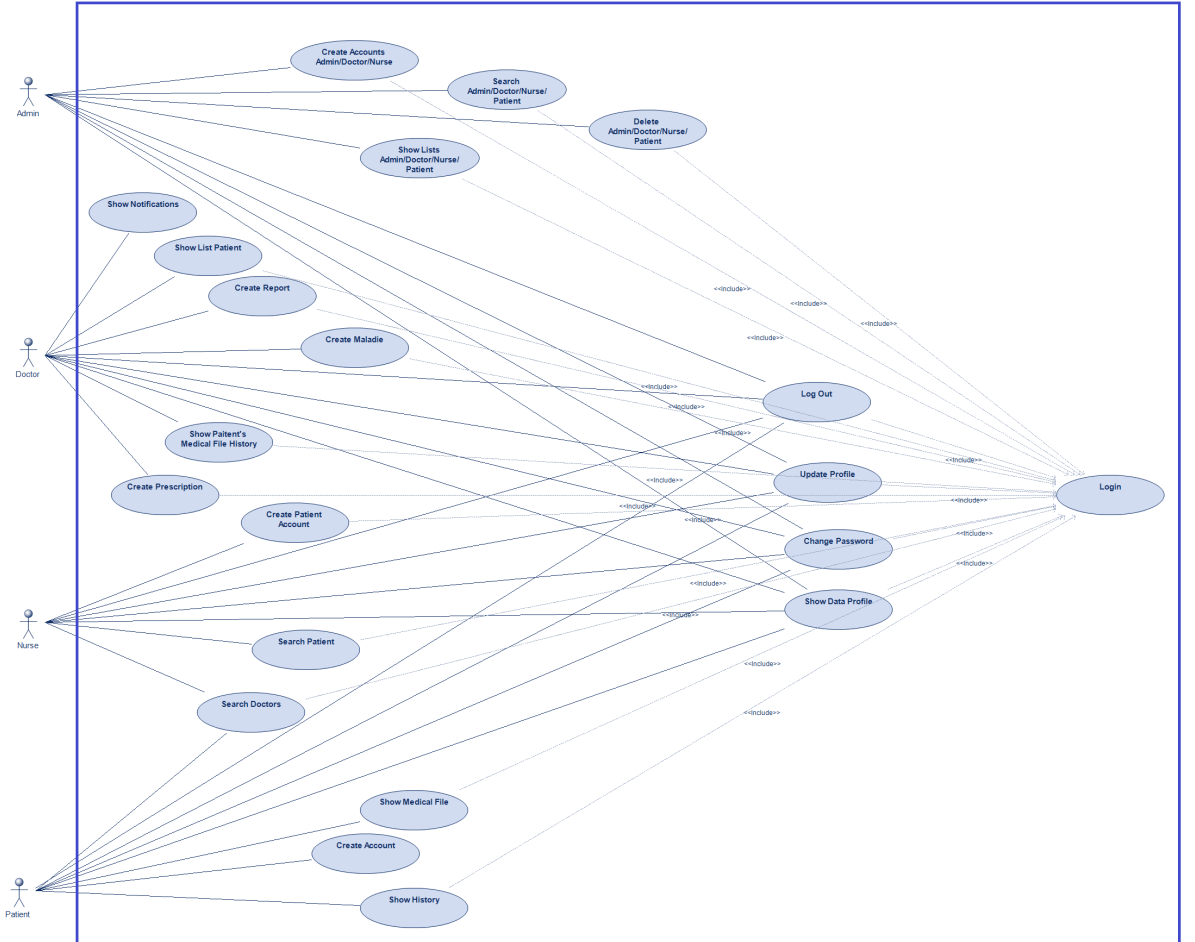


Figure 4.2: Use case system diagram.

This type of diagram delineates the system and describes how to use it. More precisely, it shows

the external functionalities (use cases), the actors (roles played by users), and the interactions between the two [84].

In the following, we present the key features of our system:

- **Registration** (see the case "Create Accounts Admin/Doctor/Nurse"): The administrator must create accounts for new admins, doctors and nurses when first logging into the system by entering their mandatory information: id, firstname, lastname, password, email, phone number... The patient also creates his own account by entering his personal information: firstname, lastname, age, phone number, emerg phone number, address...(see the case "Create Account").
- **Login** (see the case "Login"): When any actor already has an account, they can log in via this feature by entering their username and password.
- **Show and edit profile** (see the cases "Show Data Profile", "Update Profile", "Change Password"): At this level, all actors can show his personal informations and change:
 - Firstname, lastname, phone number, address...
 - Password
- **Admin level** (see the cases "Search Admin/Doctor/Nurse/Patient", "Show Lists Admin/Doctor/Nurse/Patient", "Delete Admin/Doctor/Nurse/Patient"): Everything related to hospital management, such as displaying a list of administrators doctors and nurses working there, and a list of patients being treated there.
- **Doctor level** (see the cases "Show List Patient", "Show Patient's Medical File History", "Create Report", "Create Maladie", "Create Prescription", "Show Notifications"): At this level, the doctor can view his patients and their medical files, to be able to diagnose it and write them a prescription or report using a local data base, also can show the notification in case of unnatural rate.
- **Nurse level** (see the cases "Create Patient Account", "PatientExists", "Search Doctor"): He can create patient account for emergency situations or in case of unregistered patients, also can search about patients through his username and doctors through his speciality or his name.
- **Patient level** (see the cases "Show Medical File", "Search Doctor", "Show History"): After he create his account and login, he can show all information about his medical file with details, and search doctor by his specialization or his name.
- **Disconnection** (see the case "Log Out"): If users (Admin, Doctor, Nurse and patient) wish, they can keep their account open, and if they don't, they can log out.

4.3.2 Class Diagram

Class diagrams are abstract representations of system objects and how these objects are defined. They help show the static structure of a system [84].

4.3.3 Sequence Diagrams

Sequence diagrams: (or dynamic UML diagrams) representing a scenario (the exchange of messages) of a software program and describe how the system elements interact with each other and with the actors [84].

From the scenario 4.4, the admin can create accounts of other admins, doctors and nurses ('RegistreAdmin', 'RegistreDoctor' and 'RegistreNurse' methods to Admin), he just has to enter the necessary information according to the account type.

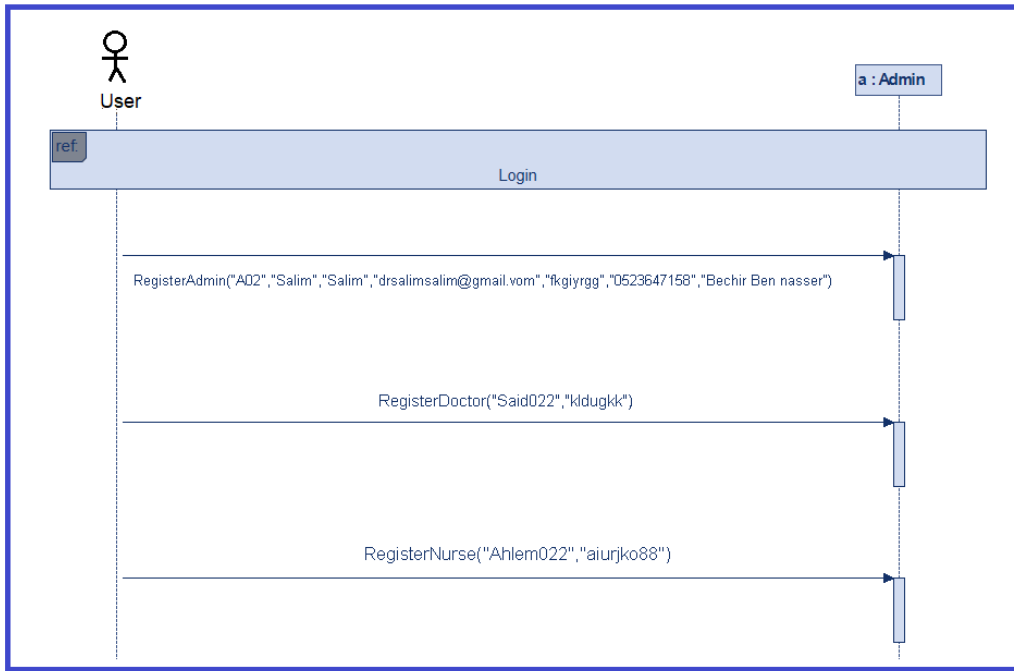


Figure 4.4: Sequence diagram: Admin scenario: Registration scenario.

The login scenario 4.5 all actors make the call to the method 'Login' from his class by entering username and password, then the method 'CheckExisting' check the existing of this user name and returns a Boolean value, if true the method 'ChekPassword' compare the entering password with the correct password, if the two are identical the method return the account (admin, doctor, nurse or patient according to the type of actor), if are not the method return error message "Password incorrect". Else ('CheckExisting' return false) means that this username does not exist (invalid username or not registered yet).

Note: CheckExisting and ChekPassword are implicit in 'Login' method.

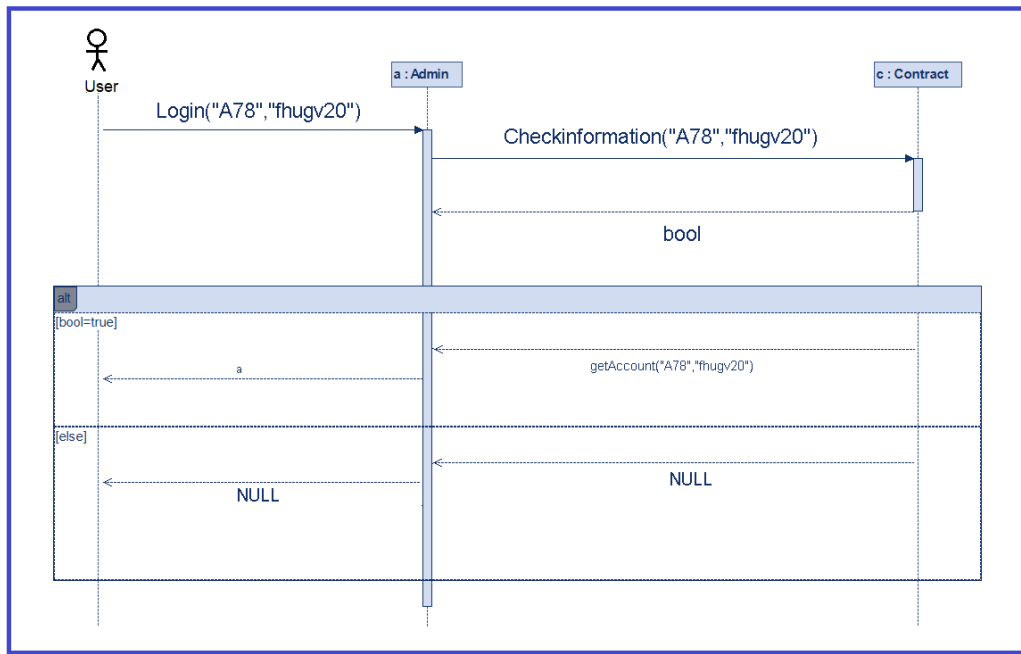


Figure 4.5: Sequence diagram: Login scenario.

The doctor can view his patients 'ShowMyPatient' and specify one to show his profile and medical file to diagnose his condition, and write an ordonnance if necessary. The figure 4.6 explains what has been said.

