

# A Healthcare Digital Twin System Based on Blockchain Technology

by

Sadman Sakib Akash  
20166025

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

Department of Computer Science and Engineering  
BRAC University  
April 2023

© 2023. BRAC University  
All rights reserved.

# Declaration

It is hereby declared that

1. The thesis submitted is my own original work while completing M.Sc. degree at BRAC University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. A part of this thesis has been published as an article titled “A Blockchain Based System for Healthcare Digital Twin”, in IEEE Access Journal having DOI: 10.1109/ACCESS.2022.3173617.
4. We have acknowledged all main sources of help.

**Student’s Full Name & Signature:**



---

Sadman Sakib Akash  
20166025

# Approval

The thesis titled, “A Healthcare Digital Twin System Based on Blockchain Technology”, submitted by

Sadman Sakib Akash (20166025)

Of Spring, 2023, the thesis has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Sc. in Computer Science and Engineering on April, 2023.

## Examining Committee:

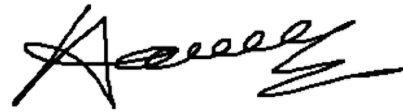
Supervisor:  
(Member)



---

Dr. Md Sadek Ferdous  
Associate Professor  
Department of Computer Science and Engineering  
BRAC University

Program Coordinator:  
(Member)



---

Dr. Amitabha Chakrabarty  
Associate Professor  
Department of Computer Science and Engineering  
BRAC University

Head of Department:  
(Chair)

---

Dr. Sadia Hamid Kazi  
Chairperson and Associate Professor  
Department of Computer Science and Engineering  
BRAC University

**Examining Committee: (cont.)**

External:  
(Member)



---

Dr. Sarker Tanveer Ahmed Rumeed  
Associate Professor  
Department of Computer Science and Engineering  
University of Dhaka

Internal:  
(Member)



---

Dr. Muhammad Iqbal Hossain  
Associate Professor  
Department of Computer Science and Engineering  
BRAC University

Internal:  
(Member)



---

Dr. S M Taiabul Haque  
Associate Professor  
Department of Computer Science and Engineering  
BRAC University

# Abstract

Digital Twin (DT) is a technology that replicates any physical phenomenon from physical space to digital space in congruous with the physical form's state. It does not confine to only spatial objects, any non-spatial scenarios can also be depicted with proper perception of the states. Though, DT technology was proposed with the incentive of revamping the intricate product lifecycle management in manufacture sector, other sectors like aviation, real states, healthcare, etc., have embraced it. By integrating DT in the healthcare sector, portrayal of patients in the digital space makes chances to create digital models, providing proper diagnosis, and evaluation facilities for digital healthcare services and Smart-health. However, determining a healthcare DT model for patient care and clinical purposes is seen as a challenging and imponderable task because of the lack of adequate data collection structures. Moreover, there are a number of problems in healthcare DT such as fragmented data and communication disorder which are making efforts futile. Also, the concept of healthcare DT is not formally defined and there lacks a consistent system architecture and data model, using which the diverse data flow of a Healthcare DT can be perceived structurally and can be used for later purposes. The collected structured data and careful simulation with proper analysis can render propitious opportunities in grievous health situations. For this reason, formulating a comprehensive healthcare data model for Healthcare DT is a prominent and preemptory task. On the other hand, there are also security and privacy issues as healthcare data can be used in malicious ways. For these reasons, to acquire the codified and finesse data with total integrity and proper access control, blockchain can be incorporated with the DT technology. In this thesis, we present a mathematical concept data model to accumulate the patient relevant data in a structured and predefined way with proper delineation. Additionally, the provided data model is described in harmony with real life contexts. Then, we have used the patient centric mathematical data model to formally define the semantic and scope of Healthcare Digital Twin system based on Blockchain. Accordingly, the proposed system is described with all the key components as well as proper protocol flow and evaluation. A short implementation of the proposed system has been conducted using *Hyperledger Fabric* and *BigchainDB* blockchains.

**Keywords:** Digital Twin (DT); Healthcare; Mathematical Model; Blockchain; Private Blockchain; Hyperledger Fabric; BigchainDB

## **Acknowledgement**

All praise to the Great Allah, for whom my thesis has been completed without any major interruption.

And to my parents, without their constant support it may not be possible. With their kind support and prayer, I am now on the verge of my graduation.