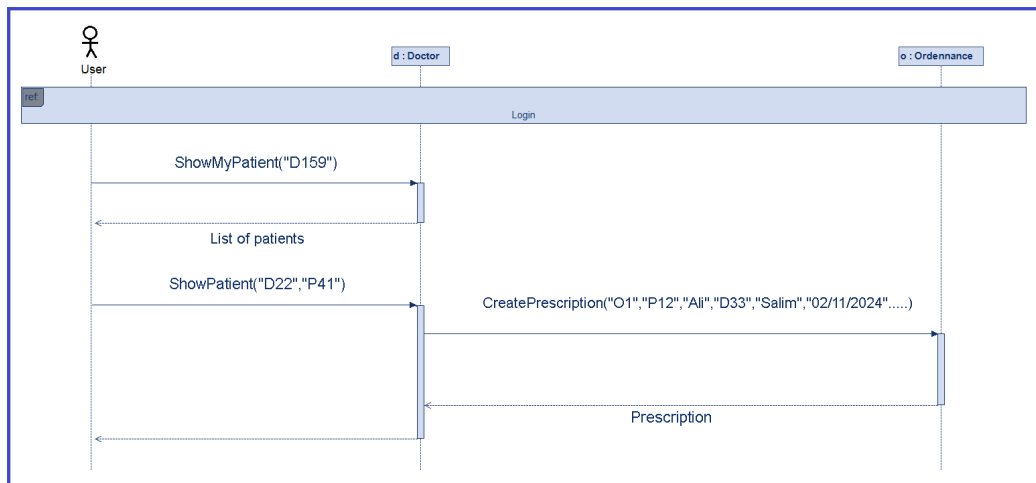


Figure 4.6: Sequence diagram: Doctor scenario: Show patients and create prescription.

Nurses have the ability to register patients and search for a specific patient in the system by his username. As shown in scenario 4.7.

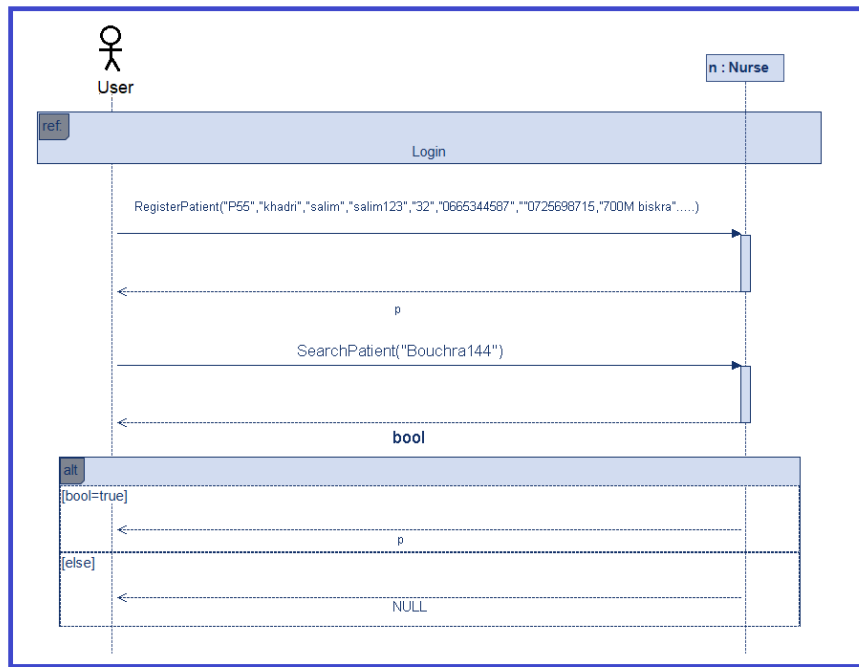


Figure 4.7: Sequence diagram: Nurse scenario: Create patient account and search patient.

The scenario 4.8 explain how patient can found doctors (by his speciality or name), and how he can view his personal prescriptions.

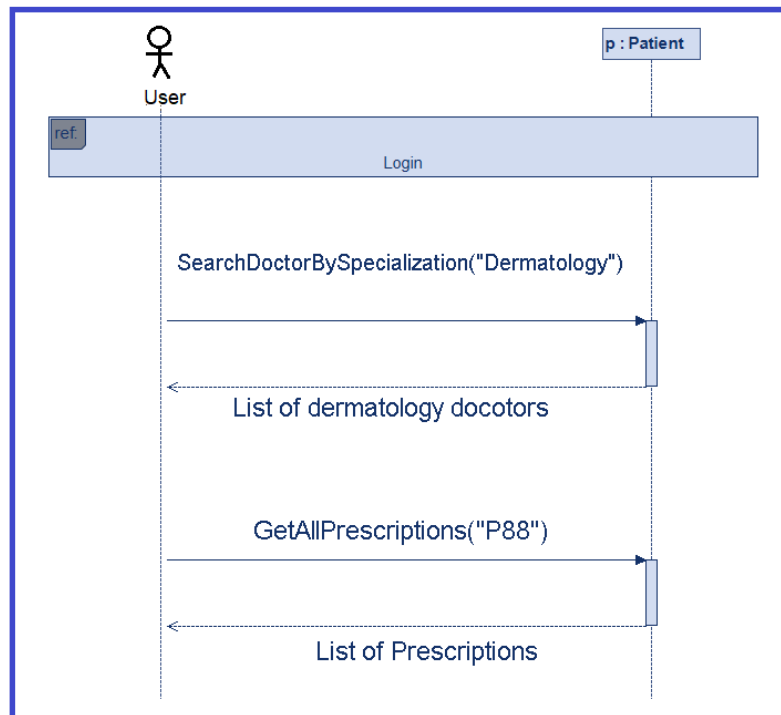


Figure 4.8: Sequence diagram: Patient scenario: Search doctors by specialization and show prescriptions.

You can find the architecture and all diagrams through this link: <https://drive.google.com/drive/saahdiagrams>

4.4 Development Environment

Our system was built on a HP laptop that equipped with:

- **Processor** Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz 1.20 GHz.
- **Installed RAM** 8.00 GB (7.69 GB usable).
- **System type** 64-bit operating system, x64-based processor.

And equipped with the following software:

- **Ubuntu 24.04** is a modern, open-source operating system based on Linux, designed for enterprise servers, desktops, cloud, and IoT devices [85].



Figure 4.9: Ubuntu.

- **Visual Studio Code (VSCode)** is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle [86].



Figure 4.10: VSCode.

- **Docker 4.37** is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly [87].



Figure 4.11: Docker.

- **Node js** is an open source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project [88].



Figure 4.12: Node js.

- **HyperText Markup Language(HTML)** is the World Wide Web's core markup language. Originally, HTML was primarily designed as a language for semantically describing scientific documents [89].
- **Cascading Style Sheets (CSS)** is a simple mechanism for adding style (e.g., fonts, colours, spacing) to Web documents [90].
- **JavaScript** is a high level, interpreted programming language commonly used to create dynamic and interactive content on websites. It is one of the core technologies of the web, alongside HTML and CSS, and is essential for building responsive, user friendly web applications. It is also used to build smart contracts for the Hyperledger Fabric framework [91].



Figure 4.13: HTML CSS and JavaScript.

4.5 Conclusion

In this chapter, we introduced the design of our system. The constructed models are developed using UML diagrams. These diagrams help to visually understand the functionality of our system. We also presented in this chapter some implementation aspects, such as the tools and environment used during the development process. In the next chapter we will present and evaluate our system.

Chapter 5

System Presentation and Evaluation

5.1 Introduction

This chapter presents our system through the interfaces of the website, and evaluation process of the developed system using selected tools to assess both functional and performance aspects, we use Postman and Hyperledger Caliper. The goal is to ensure that the system behaves as expected under various operations and to analyse key performance metrics such as response time, throughput, and network efficiency.

5.2 System Presentation

In the first section, we present our system using screenshots, which show the different functionalities available in **HOSPITALERIA**, namely:

- **Visit the home page:** the 5.1 figure represents our main interface which contains the name, logo, general information about our solution, and some information through the "About" and "Contact" sections at the bottom 5.2.

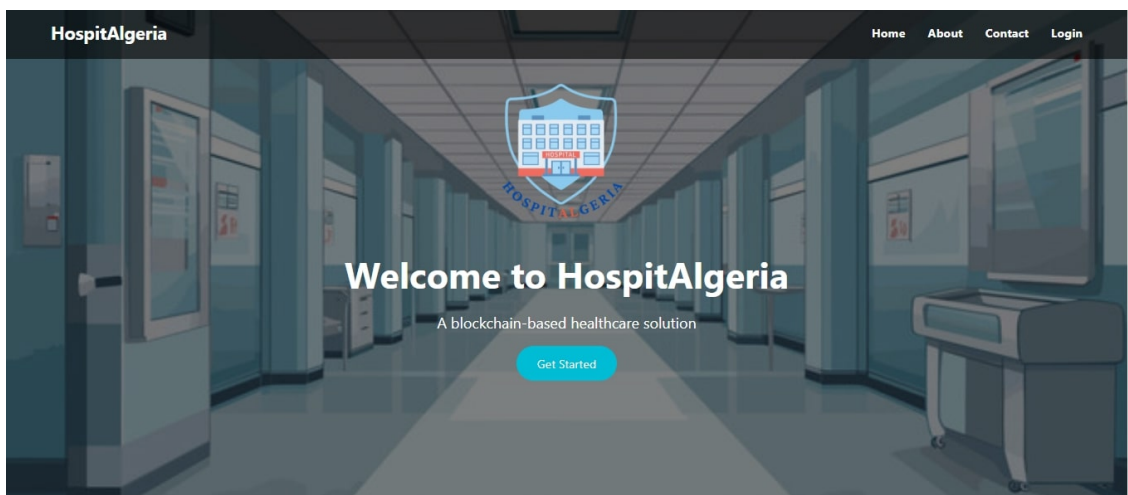


Figure 5.1: Main interface (top).

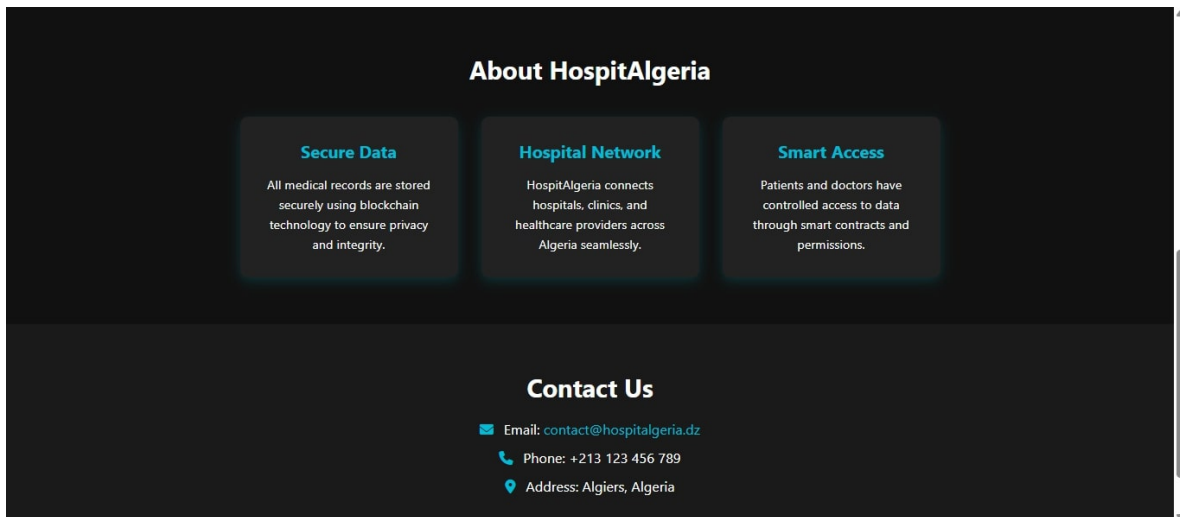


Figure 5.2: Main interface (bottom).

- **Login:** after creating the account, select the role (admin, doctor, nurse, or patient), then enter the username and password to log in to the account (figure 5.3).

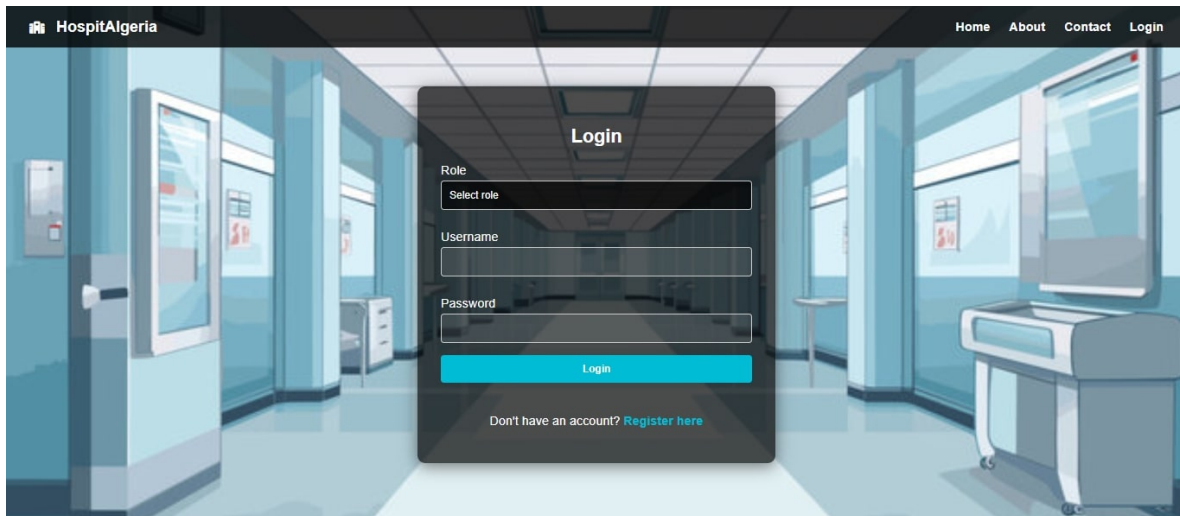


Figure 5.3: Login interface.

- **For admins:** after logging in as an administrator, he can view all lists of patients, doctors, and nurses, and can change his personal information or create new accounts for the new: administrator, doctor (figure 5.4) or nurse.

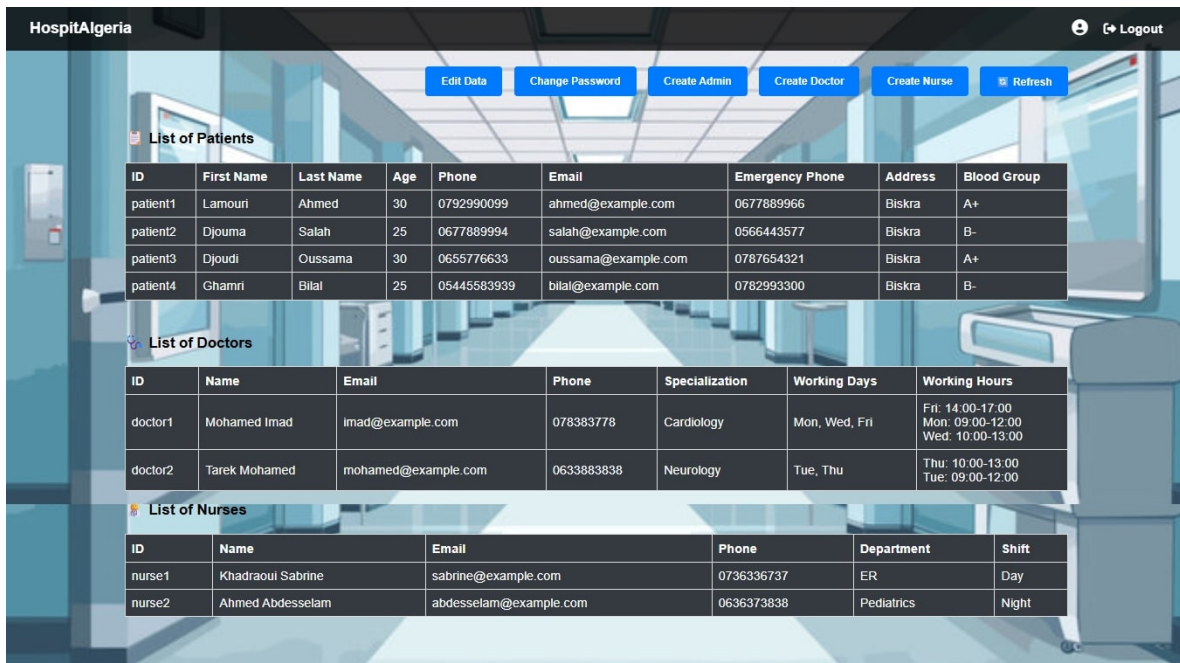


Figure 5.4: Admin interface: Hospital information.

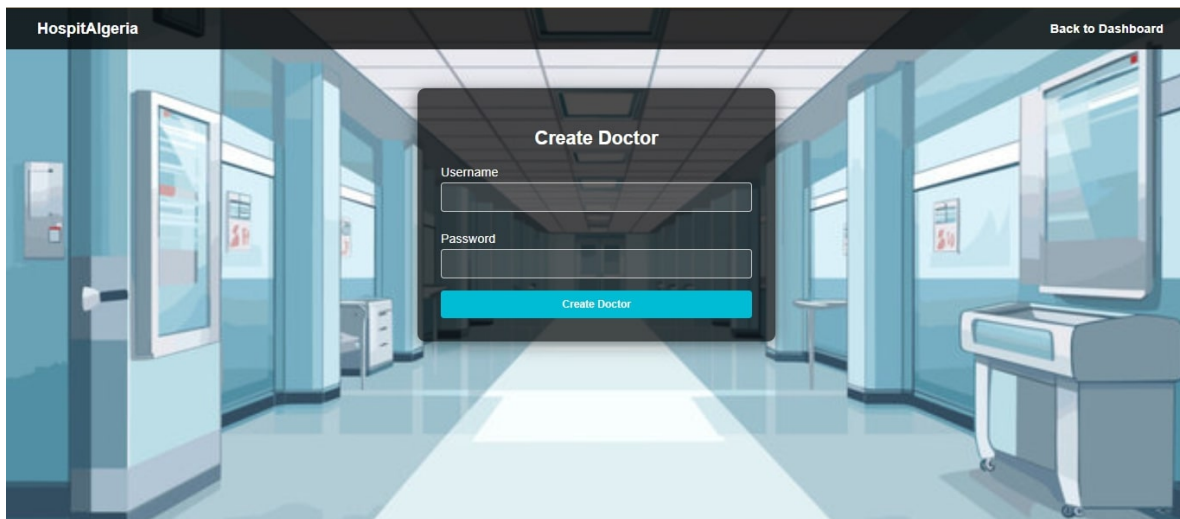


Figure 5.5: Admin interface: Create doctor account.

- **For doctors:** after logging in as a doctor, a page containing his personal information will appear like figure 5.6 show, he also can edit his profile and change his password.

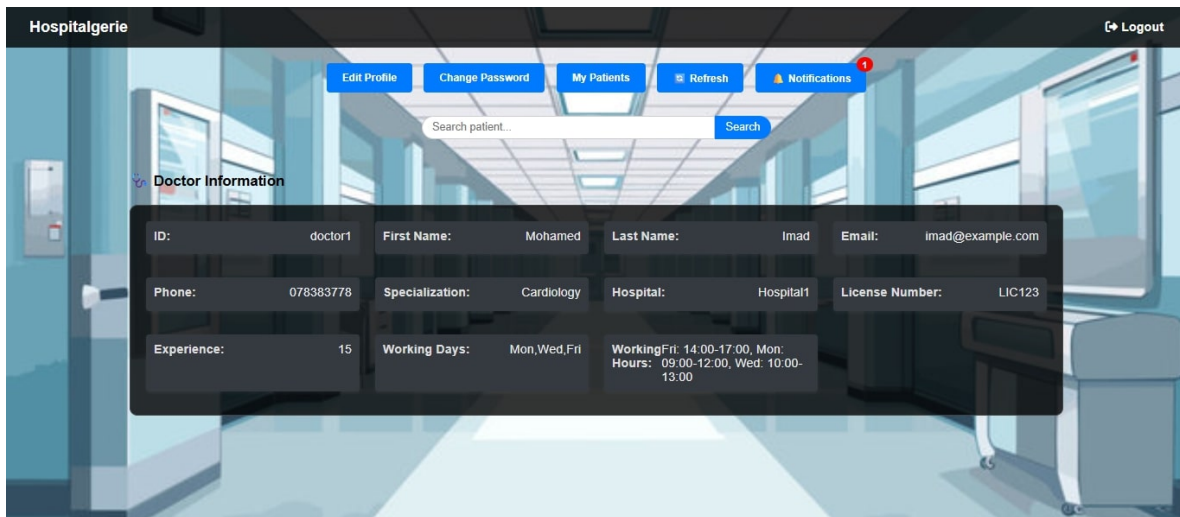


Figure 5.6: Doctor interface: Personal information.

One of the available options is 'My Patients', which contains a list of his patients, and he can choose one of them to show or edit his medical data (figure 5.7). Among the three options which are circled in red he can see: the medical history, prescriptions and reports of his patient (figure 5.8).

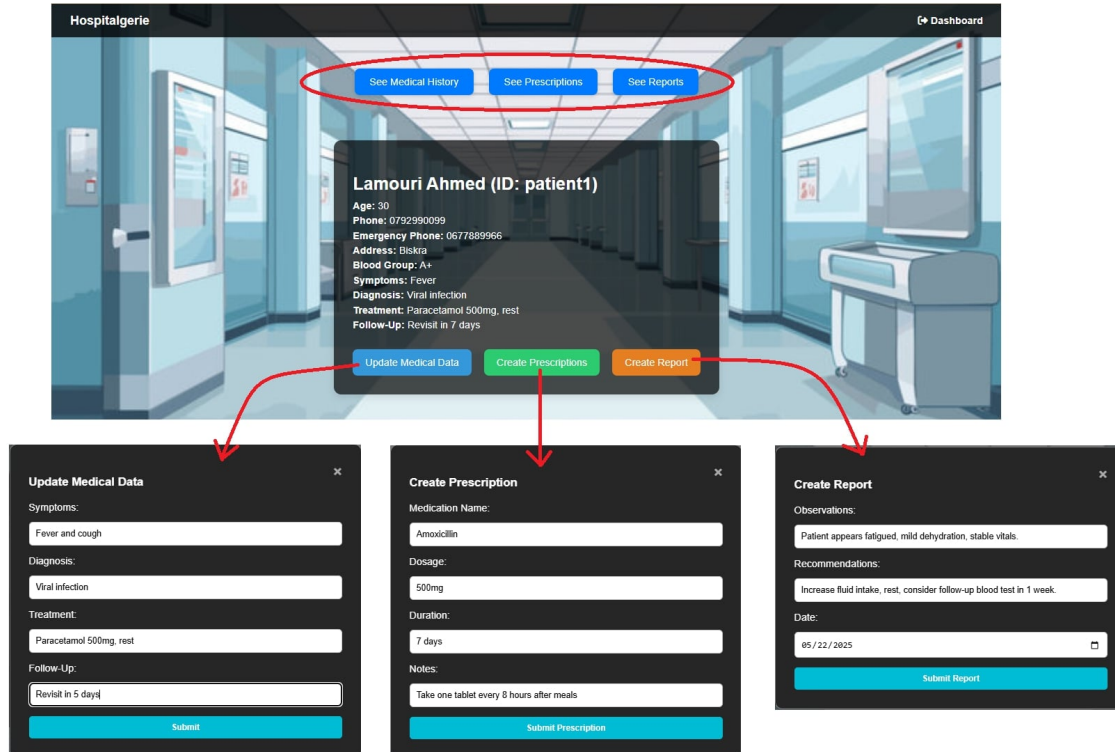


Figure 5.7: Doctor interface: Update patient medical file.