# Table of Contents

<b>Declaration</b>	<b>i</b>
<b>Approval</b>	<b>ii</b>
<b>Abstract</b>	<b>iv</b>
<b>Acknowledgment</b>	<b>v</b>
<b>Table of Contents</b>	<b>vi</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>ix</b>
<b>Nomenclature</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Research Objectives . . . . .	3
1.3 Structure . . . . .	4
<b>2 Literature Review</b>	<b>5</b>
2.1 Digital Twin (DT) . . . . .	5
2.1.1 Digital Twin for Developing a Product: . . . . .	6
2.1.2 Digital Twin for an individual instance: . . . . .	6
2.2 Healthcare Digital Twin . . . . .	6
2.3 Blockchain . . . . .	7
2.3.1 Public Blockchain: . . . . .	8
2.3.2 Private Blockchain: . . . . .	9
2.4 Smart Contracts . . . . .	9
2.5 Hyperledger Fabric . . . . .	10
2.5.1 Nodes . . . . .	10
2.5.2 Ledger: . . . . .	12
2.5.3 Channel . . . . .	12
2.5.4 Chaincode . . . . .	13
2.6 BigchainDB . . . . .	13
2.6.1 CREATE transaction . . . . .	13
2.6.2 TRANSFER transaction . . . . .	13
<b>3 Related Work</b>	<b>14</b>

<b>4</b>	<b>Proposal</b>	<b>18</b>
4.1	Methodology . . . . .	18
4.2	Mathematical Data Model . . . . .	19
4.2.1	Pre-Hospital Admit Data . . . . .	21
4.2.2	Patient Disease Diagnose Data . . . . .	23
4.2.3	Surgical Operative Procedure Data . . . . .	25
4.3	Threat Modeling . . . . .	28
4.4	Requirement Analysis . . . . .	29
4.4.1	Functional Requirements (FR) . . . . .	29
4.4.2	Security Requirements (SR) . . . . .	30
4.4.3	Privacy Requirements (PR) . . . . .	30
<b>5</b>	<b>System Architecture</b>	<b>31</b>
5.1	Architecture . . . . .	31
5.2	System Components . . . . .	32
5.2.1	Hospitals . . . . .	32
5.2.2	DApp . . . . .	33
5.2.3	System, $S$ . . . . .	33
5.2.4	Blockchain Platform . . . . .	34
<b>6</b>	<b>Implementation</b>	<b>35</b>
6.1	Development . . . . .	35
6.2	Protocol Flow . . . . .	36
6.2.1	Data Model . . . . .	37
6.2.2	Algorithm . . . . .	39
6.2.3	Protocol flow . . . . .	40
<b>7</b>	<b>Discussion</b>	<b>55</b>
7.1	Analysing Requirements . . . . .	55
7.1.1	Functional Requirements . . . . .	55
7.1.2	Security Requirements . . . . .	55
7.1.3	Privacy Requirements . . . . .	56
7.2	Fulfilment of Research Objectives . . . . .	56
7.3	Comparison . . . . .	57
7.4	Synergy with HIPAA and GDPR . . . . .	58
7.4.1	Similarities between <i>HIPAA</i> and <i>HDT</i> : . . . . .	60
7.4.2	Dissimilarities between <i>HIPAA</i> and <i>HDT</i> : . . . . .	60
7.4.3	Similarities between <i>GDPR</i> and <i>HDT</i> : . . . . .	61
7.4.4	Dissimilarities between <i>GDPR</i> and <i>HDT</i> : . . . . .	61
7.5	Advantages and Disadvantages . . . . .	62
7.6	Limitations . . . . .	63
7.7	Future Work . . . . .	63
<b>8</b>	<b>Conclusions</b>	<b>65</b>
	<b>Bibliography</b>	<b>76</b>