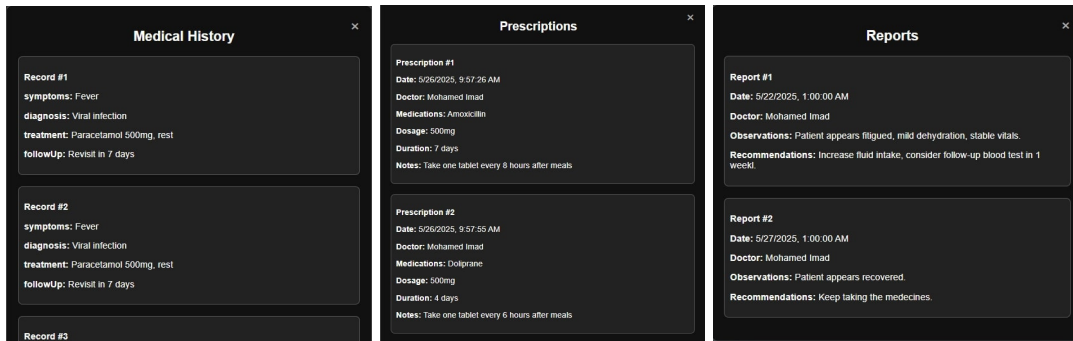


Figure 5.8: Doctor Interface: Show Patient Data

In case of abnormal rate, he receives a notification for do rapid intervention (figure 5.9).

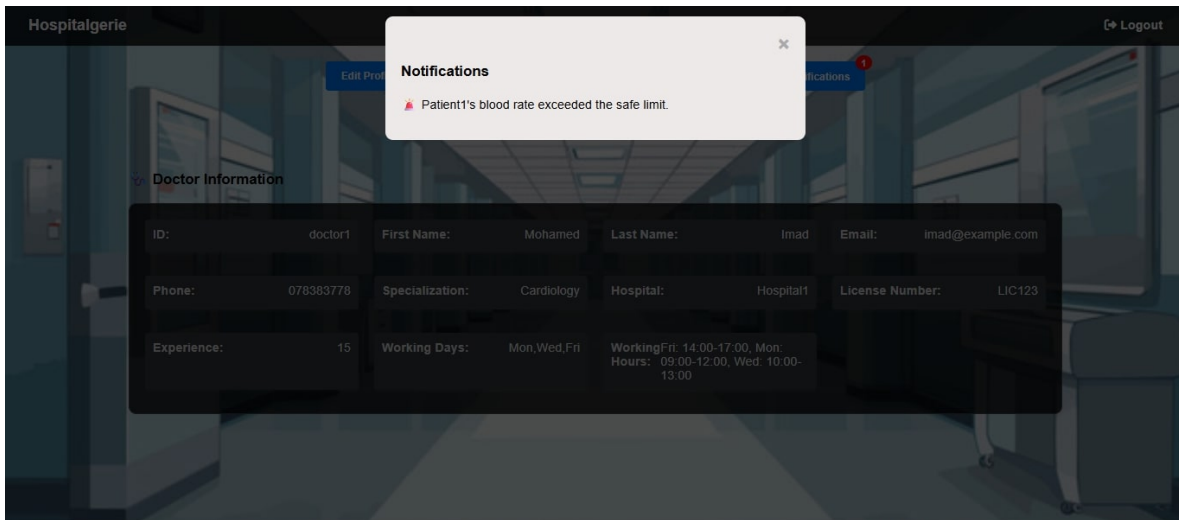


Figure 5.9: Doctor interface: Notifications.

- **For nurses:** after logging in as a nurse, he can view or edit his personal data, change his password, and search for a doctor or patient (figure 5.10). also he can create accounts for new patients (figure 5.11).

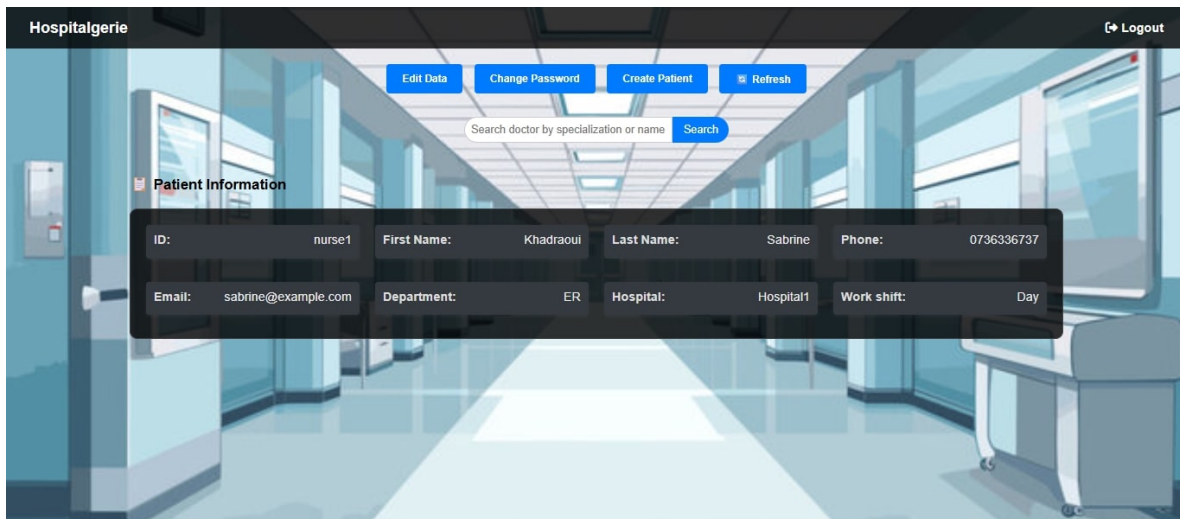


Figure 5.10: Nurse interface: Personal information.

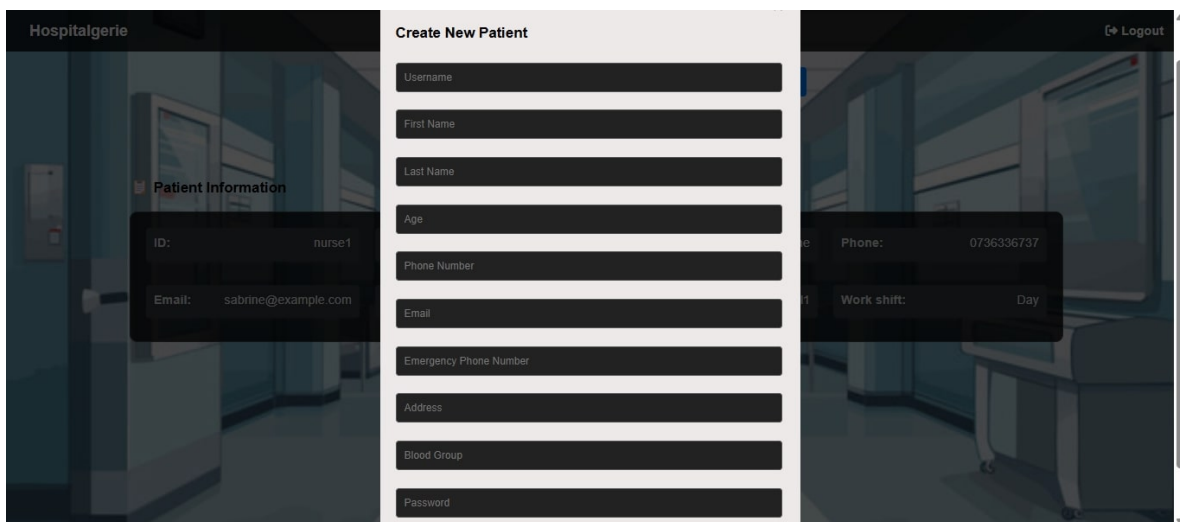


Figure 5.11: Nurse interface: Create patient account.

- **For patient:** after creating a patient account and logging in, the patient view (figure 5.13) and edit (figure 5.12) his information, and he can also view all the details and history according to his profile or medical file (figure 5.14).

The screenshot shows the 'Hospitalgerie' application interface. A modal titled 'Edit Patient Data' is open in the center, allowing for the modification of patient information. The modal contains several text input fields with the following values: 'Lamouri', 'Ahmed', '30', '0792990099', 'ahmed@example.com', '0677889966', 'Biskra', and 'A+'. A 'Save' button is located at the bottom of the modal. In the background, the 'Patient Information' form is visible, showing fields for ID, Phone, Blood Group, and Follow-up. The top navigation bar includes 'Edit Data', 'Change Password', 'Heart Rate', and 'Refresh' buttons, along with a 'Logout' link.

The screenshot displays a web application interface. At the top, a navigation bar contains links for 'Home', 'About Us', 'Contact Us', 'Services', 'Blog', and 'Sign In'. Below the navigation bar, the main content area is divided into several sections. On the left, there is a 'Patient Medical History' section. To its right is a 'Prescription Details' section. Further right is a 'Report Details' section. Red arrows originate from the navigation bar and point to these sections, illustrating the application's flow. The 'Patient Medical History' section includes a table with columns for 'Patient ID', 'Patient Name', 'Age', 'Gender', 'Blood Pressure', 'Heart Rate', 'Cholesterol', 'Glucose', 'Hemoglobin', 'Hematocrit', 'Hemoglobin A1c', 'Creatinine', 'Urea Nitrogen', 'Bilirubin', 'Alkaline Phosphatase', 'Aspartate Aminotransferase', 'Alanine Aminotransferase', 'Gamma-Glutamyl Transaminase', 'Lactate Dehydrogenase', 'Prothrombin Time', 'Partial Thromboplastin Time', 'Fibrinogen', 'D-Dimer', 'C-Reactive Protein', 'Interleukin-6', 'Interleukin-10', 'Interleukin-17', 'Interleukin-21', 'Interleukin-22', 'Interleukin-23', 'Interleukin-24', 'Interleukin-25', 'Interleukin-26', 'Interleukin-27', 'Interleukin-28', 'Interleukin-29', 'Interleukin-30', 'Interleukin-31', 'Interleukin-32', 'Interleukin-33', 'Interleukin-34', 'Interleukin-35', 'Interleukin-36', 'Interleukin-37', 'Interleukin-38', 'Interleukin-39', 'Interleukin-40', 'Interleukin-41', 'Interleukin-42', 'Interleukin-43', 'Interleukin-44', 'Interleukin-45', 'Interleukin-46', 'Interleukin-47', 'Interleukin-48', 'Interleukin-49', 'Interleukin-50', 'Interleukin-51', 'Interleukin-52', 'Interleukin-53', 'Interleukin-54', 'Interleukin-55', 'Interleukin-56', 'Interleukin-57', 'Interleukin-58', 'Interleukin-59', 'Interleukin-60', 'Interleukin-61', 'Interleukin-62', 'Interleukin-63', 'Interleukin-64', 'Interleukin-65', 'Interleukin-66', 'Interleukin-67', 'Interleukin-68', 'Interleukin-69', 'Interleukin-70', 'Interleukin-71', 'Interleukin-72', 'Interleukin-73', 'Interleukin-74', 'Interleukin-75', 'Interleukin-76', 'Interleukin-77', 'Interleukin-78', 'Interleukin-79', 'Interleukin-80', 'Interleukin-81', 'Interleukin-82', 'Interleukin-83', 'Interleukin-84', 'Interleukin-85', 'Interleukin-86', 'Interleukin-87', 'Interleukin-88', 'Interleukin-89', 'Interleukin-90', 'Interleukin-91', 'Interleukin-92', 'Interleukin-93', 'Interleukin-94', 'Interleukin-95', 'Interleukin-96', 'Interleukin-97', 'Interleukin-98', 'Interleukin-99', 'Interleukin-100'. The 'Prescription Details' section includes a table with columns for 'Prescription ID', 'Prescription Name', 'Dosage', 'Frequency', 'Duration', 'Side Effects', 'Contraindications', 'Interactions', 'Warnings', 'Precautions', 'Instructions', 'Notes'. The 'Report Details' section includes a table with columns for 'Report ID', 'Report Name', 'Date', 'Time', 'Location', 'Status', 'Comments'. The 'Patient Medical History' section includes a table with columns for 'Patient ID', 'Patient Name', 'Age', 'Gender', 'Blood Pressure', 'Heart Rate', 'Cholesterol', 'Glucose', 'Hemoglobin', 'Hematocrit', 'Hemoglobin A1c', 'Creatinine', 'Urea Nitrogen', 'Bilirubin', 'Alkaline Phosphatase', 'Aspartate Aminotransferase', 'Alanine Aminotransferase', 'Gamma-Glutamyl Transaminase', 'Lactate Dehydrogenase', 'Prothrombin Time', 'Partial Thromboplastin Time', 'Fibrinogen', 'D-Dimer', 'C-Reactive Protein', 'Interleukin-6', 'Interleukin-10', 'Interleukin-17', 'Interleukin-21', 'Interleukin-22', 'Interleukin-23', 'Interleukin-24', 'Interleukin-25', 'Interleukin-26', 'Interleukin-27', 'Interleukin-28', 'Interleukin-29', 'Interleukin-30', 'Interleukin-31', 'Interleukin-32', 'Interleukin-33', 'Interleukin-34', 'Interleukin-35', 'Interleukin-36', 'Interleukin-37', 'Interleukin-38', 'Interleukin-39', 'Interleukin-40', 'Interleukin-41', 'Interleukin-42', 'Interleukin-43', 'Interleukin-44', 'Interleukin-45', 'Interleukin-46', 'Interleukin-47', 'Interleukin-48', 'Interleukin-49', 'Interleukin-50', 'Interleukin-51', 'Interleukin-52', 'Interleukin-53', 'Interleukin-54', 'Interleukin-55', 'Interleukin-56', 'Interleukin-57', 'Interleukin-58', 'Interleukin-59', 'Interleukin-60', 'Interleukin-61', 'Interleukin-62', 'Interleukin-63', 'Interleukin-64', 'Interleukin-65', 'Interleukin-66', 'Interleukin-67', 'Interleukin-68', 'Interleukin-69', 'Interleukin-70', 'Interleukin-71', 'Interleukin-72', 'Interleukin-73', 'Interleukin-74', 'Interleukin-75', 'Interleukin-76', 'Interleukin-77', 'Interleukin-78', 'Interleukin-79', 'Interleukin-80', 'Interleukin-81', 'Interleukin-82', 'Interleukin-83', 'Interleukin-84', 'Interleukin-85', 'Interleukin-86', 'Interleukin-87', 'Interleukin-88', 'Interleukin-89', 'Interleukin-90', 'Interleukin-91', 'Interleukin-92', 'Interleukin-93', 'Interleukin-94', 'Interleukin-95', 'Interleukin-96', 'Interleukin-97', 'Interleukin-98', 'Interleukin-99', 'Interleukin-100'. The 'Prescription Details' section includes a table with columns for 'Prescription ID', 'Prescription Name', 'Dosage', 'Frequency', 'Duration', 'Side Effects', 'Contraindications', 'Interactions', 'Warnings', 'Precautions', 'Instructions', 'Notes'. The 'Report Details' section includes a table with columns for 'Report ID', 'Report Name', 'Date', 'Time', 'Location', 'Status', 'Comments'.