# List of Figures

2.1	Components of <i>Hyperledger Fabric</i> . . . . .	11
4.1	Methodology for the preparation of the system proposal. . . . .	19
4.2	Methodology for the proposed system. . . . .	20
4.3	Pre-Hospital Admit Data (PHA): The necessary data required before introducing a patient to the system. . . . .	22
4.4	Patient Disease Diagnose Data (PDD): The data accumulated while patient goes through disease assessment phase. . . . .	24
4.5	Surgical Operative Procedure (SOP): All the surgery pertinent data. . . . .	26
5.1	High-level Architecture and System Components . . . . .	32
6.1	The Implemented <i>Hyperledger Fabric</i> Network for <i>HDT</i> . . . . .	36
6.2	The User Interface for Registration Request . . . . .	42
6.3	Registration Flow. . . . .	43
6.4	The User Interface for Login Request . . . . .	44
6.5	Write Data Flow. . . . .	45
6.6	The User Interface for Write Data Request . . . . .	46
6.7	A <i>BigchainDB</i> Transaction for storing a file's encrypted raw data. . . . .	47
6.8	A <i>Hyperledger Fabric</i> Transaction for Write Data Request. . . . .	47
6.9	The User Interface for Health State Query Request . . . . .	48
6.10	HealthState Query Flow under system <i>H</i> . . . . .	49
6.11	HealthState Query Flow under system <i>S</i> . . . . .	50
6.12	The Output of the Health State Query Request. . . . .	51
6.13	Delete Data Protocol. . . . .	51
6.14	Share Data Flow. . . . .	53
6.15	The User Interface for Share Data Request. . . . .	53
6.16	The User Interface for the action of Share Data Request. . . . .	53
6.17	A <i>Hyperledger Fabric</i> Transaction for Share Data Request. . . . .	54

# List of Tables

4.1	Notation & semantics for Pre-Hospital Admit data . . . . .	21
4.2	Notation & Semantics for Patient Disease Diagnose Data. . . . .	23
4.3	Notation & Semantics for Surgical Operative Procedure. . . . .	25
6.1	Cryptographic and other Notations. . . . .	37
6.2	Data Model. . . . .	38
6.3	Registration Protocol. . . . .	43
6.4	Write Data Protocol. . . . .	44
6.5	HealthState Query Protocol under system $H$ . . . . .	46
6.6	HealthState Query Protocol under system $S$ . . . . .	48
6.7	Delete Data Protocol. . . . .	49
6.8	Share Data Protocol. . . . .	52
7.1	Comparison among some recent digital twin research works. . . . .	59

# Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

*AES* Advanced Encryption Standard

*AI* Artificial Intelligence

*API* Application Programming Interface

*CA* Certificate Authority

*CGM* Continuous Glucose Monitoring

*CNN* Convolutional Neural Network

*CPS* Cyber-Physical Systems

*CPU* Central Processing Unit

*DLT* Distributed Ledger Technology

*DSS* Decision Support System

*DT* Digital Twin

*E* The set of all equipment

*ECG* Electrocardiogram

*EHRs* Electronic Health Records

*GDT* Geometric Dimensioning and Tolerancing

*GPU* Graphics processing unit

*H* All the Hospitals under the proposed system model

*HD* High Definition

*HDD* Hard Disk Drive

*HDT* Healthcare Digital Twin

*HFM* Health Care Facilities Management

*HP* Hospitals that are not using the proposed System

*IoT* Internet of Things

*IVHM* Integrated vehicle health management

*LIME* Local Interpretable Model-Agnostic Explanations

*LTS* Long Term Support

*ML* Machine Learning

*MSP* Membership Service Provider

*NPM* Node Package Manager

*P* All the patients under the proposed system model

*P2P* Peer-to-peer

*PC* Personal Computer

*PH* The set of all physicians and surgeons

*PLM* Product Lifecycle Management

*RAM* Random Access Memory

*RFID* Radio-Frequency Identification

*RSA* Rivest–Shamir–Adleman

*S* Systems designated to extract data outside from our proposed system model with the help of special *API*

*SG* The set of all completed surgeries

*SHA* Secure Hash Algorithm

*SN* The set of all sensors

*SSD* Solid-State Drive

# Chapter 1

## Introduction

After the revolutionary advancement of the new information technologies in the scopes of Cloud Computing, Internet of Things, Big Data, and Artificial Intelligence, the new era of Smart Things is prevalent. Consequently, with more opportunities at hand, interconnection, interaction, and smart control of system components are opening new paradigms in technological evolution. The novel concept of Digital Twin (DT) is one of the paramount products of this technological advancement [1]. The DT concept was first introduced by Michael Grieves in 2002 for Product Lifecycle Management (PLM) where the system's primary elements were real space, virtual space, and the connection for data flow from real space to virtual space [2]. The conceptual model was referred to as the Mirrored Spaces Model [3].

The stereotypical product management systems had a lot of drawbacks, for example: generating knowledge, forthcoming demands, data redundancy, depiction of product lifecycle, and so on. If a product's lifecycle can be understood from a manufacturer's point of view then with proper utility and protocol flow it can be deployed with a DT system and the degree of amelioration will be immense. PLM comprises of a myriad of convoluted processes and there are involvement of numerous factors for maximizing the revenue, in this case Cyber-Physical Systems (CPS), deployed at the manufacturing industries is a phenomenal step forward [4].