47

5.3 System evaluation

5.3.1 Evaluator Tools

Two main evaluators are employed: Postman, for testing and validating API endpoints, and Hyperledger Caliper, to measure the performance of the blockchain network:

5.3.1.1 Postman

Postman is a comprehensive API platform designed to simplify and streamline each step of the API lifecycle. It enables developers to design, build, test, and collaborate on APIs efficiently. With features like API repositories, collaborative workspaces, automated testing, and documentation tools, Postman supports both individual developers and teams in creating better APIs faster [92].



Figure 5.15: Postman.

5.3.1.2 Hyperledger Caliper

Hyperledger Caliper is a blockchain benchmark tool, it allows users to measure the performance of a blockchain implementation with a set of predefined use cases. Hyperledger Caliper will produce reports containing a number of performance indicators to serve as a reference when using the following blockchain solutions: Hyperledger Besu, Ethereum and Hyperledger Fabric [93].



Figure 5.16: Hyperledger Caliper.

5.3.2 Interpretation Results

5.3.2.1 Performance of API

We use Postman to validate the functionality of our API in the following cases:

- **PUT Request (Update Admin Details)**
 - **Request Type:** PUT.
 - **Request Body:** JSON with admin details (e.g., username, firstName, age, etc.).
 - **Status Code:** 200 OK (circled in red).

- **Response Message:** "Admin data updated successfully".

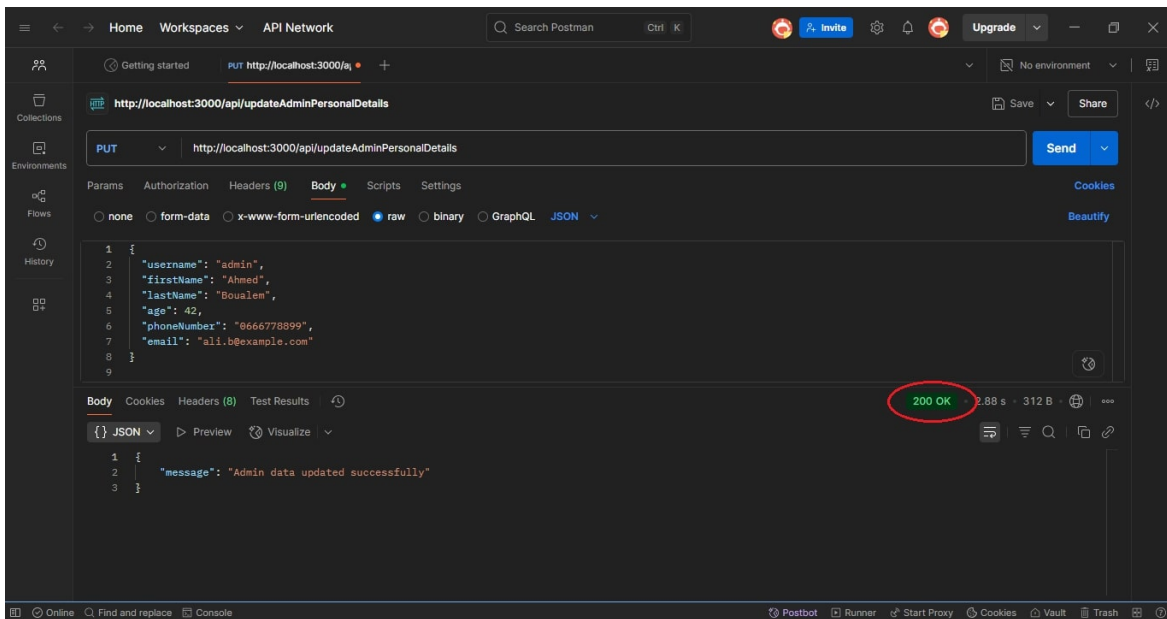


Figure 5.17: Update admin details.

- **GET Request (Retrieve Doctor List)**

- **Request Type:** GET.
- **Request Body:** JSON list of doctors with details (ID, name, specialization, working days and hours).
- **Status Code:** 200 OK (circled in red).

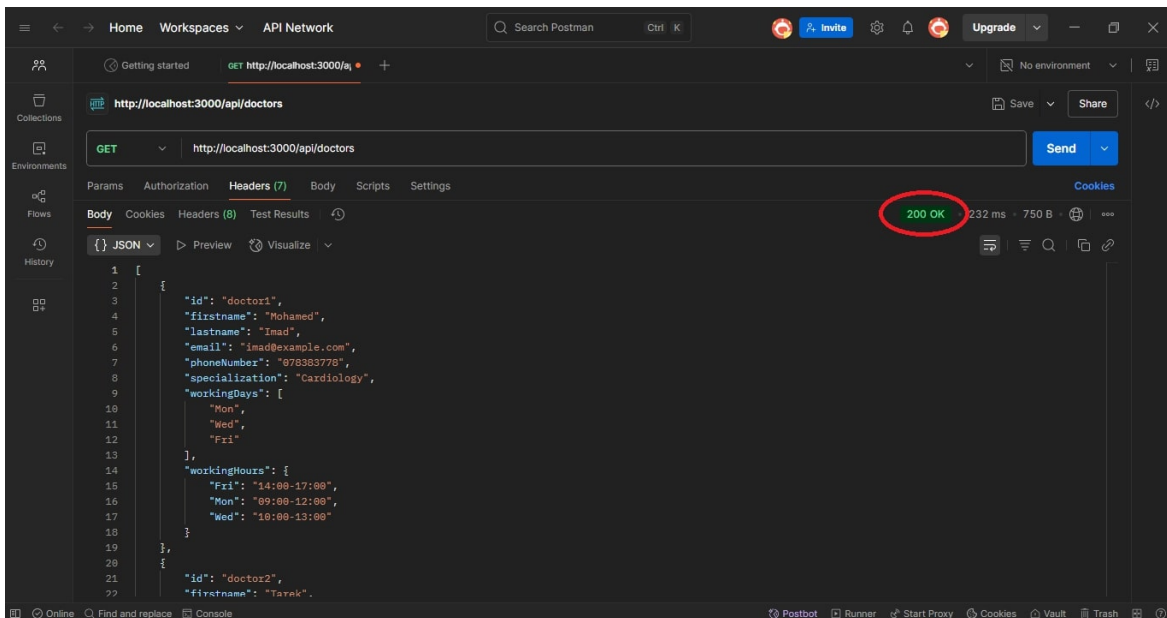


Figure 5.18: Retrieve doctor list.

Through the two screenshots result 5.17 and 5.18, our API **correctly** handled the update request and returned a success status, indicating the changes (e.g., name, age, phone number) were accepted. and **correctly** fetched the list of doctors and their schedules, indicating proper database interaction and serialization. and the 200 OK responses confirm that:

- The update operation works.
- The data retrieval endpoint functions correctly.
- Our API handles both modification and fetching of healthcare personnel data without errors.

5.3.2.2 Performance of Network

To evaluate the performance of the proposed eHealth blockchain network, a benchmark test was conducted using **Hyperledger Caliper**, a widely adopted tool for measuring the performance of blockchain systems. The goal was to assess the responsiveness, stability, and scalability of the network during frequent read operations.

- **Benchmark Scenario**

The test focused on the **ReadDoctor** operation, which simulates reading a doctor's data from the ledger. This operation is critical in healthcare systems where users (e.g., patients, nurses, and other doctors) frequently access doctor information. The test environment consisted of a Hyperledger Fabric network with the following configuration:

- **Channel:** mychannel
- **Smart Contract:** basic
- **Organization:** Hosp1MSP
- **Client Identity:** User1
- **Workers:** 2 concurrent clients
- **Test Duration:** 30 seconds
- **Transaction Rate:** 100 transactions per second per worker
- **Tested Assets:** 10 doctor entries

- **Results**

The benchmark executed a total of **4,138 read transactions**, all of which were successfully processed with **zero failures**, indicating excellent stability and correctness of the chaincode logic.

- **Success Rate:** 100% (4,138/4,138)
- **Latency:**
 - * Minimum: 0.01 seconds
 - * Maximum: 0.13 seconds
 - * Average: 0.02 seconds

- **Send Rate:** 139.7 transactions per second (TPS)
- **Throughput:** 139.7 TPS

• Analysis

The observed performance metrics reflect a highly responsive and stable network. The low latency indicates minimal delay in processing read requests, likely due to a simple endorsement policy and low system congestion. Furthermore, the perfect success rate and matching send rate and throughput confirm that the system can maintain high availability and responsiveness under moderate load conditions.