Now, with the availability of IoT devices and fast network coverage, the incorporation of DT is not infeasible anymore. Moreover, the ongoing developments and the increase of applications of DT in PLM, has made the concept more sophisticated and decisive. To date, the most commonly used definition of digital twin was proposed by Glaesegen and Stargel in 2012: "Digital twin means an integrated multiphysics, multiscale, probabilistic simulation of a complex product, which functions to mirror the life of its corresponding twin" [5]. DT is not only confined in the scope of PLM, but also other sectors, e.g., healthcare [6]–[8], aerospace [9], [10], energy [11], [12], automobile [13], [14], and so on, are also invested in it. Currently, there is a surge in improving the DT technology in the healthcare situation and a myriad of developments are ongoing in the healthcare sector with respect to Artificial Intelligence [15]–[17], Big data [18], [19], and in other spectrum.

## 1.1 Motivation

There are some significant developments in the healthcare sector pertinent to DT. Madubuike-Obinna *et al.* [20] have presented a DT system architecture for Health Care Facilities Management (HFM). The system will consist of six layers to adopt DT technology to operate efficaciously. The authors have proposed to convert the traditional Healthcare Facilities into a real time system which will consider a myriad of factors. Their prime objective is to reduce the complexity of health care management processes with the help of virtual prototyping of physical assets and phenomena. Patrone-Carlotta *et al.* [21] have provided a framework for DT in the healthcare sector with the incorporation of Decision Support System (DSS). With the support of Data Mining and converting the physical state into a simulation, the system will generate resource scheduling, can minimize waiting time for a service, and will lead to a point where the leaner use of resources will be assured. Barbiero *et al.* [22] have presented a framework which will render a holistic view over current and future physiological conditions. Their primary objective is to turn the current medicinal treatment and curative discipline process into a preventative and interdisciplinary science, aiming at providing personalized, systemic, and precise treatment plans to patients. Advanced Artificial Intelligence (AI) approaches and integrated mathematical modeling will be encompassed to generate solutions for each patient according to their unique traits. These are active research areas and a lot of state of the art technologies are being used in the healthcare sector for DT.

Nevertheless, the recent developments in DT for the healthcare sector have some shortcomings from the stance of data sharing, storage, and access control [23]. With an astute perception, it can be observed that most of the research works have been conducted without any due consideration of how the compact system will work coherently with all the segregated elements of the system [24]. DT system will have all the traditional systems' protocol flow in addition to its mandatory feature of perceiving the physical state real time to update the virtual state.

Moreover, without proper system architecture, collecting insurmountable health data will cause a disarray and it will further complicate the situation with the involvement of data transformation techniques [25]. Additionally, the mixture of virtual state with new received physical state and for convergence, sometimes real data needs to be added with organic data, is also another concern [26]. For extracting knowledge from the collected data, the raw data will go through several data filtration and transformation processes.

Furthermore, people usually are indifferent toward their security of health data which leads to integrity and confidentiality breaches [27]. Not only this but also for a healthcare system to operate nationwide or internationally, while covering these wide distributed healthcare sectors and associated stakeholders, a distributed and strong governance is preemptive. Moreover, most of the healthcare systems are under a government central authority which needs to be changed.

## 1.2 Research Objectives

To ameliorate the complications or drawbacks mentioned before, a codified framework or system architecture is needed which is missing in the recent research works. So, in this thesis, we have provided a system architecture for DT in the healthcare sector for patients. Additionally we have presented a defined data model according to which the DT will collect data for a patient. Collecting unnecessary data is redundant and will perpetuate the involvement of other data filtering or transformation processes. For this reason having a defined data model will help deciding in which way DT will perceive which healthcare data from which dimensions for patients. With the proposed data model a patient can be uniquely identified as well as the patient portfolio can be generated with clinical data.

To solve the issue of the wide healthcare sector, a distributed network can be implemented by enforcing a distributed storage facility without any central governing authority. The blockchain technology can be utilized here. Blockchain renders the services of collecting intricate and diverse data immutably with sharing mechanism and proper access control. To accumulate varieties of health data in a structured and distributive way a blockchain based healthcare DT system is proposed and implemented. In a blockchain based DT system in the healthcare sector for patients, blockchain provides proper security and integrity, on the other hand, DT provides proper data aggregation, analysis, prognosis, and representation services which are favorable for building a proper healthcare DT.