These results validate the effectiveness of the Hyperledger Fabric based architecture for real time healthcare applications (show figures 5.19, 5.20 and 5.21).

```
nidou@nidou:~/go/src/github.com/min/caliper-workspace$ npx caliper launch manager \
--caliper-workspace /
--caliper-networkconfig networks/networkConfig.yaml \
--caliper-benchmarkconfig benchmarks/myAssetBenchmark.yaml \
--caliper-flow-only-test
2025.05.28-12:48:43.240 info [caliper] [cli-launch-manager] Set workspace path: /home/nidou/go/src/github.com/min/caliper-workspace
2025.05.28-12:48:43.242 info [caliper] [cli-launch-manager] Set benchmark configuration path: /home/nidou/go/src/github.com/min/caliper-workspace/benchmarks/myAssetBenchmark.yaml
2025.05.28-12:48:43.242 info [caliper] [cli-launch-manager] Set network configuration path: /home/nidou/go/src/github.com/min/caliper-workspace/networks/networkConfig.yaml
2025.05.28-12:48:43.244 info [caliper] [cli-launch-manager] Set SUT type: fabric
2025.05.28-12:48:43.254 info [caliper] [benchmark-validator] No observer specified, will default to 'none'
2025.05.28-12:48:43.355 info [caliper] [caliper-engine] Starting benchmark flow
2025.05.28-12:48:43.355 info [caliper] [caliper-engine] Skipping start commands due to benchmark flow conditioning
2025.05.28-12:48:43.356 info [caliper] [caliper-engine] Skipping initialization phase due to benchmark flow conditioning
2025.05.28-12:48:43.356 info [caliper] [caliper-engine] Skipping install smart contract phase due to benchmark flow conditioning
2025.05.28-12:48:43.347 info [caliper] [FabricConnectorFactory] Initializing peer gateway connector compatible with installed fabric-gateway SDK: 1.5.0
2025.05.28-12:48:43.499 info [caliper] [IdentityManager] Adding User1 (admin=false) as User1 for organization HsplaMSP
2025.05.28-12:48:44.007 info [caliper] [monitor.js] No resource monitors specified
2025.05.28-12:48:44.010 info [caliper] [default-observer] Observer interval set to 5000 milliseconds
2025.05.28-12:48:44.016 info [caliper] [round-orchestrator] Preparing worker connections
2025.05.28-12:48:44.017 info [caliper] [worker-orchestrator] Launching worker 1 of 2
2025.05.28-12:48:44.025 info [caliper] [worker-orchestrator] Launching worker 2 of 2
2025.05.28-12:48:44.028 info [caliper] [worker-orchestrator] Messenger not configured, entering configure phase...
2025.05.28-12:48:44.030 info [caliper] [worker-orchestrator] No existing workers detected, entering worker launch phase...
2025.05.28-12:48:44.030 info [caliper] [worker-orchestrator] Waiting for 2 workers to be connected...
2025.05.28-12:48:45.233 info [caliper] [cli-launch-worker] Set workspace path: /home/nidou/go/src/github.com/min/caliper-workspace
2025.05.28-12:48:45.254 info [caliper] [cli-launch-worker] Set benchmark configuration path: /home/nidou/go/src/github.com/min/caliper-workspace/benchmarks/myAssetBenchmark.yaml
2025.05.28-12:48:45.255 info [caliper] [cli-launch-worker] Set network configuration path: /home/nidou/go/src/github.com/min/caliper-workspace/networks/networkConfig.yaml
2025.05.28-12:48:45.255 info [caliper] [cli-launch-worker] Set SUT type: fabric
2025.05.28-12:48:45.257 info [caliper] [cli-launch-worker] Set workspace path: /home/nidou/go/src/github.com/min/caliper-workspace
2025.05.28-12:48:45.258 info [caliper] [cli-launch-worker] Set benchmark configuration path: /home/nidou/go/src/github.com/min/caliper-workspace/benchmarks/myAssetBenchmark.yaml
2025.05.28-12:48:45.258 info [caliper] [cli-launch-worker] Set network configuration path: /home/nidou/go/src/github.com/min/caliper-workspace/networks/networkConfig.yaml
2025.05.28-12:48:45.258 info [caliper] [cli-launch-worker] Set SUT type: fabric
2025.05.28-12:48:45.259 info [caliper] [worker-orchestrator] 2 workers connected, progressing to worker assignment phase.
2025.05.28-12:48:45.259 info [caliper] [worker-orchestrator] Workers currently unassigned, awaiting index assignment...
2025.05.28-12:48:45.259 info [caliper] [worker-orchestrator] Waiting for 2 workers to be assigned...
2025.05.28-12:48:45.259 info [caliper] [worker-orchestrator] 2 workers assigned, progressing to worker initialization phase.
2025.05.28-12:48:46.148 info [caliper] [worker-orchestrator] Waiting for 2 workers to be ready...
2025.05.28-12:48:46.151 info [caliper] [worker-message-handler] Initializing Worker#1...
2025.05.28-12:48:46.151 info [caliper] [worker-message-handler] Initializing Worker#0...
2025.05.28-12:48:46.151 info [caliper] [FabricConnectorFactory] Initializing peer gateway connector compatible with installed fabric-gateway SDK: 1.5.0
2025.05.28-12:48:47.002 info [caliper] [FabricConnectorFactory] Initializing peer gateway connector compatible with installed fabric-gateway SDK: 1.5.0
2025.05.28-12:48:47.008 info [caliper] [IdentityManager] Adding User1 (admin=false) as User1 for organization HsplaMSP
2025.05.28-12:48:47.012 info [caliper] [worker-message-handler] Worker#0 initialized
2025.05.28-12:48:47.019 info [caliper] [IdentityManager] Adding User1 (admin=false) as User1 for organization HsplaMSP
2025.05.28-12:48:47.022 info [caliper] [worker-message-handler] Worker#1 initialized
2025.05.28-12:48:47.023 info [caliper] [worker-message-handler] 2 workers ready, progressing to test preparation phase.
2025.05.28-12:48:47.024 info [caliper] [round-orchestrator] Started round 1 (readAsset)
2025.05.28-12:48:47.026 info [caliper] [worker-message-handler] Preparing Worker#0 for Round#0
```

Figure 5.19: Hyperledger Caliper results 1.

```
2025.05.28-12:48:47.075 info [caliper] [caliper-worker] Info: worker 1 prepare test phase for round 0 is starting...
Registering doctor with username: doctor_1_0
Registering doctor with username: doctor_0_0
Registering doctor with username: doctor_0_1
Registering doctor with username: doctor_1_1
Registering doctor with username: doctor_0_2
Registering doctor with username: doctor_1_2
Registering doctor with username: doctor_1_3
Registering doctor with username: doctor_0_3
Registering doctor with username: doctor_0_4
Registering doctor with username: doctor_1_4
Registering doctor with username: doctor_0_5
Registering doctor with username: doctor_1_5
Registering doctor with username: doctor_0_6
Registering doctor with username: doctor_1_6
Registering doctor with username: doctor_1_7
Registering doctor with username: doctor_0_7
Registering doctor with username: doctor_1_8
Registering doctor with username: doctor_0_8
Registering doctor with username: doctor_0_9
Registering doctor with username: doctor_1_9
2025.05.28-12:49:10.179 info [caliper] [connectors/peer-gateway/PeerGateway] disconnecting gateway for user User1
2025.05.28-12:49:10.180 info [caliper] [connectors/peer-gateway/PeerGateway] disconnecting gateway for user User1
2025.05.28-12:49:10.180 info [caliper] [connectors/peer-gateway/PeerGateway] disconnecting gRPC client at peer peer0.hspl.example.com
2025.05.28-12:49:10.183 info [caliper] [caliper-worker] Info: worker 1 prepare test phase for round 0 is completed
2025.05.28-12:49:10.183 info [caliper] [worker-message-handler] Worker#1 prepared for Round#0
2025.05.28-12:49:10.183 info [caliper] [caliper-worker] Info: worker 0 prepare test phase for round 0 is completed
2025.05.28-12:49:10.183 info [caliper] [worker-message-handler] Worker#0 prepared for Round#0
2025.05.28-12:49:10.194 info [caliper] [round-orchestrator] 2 workers prepared, progressing to test phase.
2025.05.28-12:49:10.196 info [caliper] [round-orchestrator] Monitors successfully started
2025.05.28-12:49:10.200 info [caliper] [worker-message-handler] Worker#0 is starting Round#0
2025.05.28-12:49:10.200 info [caliper] [worker-message-handler] Worker#1 is starting Round#0
2025.05.28-12:49:10.209 info [caliper] [connectors/peer-gateway/PeerGateway] Generating contract map for user User1
2025.05.28-12:49:10.213 info [caliper] [connectors/peer-gateway/PeerGateway] Generating contract map for user User1
2025.05.28-12:49:10.213 info [caliper] [caliper-worker] Worker #0 starting workload loop
2025.05.28-12:49:10.213 info [caliper] [caliper-worker] Worker #1 starting workload loop
2025.05.28-12:49:10.213 info [caliper] [default-observer] [readAsset Round 0 Transaction Info] - Submitted: 928 Succ: 928 Fail:0 Unfinished:2
2025.05.28-12:49:10.217 info [caliper] [default-observer] [readAsset Round 0 Transaction Info] - Submitted: 1027 Succ: 1027 Fail:0 Unfinished:3
2025.05.28-12:49:10.217 info [caliper] [default-observer] [readAsset Round 0 Transaction Info] - Submitted: 1716 Succ: 1716 Fail:0 Unfinished:0
2025.05.28-12:49:10.217 info [caliper] [default-observer] [readAsset Round 0 Transaction Info] - Submitted: 2449 Succ: 2449 Fail:0 Unfinished:4
2025.05.28-12:49:10.217 info [caliper] [default-observer] [readAsset Round 0 Transaction Info] - Submitted: 3170 Succ: 3170 Fail:0 Unfinished:2
2025.05.28-12:49:10.217 info [caliper] [default-observer] [readAsset Round 0 Transaction Info] - Submitted: 3972 Succ: 3972 Fail:0 Unfinished:2
Deleting doctor with username: doctor_1_0
Deleting doctor with username: doctor_0_0
Deleting doctor with username: doctor_0_1
Deleting doctor with username: doctor_1_1
2025.05.28-12:49:45.196 info [caliper] [default-observer] Resetting txCount indicator count
```