The main research objectives (**R.O.**) are presented below:

**R.O. 1** *To represent the patient data in a defined and structured way, a patient centric mathematical data model will be developed. For better comprehension, real life contexts of patients' different treatment phases will be provided in accordance with the mathematical data model.*

**R.O. 2** *To identify vulnerabilities and define countermeasures for mitigating plausible threats to the system, a proper threat modeling and requirement analysis will be done based on the system's components.*

**R.O. 3** *To integrate healthcare digital twin system with Blockchain technology, a proper architecture along with all the required components will be elaborated.*

**R.O. 4** *To build a blockchain integrated system, the dependent system has to have pre-emptively defined participants and smart contracts, for this reason a proper protocol flow for utilizing the blockchain based system will be developed. It will describe how the system can be used in the different scenarios and how different components communicate with each other.*

**R.O. 5** *To implement the proposed system model in accordance with the protocol flow and the data model, private blockchain platform and distributed database will be incorporated.*

**R.O. 6** *To illustrate how the system has fulfilled all the requirements, a detailed analysis of the proposed system will be conducted.*

## 1.3 Structure

Chapter 2 presents a brief background on Digital Twin, Healthcare Digital Twin, Blockchain, Smart Contracts, *Hyperledger Fabric*, and *BigchainDB*. Next, we critically review a number of relevant researches in Chapter 3. In Chapter 4, the mathematical data model is presented with some pragmatic examples along with the threat modeling and the requirement analysis. Later the system architecture and the system components are delineated in Chapter 5. Then, we provide the detailed implementation process in accordance with protocol flow in Chapter 6. After that, we discuss how the proposed system for Healthcare DT has helped to satisfy different requirements and research objectives, advantages and disadvantages, a comparison with *HIPAA* [28] and *GDPR* [29] as well as with some recent research works, limitations, and future work of the presented model in Chapter 7. Finally, Chapter 8 concludes our findings.



# Chapter 2

## Literature Review

In this chapter we present a brief background information on Digital Twin, Healthcare Digital Twin, Blockchain, *Hyperledger Fabric*, and *BigchainDB*.

### 2.1 Digital Twin (DT)

DT means digital replication of anything from physical space. The primary condition of being a digital twin is both the virtual and the physical properties need to be congruous [30]. There will be a link for the system to acquire data from physical space with the help of sensors. For that reason perceived data from physical space of any length is obligatory. The more domain values can be accumulated of any physical entity, the more refined the twin will be. DT stands for the portrayal of the anatomy of a digital asset which is the depiction of a physical phenomenon from a physical space [31]. This is a complicated system which keeps the balance between virtual space and physical space and can develop cognitive knowledge about the physical environment [32]. With the rapid growth of new generation of information technologies like RFID and IoT, collecting data from each step of a physical phenomenon has become very efficient and effortless [33].

The basic concept of DT is to perceive a lifecycle of an entity so that it can understand the present physical state by converting it into a virtually representable twin for projecting a future state or delving deep into the entity's physical conditions [34]. Now an entity could be a physical object or a phenomenon which is not tangible. For example: a department management system, a project for building a DT based factory or wind turbine, and so on. When a DT will be deployed for a physical entity it can stay through its full lifecycle like create, build, operate/support, and dispose, of the physical entity [35]. Moreover, the experience knowledge of the entity after disposal can be also used for other homogeneous entities. This can happen only when the DT has been used at the create phase of the physical entity. In this situation, from the pre-development phase to the disposal phase, DT can accumulate all the necessary data.

DT can be classified into 2 types depending on for which purpose it will be used [2]:

### **2.1.1 Digital Twin for Developing a Product:**

This type of DT represents a digital prototype of a physical product which has yet to be developed. Usually this type of DT is deployed during the create phase because the physical product does not exist. For this reason, DT will be affiliated with the full lifecycle of the particular product's development phases. In this way, DT will have all the necessary information needed to build the product. By using previous knowledge and predictive mechanism, DT can determine the workflow and the behavior of the product [36]. In this respect, by assessing the current state of the development, product assembly, problem in the process, and so on, DT can provide the decisive plan for the workflow. For example: a DT can be implemented while at the manufacturing phase of a hospital. In this case, product life-cycle management is applicable. Life-cycle management and Building Phase of PLM will be crucial factors for the manufacturing phase [37]. By feeding hospital design plans, forward process estimation, process development, product assembly, and other necessary data to DT, it can deduce how the current workflow is going and how the future plans need to be developed by generating cognitive knowledge [38]. At the end of the lifecycle of the product, the generated knowledge of the full process can be recorded as experience for it to be utilized in other projects for DT systems in future. For these reasons, this type of DT is very significant for the advancement of Digital Twin technology.