Figure 5.20: Hyperledger Caliper results 2.

```

Deleting doctor with username: doctor_1.4
2025.05.28-12:59:59.196 info [caliper] [default-observer] [readAsset Round 0 Transaction Info] - Submitted: 0 Succ: 0 Fail:0 Unfinished:0
Deleting doctor with username: doctor_0.5
Deleting doctor with username: doctor_1.5
Deleting doctor with username: doctor_1.6
Deleting doctor with username: doctor_0.6
2025.05.28-12:59:55.200 info [caliper] [default-observer] [readAsset Round 0 Transaction Info] - Submitted: 0 Succ: 0 Fail:0 Unfinished:0
Deleting doctor with username: doctor_0.7
Deleting doctor with username: doctor_1.7
Deleting doctor with username: doctor_0.8
Deleting doctor with username: doctor_1.8
2025.05.28-12:59:00.844 info [caliper] [default-observer] [readAsset Round 0 Transaction Info] - Submitted: 0 Succ: 0 Fail:0 Unfinished:0
Deleting doctor with username: doctor_0.9
Deleting doctor with username: doctor_1.9
2025.05.28-12:59:04.011 info [caliper] [connectors/peer-gateway/PeerGateway] disconnecting gateway for user User1
2025.05.28-12:59:04.011 info [caliper] [connectors/peer-gateway/PeerGateway] disconnecting gRPC client at peer peer0.hospl.example.com
2025.05.28-12:59:04.012 info [caliper] [worker-message-handler] Worker#1 finished Round#0
2025.05.28-12:59:04.015 info [caliper] [connectors/peer-gateway/PeerGateway] disconnecting gateway for user User1
2025.05.28-12:59:04.016 info [caliper] [connectors/peer-gateway/PeerGateway] disconnecting gRPC client at peer peer0.hospl.example.com
2025.05.28-12:59:04.016 info [caliper] [worker-message-handler] Worker#0 finished Round#0
2025.05.28-12:59:09.027 info [caliper] [default-observer] [readAsset Round 0 Transaction Info] - Submitted: 0 Succ: 0 Fail:0 Unfinished:0
2025.05.28-12:59:09.029 info [caliper] [report-builder] ### Test result ###
2025.05.28-12:59:09.056 info [caliper] [report-builder]

Name Succ Fail Send Rate (TPS) Max Latency (s) Min Latency (s) Avg Latency (s) Throughput (TPS)
readAsset 4138 0 139.7 0.13 0.01 0.02 139.7

2025.05.28-12:59:09.062 info [caliper] [round-orchestrator] Finished round 1 (readAsset) in 30.122 seconds
2025.05.28-12:59:09.063 info [caliper] [monitor.js] Stopping all monitors
2025.05.28-12:59:09.064 info [caliper] [report-builder] ### All test results ###
2025.05.28-12:59:09.065 info [caliper] [report-builder]

Name Succ Fail Send Rate (TPS) Max Latency (s) Min Latency (s) Avg Latency (s) Throughput (TPS)
readAsset 4138 0 139.7 0.13 0.01 0.02 139.7

2025.05.28-12:59:09.098 info [caliper] [report-builder] Generated report with path /home/midou/go/src/github.com/min/caliper-workspace/report.html
2025.05.28-12:59:09.099 info [caliper] [monitor.js] Stopping all monitors
2025.05.28-12:59:09.100 info [caliper] [worker-orchestrator] Sending exit message to connected workers
2025.05.28-12:59:09.101 info [caliper] [round-orchestrator] Benchmark finished in 82.077 seconds, Total rounds: 1, Successful rounds: 1, Failed rounds: 0.
2025.05.28-12:59:09.101 info [caliper] [caliper-engine] Skipping end command due to benchmark flow conditioning
2025.05.28-12:59:09.101 info [caliper] [worker-message-handler] Worker#0 is exiting
2025.05.28-12:59:09.101 info [caliper] [cli-launch-manager] Benchmark successfully finished!
2025.05.28-12:59:09.101 info [caliper] [worker-message-handler] Worker#1 is exiting
midou@midou:~/go/src/github.com/min/caliper-workspace$

```

Figure 5.21: Hyperledger Caliper results 3.

We use for results visualization **Chart.js**, is a free, open source JavaScript library. It supports eight chart types: bar, line, area, pie (doughnut), bubble, radar, polar, and scatter. It is rendered using HTML5 canvas elements, and it is widely recognized as one of the best data visualization libraries [94]. The figure 5.22 represents the results of the Hyperledger Caliper evaluation plotted by the chart.js library.



Figure 5.22: Hyperledger Caliper Results Graphically.

5.4 Conclusion

The evaluation results demonstrate that the system performs reliably in both application layer API interactions and blockchain network operations. Postman testing confirmed the correctness and responsiveness of API endpoints, while Hyperledger Caliper provided insights into the network's transaction handling and latency. Together, these tools validate the system's effectiveness and support its readiness for real world deployment, particularly in security-sensitive environments like healthcare.

Chapter 6

Conclusion and Perspectives

This project focused on enhancing the security and integrity of healthcare data in an e-Health context using **Hyperledger Fabric**, a permissioned blockchain platform. Instead of relying on real sensors, we simulated health related data to represent the kind of information typically generated in Wireless Body Area Networks (WBAN).

The simulated data was integrated into the blockchain system to demonstrate how Hyperledger Fabric can ensure **confidentiality**, **integrity**, and **traceability**. Through smart contracts (chain-code), the system enforces access control policies and verifies the validity of records, reducing the risk of data tampering or unauthorized access.

The decentralized architecture guarantees data immutability and enables secure interaction between healthcare stakeholders such as doctors, hospitals, and administrators. This highlights the strong potential of blockchain technology to address key challenges in healthcare data management.

Several future directions can be considered to extend this work:

- **Integration of real sensors:**

Incorporate actual WBAN devices to collect real-time data and automatically record it on the blockchain.

- **Scalability evaluation:**

Test the blockchain network under higher data loads to evaluate performance in a broader healthcare ecosystem.

- **Advanced privacy techniques:**

Implement methods such as zero knowledge proofs or homomorphic encryption to provide stronger protection for sensitive patient data.

This project demonstrates that **Hyperledger Fabric** provides a robust foundation for building secure and transparent healthcare data systems, even when working with simulated input. It paves the way for reliable, decentralized e-Health solutions.

Bibliography

- [1] Somia Sahraoui. “Mécanismes de sécurité pour l’intégration des RCSFs à l’IoT (Internet of Things)”. PhD thesis. Université de Batna 2, 2016.
- [2] Iris ten Klooster et al. “Clarifying the Concepts of Personalization and Tailoring of eHealth Technologies: Multimethod Qualitative Study”. In: *J Med Internet Res* (13 Nov 2024). URL: <https://www.jmir.org/2024/1/e50497>.
- [3] Erika Renzi et al. “The Impact of eHealth Interventions on the Improvement of Self-Care in Chronic Patients: An Overview of Systematic Reviews”. In: *Life* 12.8 (2022). URL: <https://www.mdpi.com/2075-1729/12/8/1253>.
- [4] Somia Sahraoui et al. “Sensor-based wearable system for the detection and automatic treatment of nocturnal hypoglycaemia”. In: *Healthcare technology letters* 5.6 (2018), pp. 239–241. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC6275132/>.
- [5] Ariesta Milanti et al. “Determinants and Outcomes of eHealth Literacy in Healthy Adults: A Systematic Review”. In: *PLOS ONE* 18.10 (2023), e0291229. URL: <https://doi.org/10.1371/journal.pone.0291229>.
- [6] World Health Organization Regional Office for the Eastern Mediterranean: *Ehealth*. URL: <https://www.emro.who.int/health-topics/ehealth/>.
- [7] Maria Helena da Fonseca et al. “E-Health Practices and Technologies: A Systematic Review from 2014 to 2019”. In: *Healthcare* (10 Sep 2021). URL: <https://www.mdpi.com/2227-9032/9/9/1192>.
- [8] Chiranjeev Sanyal et al. “Economic evaluations of eHealth technologies: A systematic review”. In: *PLOS ONE* (13 June 2018). URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0198112>.
- [9] Anindya Ghose et al. “Empowering Patients Using Smart Mobile Health Platforms: Evidence From A Randomized Field Experiment”. In: (Feb 2021). arXiv: 2102.05506. URL: <https://arxiv.org/pdf/2102.05506>.
- [10] *softwarehut: Examples of eHealth Applications: Healthcare in the Digital Age*. URL: <https://softwarehut.com/blog/business/examples-of-ehealth-applications>.
- [11] *OpenMRS Implementation in Nigeria*. URL: <https://www.wired.com/2012/02/ehealth-nigeria>.

-
- [12] Thanveer Shaik et al. “Remote patient monitoring using artificial intelligence: Current state, applications, and challenges”. In: *WIREs Data Mining and Knowledge Discovery* (2023). ISSN: 1942-4795. URL: <https://arxiv.org/pdf/2301.10009>.
 - [13] Paolo Biancone et al. “E-health for the future. Managerial perspectives using a multiple case study approach”. In: *Technovation* (2023). ISSN: 0166-4972. URL: <https://www.sciencedirect.com/science/article/pii/S0166497221001875>.
 - [14] *Financial Times: Medical centres compete to achieve ‘smart hospital’ status*. URL: <https://www.ft.com/content/2805edfd-36db-4a58-b93f-411a18c6e003?utm>.
 - [15] *The Wall Street Journal: A ‘Digital Twin’ of Your Heart Lets Doctors Test Treatments Before Surgery*. URL: <https://www.wsj.com/health/healthcare/digital-heart-surgery-patient-treatment-c35ec4be?>.
 - [16] *Roots Analysis: Blockchain in Healthcare Market*. URL: <https://www.rootsanalysis.com/reports/blockchain-technology-in-healthcare-market.html>.
 - [17] Gousia Habib et al. “Blockchain Technology: Benefits, Challenges, Applications, and Integration of Blockchain Technology with Cloud Computing”. In: *Future Internet* (2022). URL: <https://www.mdpi.com/1999-5903/14/11/341>.
 - [18] Gautami Tripathi, Mohd Abdul Ahad, and Gabriella Casalino. “A comprehensive review of blockchain technology: Underlying principles and historical background with future challenges”. In: *Decision Analytics Journal* (2023), p. 100344. URL: <https://www.sciencedirect.com/science/article/pii/S2772662223001844>.
 - [19] Somia Sahraoui and Abdelmalik Bachir. “Lightweight Consensus Mechanisms in the Internet of Blockchained Things: Thorough Analysis and Research Directions”. In: *Digital Communications and Networks* (2025). URL: <https://www.sciencedirect.com/science/article/pii/S2352864824001767>.
 - [20] Janak Damre et al. “BLOCKCHAIN: TYPES AND BENEFITS”. In: *INTERNATIONAL JOURNAL OF CURRENT SCIENCE* (2022). URL: <https://rjpn.org/ijcspub/papers/IJCSP22A1066.pdf>.
 - [21] PK Paul et al. “Blockchain Technology and its Types—A Short Review”. In: *International Journal of Applied Science and Engineering* (December 2021). URL: https://www.researchgate.net/publication/359051731_Blockchain_Technology_and_its_Types-A_Short_Review.
 - [22] Noah Fields. *komodoplatform: 4 Types of Blockchain Technology Explained*. URL: <https://komodoplatform.com/en/academy/blockchain-technology-types/>.
 - [23] GANESAN SUBRAMANIAN et al. “Crypto Pharmacy – Digital Medicine: A Mobile Application Integrated With Hybrid Blockchain to Tackle the Issues in Pharma Supply Chain”. In: *IEEE open journal of computer society* (5 January 2021). URL: https://www.researchgate.net/publication/348254198_Crypto_Pharmacy_-_Digital_Medicine_A_Mobile_Application_Integrated_With_Hybrid_Blockchain_to_Tackle_the_Issues_in_Pharma_Supply_Chain.

- [24] Christine Campbell. *TechTarget: What are the 4 different types of blockchain technology?* Accessed: 2025-04-28 11:19. 31 Mars 2025. URL: <https://www.techtarget.com/searchcio/feature/What-are-the-4-different-types-of-blockchain-technology>.
- [25] Reham El-Gazzar, Karen Stendal, and Theodore Zahariadis. "Blockchain Technology and Database Management System". In: *Blockchain Technology for Data Privacy Management*. Springer, 2021, pp. 19–39. DOI: 10.1007/978-3-030-71788-9_2. URL: https://link.springer.com/chapter/10.1007/978-3-030-71788-9_2.
- [26] Nabil El Ioini, Housseem El Bitar, and Schahram Dustdar. "A survey on the integration of blockchains and databases". In: *Computer Science Review* 48 (2023), p. 100547. DOI: 10.1016/j.cosrev.2023.100547. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10124707/>.
- [27] Weichu Deng, Teng Huang, and Haiyang Wang. "A Review of the Key Technology in a Blockchain Building Decentralized Trust Platform". In: *Mathematics* 11 (2023). URL: <https://www.mdpi.com/2227-7390/11/1/101>.
- [28] *Systemdesignschool: Understanding Peer-to-Peer Architecture*. URL: <https://systemdesignschool.io/blog/peer-to-peer-architecture>.
- [29] Nader Medhat. *Medium: Understanding CAP theorem*. Feb 24, 2021. URL: <https://medium.com/nerd-for-tech/understand-cap-theorem-751f0672890e>.
- [30] Nader Medhat. *Cedanet Pty Ltd: The fallacies of distributed computing*. 2023. URL: <https://cedanet.com.au/contact.php>.
- [31] Bhabendu Kumar Mohanta et al. "Blockchain technology: A survey on applications and security privacy Challenges". In: *Internet of Things* 8 (2019), p. 100107. ISSN: 2542-6605. DOI: <https://doi.org/10.1016/j.iot.2019.100107>. URL: <https://www.sciencedirect.com/science/article/pii/S2542660518300702>.
- [32] Bharati Naikwadi et al. "A Systematic Review of Blockchain Technology and Its Applications". In: Dec. 2021. ISBN: 9781643682167. DOI: 10.3233/APC210230.
- [33] Chun Chen. *Blockchain: Research and Applications*. PhD Research Contributions. Hangzhou, China: Zhejiang University College of Computer Science and Technology, 2023. URL: <https://www.sciencedirect.com/journal/blockchain-research-and-applications>.
- [34] Sezer Bozkus Kahyaoglu and Vahap Tecim, eds. *Exploring Blockchain Applications: Management Perspectives*. Boca Raton, FL: CRC Press, 2024. ISBN: 9781032485393. URL: <https://www.routledge.com/Exploring-Blockchain-Applications-Management-Perspectives/Kahyaoglu-Tecim/p/book/9781032485393>.
- [35] Elli Androulaki, Artem Barger, et al. "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains". In: *arXiv* (2018). URL: <https://arxiv.org/pdf/1801.10228>.
- [36] *BIGLAUNCH: 101blockchains*. URL: <https://101blockchains.com/hyperledger-project/>.

- [37] Pradnya B. Patil and M. Sangeetha. “A Comprehensive Performance Analysis of a Hyperledger Fabric-powered Blockchain Network for Cross-Border Fund Transfers”. In: *Procedia Computer Science* 233 (2024). 5th International Conference on Innovative Data Communication Technologies and Application (ICIDCA 2024), pp. 723–732. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2024.03.261>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050924006215>.
- [38] Jeeta Ann Chacko, Ruben Mayer, and Hans-Arno Jacobsen. *Why Do My Blockchain Transactions Fail? A Study of Hyperledger Fabric (Extended version)**. 2021. arXiv: 2103.04681 [cs.DC]. URL: <https://arxiv.org/abs/2103.04681>.
- [39] Elli Androulaki et al. “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains”. In: *arXiv preprint arXiv:1801.10228* (2018). URL: <https://arxiv.org/pdf/1801.10228>.
- [40] *Geeksforgeeks: Hyperledger Fabric Component Design*. URL: <https://www.geeksforgeeks.org/hyperledger-fabric-component-design/>.
- [41] *Geeksforgeeks: Consensus Algorithms in Blockchain*. URL: <https://www.geeksforgeeks.org/consensus-algorithms-in-blockchain/>.
- [42] Jeeta Ann Chacko, Ruben Mayer, and Hans-Arno Jacobsen. “Why Do My Blockchain Transactions Fail? A Study of Hyperledger Fabric (Extended version)”. In: *Technical University of Munich and University of Toronto* (2025). URL: <https://arxiv.org/pdf/2103.04681>.
- [43] Ali Moin et al. “Adaptive Body Area Networks Using Kinematics and Biosignals”. In: *arXiv preprint arXiv:1807.09723* (2018). URL: <https://arxiv.org/abs/1807.09723>.
- [44] David Samuel Bhatti et al. “A Survey on Wireless Wearable Body Area Networks: A Perspective of Technology and Economy”. In: *Sensors* 22.20 (2022). URL: <https://www.mdpi.com/1424-8220/22/20/7722>.
- [45] *Medical Design Briefs*. URL: <https://www.medicaldesignbriefs.com/component/content/article/29112-passive-hardware-considerations-for-medical-body-area-network-transceivers>.
- [46] Mohammad Yaghoubi, Khandakar Ahmed, and Yuan Miao. “Wireless Body Area Network (WBAN): A Survey on Architecture, Technologies, Energy Consumption, and Security Challenges”. In: *Journal of Sensor and Actuator Networks* 11.4 (2022). URL: <https://www.mdpi.com/2224-2708/11/4/67>.
- [47] *indiamart: ECG Sensor with ECG Cable and Electrodes*. URL: <https://www.indiamart.com/proddetail/ecg-sensor-with-ecg-cable-and-electrodes-ad8232-20592992848.html>.
- [48] *iworx: Ultrasonic Blood Flow Sensors*. URL: <http://iworx.com/products/sensors/rasensors/ultrasonic-blood-flow-sensors/?v=0de7b6a61a70>.
- [49] *miliohm: Heart Rate Sensor Tutorial*. URL: <https://miliohm.com/pulse-heart-rate-sensor-tutorial/>.

-
- [50] *vernier: Blood Pressure Sensor*. URL: https://www.vernier.com/product/blood-pressure-sensor/?srsltid=AfmBOoqasDWSOGhFDszUpXH2Vm4EKBFbPHmWFKyrhDrK38Da3DANJ_xm.
 - [51] *core: Thermal Sensor*. URL: <https://corebodytemp.com/collections/products/products/core2>.
 - [52] *amazon: Respiration Rate*. URL: <https://www.amazon.com/respiratory-rate-monitor/s?k=respiratory+rate+monitor>.
 - [53] *viatomtech: KidsO2*. URL: <https://www.viatomtech.com/po4>.
 - [54] *AARP: New Glucose System Requires No Fingerstick*. URL: <https://www.aarp.org/health/drugs-supplements/fda-approves-glucose-device/>.
 - [55] Partha Roy, Pradeep Kumar, and Victor Chang. "A hybrid classifier combination for home automation using EEG signals". In: *Neural Computing and Applications* 32 (Oct. 2020). URL: https://www.researchgate.net/publication/339665310_A_hybrid_classifier_combination_for_home_automation_using_EEG_signals.
 - [56] Subhajit Chatterjee, Bhaskar Pandey, and Vinod Kumar. "Blockchain and artificial intelligence technology in e-Health". In: *Environmental Science and Pollution Research* (2021). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8412875/>.
 - [57] Muhammad Irfan, Kashif Saleem, Nauman Aslam, et al. "Blockchain integration in healthcare: A comprehensive investigation". In: *Journal of Personalized Medicine* (2023). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11082361/>.
 - [58] Tim K Mackey, Tsung-Ting Kuo, Bhaskara Gummadu, et al. "The role of blockchain technology in telehealth and telemedicine". In: *Journal of Medical Internet Research* (2021). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7842132/>.
 - [59] M. Gnanambigai and R. Arokia Renjit. "Blockchain in eHealth". In: *Proceedings of the International Conference on eKNOW 2022*. International Academy, Research, and Industry Association (IARIA). 2022. URL: https://www.iaria.org/conferences2022/fileseKNOW22/68006_EKNOW.pdf.
 - [60] Chibueze I Agbo, Qusay H Mahmoud, and JM Eklund. "Blockchain technology applications in healthcare: An overview". In: *Healthcare Analytics* (2021). URL: <https://www.sciencedirect.com/science/article/pii/S266660302100021X>.
 - [61] *Primebook*. URL: <https://www.primebook.in/>.
 - [62] Muhamed Turkanovic et al. "EduCTX: A Blockchain-Based Higher Education Credit Platform". In: *IEEE Access* (2018), pp. 5112–5127. URL: <http://dx.doi.org/10.1109/ACCESS.2018.2789929>.
 - [63] *AgriLedger*. URL: <https://agrilledger.com/>.
 - [64] *J.P.Morgan payments*. URL: <https://developer.payments.jpmorgan.com/>.
 - [65] *Business Insider*. URL: <https://www.businessinsider.com/>.

-
- [66] *Janus Henderson to follow BlackRock and Fidelity into tokenisation*. URL: <https://www.ft.com/content/648f2249-5783-4e98-8412-4056f56ad1b0>.
 - [67] Mohammad Hajian Berenjestanaki et al. "Blockchain-Based E-Voting Systems: A Technology Review". In: *Electronics* 13.1 (2024). URL: <https://www.mdpi.com/2079-9292/13/1/17>.
 - [68] *Government Technology: West Virginia Becomes First State to Test Mobile Voting by Blockchain in a Federal Election*. URL: <https://www.govtech.com/biz/west-virginia-becomes-first-state-to-test-mobile-voting-by-blockchain-in-a-federal-election.html>.
 - [69] *blockapps: Use Cases: Blockchain for Public Records and Identity Verification*. URL: <https://blockapps.net/blog/use-cases-blockchain-for-public-records-and-identity-verification/>.
 - [70] *Brookings*. URL: <https://blockapps.net/blog/use-cases-blockchain-for-public-records-and-identity-verification/>.
 - [71] *e_Estonia*. URL: <https://e-estonia.com/solutions/cyber-security/ksi-blockchain/>.
 - [72] Dr Yusuf Perwej et al. "Blockchain for Healthcare Management: Enhancing Data Security and Transparency". In: *South Eastern European Journal of Public Health* (2025). URL: <https://seejph.com/index.php/seejph/article/view/3831>.
 - [73] Jihyeon Ryu and Taeseok Kim. "Enhancing Hospital Data Security: A Blockchain-Based Protocol for Secure Information Sharing and Recovery". In: *Electronics* (2025). URL: <https://www.mdpi.com/2079-9292/14/3/580>.
 - [74] Abayomi Agbeyangi, Olukayode Oki, and Aphelele Mgidi. *Blockchain in Healthcare: Implementing Hyperledger Fabric for Electronic Health Records at Frere Provincial Hospital*. 2024. URL: <https://arxiv.org/abs/2407.15876>.
 - [75] Rahul Ganpatrao Sonkamble et al. "Secure Data Transmission of Electronic Health Records Using Blockchain Technology". In: *Electronics* (2023). URL: <https://www.mdpi.com/2079-9292/12/4/1015>.
 - [76] Samiha Fairouz et al. "A Secure Medical History Card Powered by Blockchain Technology". In: *Advances in Science, Technology and Engineering Systems Journal* (2023). URL: https://www.astesj.com/publications/ASTESJ_080611.pdf.
 - [77] Mayank Pandey et al. *Security of Healthcare Data Using Blockchains: A Survey*. 2021. URL: <https://arxiv.org/abs/2103.12326>.
 - [78] Rui Zhang, Rui Xue, and Ling Liu. *Security and Privacy for Healthcare Blockchains*. 2021. URL: <https://arxiv.org/abs/2106.06136>.
 - [79] Amit Kumar Gautam and Rakesh Kumar. "A comprehensive study on key management, authentication and trust management techniques in wireless sensor networks". In: *SN Applied Sciences* 3.1 (2021), p. 50.
 - [80] Sm Rakibul Hasan Remu et al. "Naive Bayes based Trust management model for wireless body area networks". In: *Proceedings of the international conference on computing advancements*. 2020, pp. 1–4.

- [81] Sameer Farooq, Deepak Prashar, and Kiran Jyoti. “Hybrid encryption algorithm in wireless body area networks (WBAN)”. In: *Intelligent Communication, Control and Devices: Proceedings of ICICCD 2017*. Springer. 2018, pp. 401–410.
- [82] Seyed Mahmood Hashemi. “Secure routing of WBAN with Monarchy Butterfly Optimization”. In: *Proceedings of the 2017 2nd International Conference on Communication and Information Systems*. 2017, pp. 155–158.
- [83] *Unified Modeling Language: What is UML?* URL: <https://www.uml.org/>.
- [84] *Altkom software: UML*. URL: <https://www.altkomsoftware.com/blog/use-case-diagrams-an-introduction/>.
- [85] *Ubuntu*. URL: <https://ubuntu.com/>.
- [86] *Visual Studio Code*. URL: <https://code.visualstudio.com/>.
- [87] *Docker*. URL: <https://www.docker.com/>.
- [88] *Node js*. URL: <https://nodejs.org/en>.
- [89] *HTML*. URL: <https://html.spec.whatwg.org/>.
- [90] *Cascading Style Sheets*. URL: <https://www.w3.org/Style/CSS/Overview.en.html>.
- [91] *What is JavaScript?* URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/What_is_JavaScript.
- [92] *Postman*. URL: <https://www.postman.com/>.
- [93] *Hyperledger Caliper*. URL: <https://www.lfdecentralizedtrust.org/projects/caliper>.
- [94] *Chart.js*. URL: <https://www.chartjs.org/>.