### **2.1.2 Digital Twin for an individual instance:**

This type of DT represents the virtual state of a physical product or a non-spatial phenomenon. It can constantly update the virtual state in congruous to the physical state with the help of IoT devices [39]. Usually, this type of DT describes a specific corresponding physical product that an individual DT always stays linked to. Not necessarily, but a set of use cases are required for this to operate soundly, e.g., a fully annotated 3D model with Geometric Dimensioning and Tolerancing (GD&T), the list of components that are affiliated with the instance, the periodical update of physical state, the result of empirical tests on the instance, and operational prime states [2]. Let us assume a scenario where an automobile has been built with its all important parts integrated with IoT devices. As a result, DT can perceive all the dynamic data constantly and can take all the measurements it needs to define the current state in the virtual space which is symmetrical to the physical state of the automobile. It is plausible with the extant technology termed Integrated Vehicle Health Management (IVHM) [40]. The maintenance time of different parts, the performance of the product, and other significant factors can be devised remotely. The collected data later can be used for deriving knowledge of the instance. Same way, this type of DT mechanism can be used for patients in the healthcare sector with the help of wearable IoT devices.

## **2.2 Healthcare Digital Twin**

From the perspective of building a hospital falls under manufacturing, subsequently a digital twin for building a product can be utilized here. However, for caring a patient in the healthcare sector, digital twin for an instance, should be used. There

is no certain definition or protocols to define what are the minimum requirements to call a system a healthcare Digital Twin. With the best estimation, patient centric healthcare digital twin corresponds to a system where the virtual patient in the virtual space is constantly regulated with the forthcoming data collected from the physical space with the help of different IoT devices.

For building a patient centric healthcare digital twin, a lot of data is needed to represent the virtual patient. Now collecting data for the patient needs to be modulated carefully. A patient's data can be acquired from a myriad of sources. Additionally, collecting data from only a few domains will not render a holistic picture of the patient who might face uncertain health mishaps. However, accumulating data from any aspect will perpetuate different costly data transformation processes [41]. For these reasons, defining the data sources and from which domains collecting data would be more efficient is a preemptive task for building a DT instance for a patient in the healthcare sector.

One other important factor that needs to be considered is that the human body is full of complexities and defining the causal effect is nuanced [42]. As a result, from the perspective of healthcare, the utilization of collected data varies from patient to patient. For this reason, from birth to date if all the necessary data of a patient can be acquired succinctly then DT can devise the prolific result and can provide prediction on the potential threats based on the real time perceived data [43]. Moreover, sometimes the causal factor of a disease or co-morbidity can play differently based on external influences like work environment, age, social activities, and so on.

Healthcare DT is useful in many scenarios. There may be situations where a patient needs to be observed remotely at a particular moment and a patient centric DT could be very useful for guiding instruction and plausible threats. Additionally, patient centric DT can be for the whole body or just for a specific organ or for a particular disease. It is also a concern that there is still no automated 3D model generator for digital twin visual representation that can automatically generate the physical structure from the perceived data [44].

## 2.3 Blockchain

A distributed ledger is the consensus of multiple copies of digital data which can be spread out geographically with proper autonomy [45]. The digital data can be replicated, shared, and synchronized throughout the affiliated P2P network with the means of consensus protocol. Blockchain is a distributed ledger with the attribute of linear progression [46]. Blockchain contains data in the form of blocks and the blocks are distributed to all peer nodes over the network. The distribution of blocks throughout the network is done in a defined way with the help of cryptography.

Blockchain is an ordered back linked list consisting of blocks where each block is connected with its previous block. The previous block, a field in the blocks' head, is considered as the parent block for the newly generated block [47]. For uniquely identifying each block, hash value of the block data is used [48]. The newly generated

block stores the previous block's hash value as the previous block, for this reason blockchain can only be traversed backward. By traversing this way, the blockchain can be iterated all the way to the first block which is called the Genesis block, conventionally generated during the deployment of the blockchain [49].

Because of the replication attribute of the blockchain over the network, the immutability and the non-reputability of the data can be easily achieved [50]. Moreover, the hash value of each block data retains the integrity of the stored data [51]. Consequently, the data stored in one node is impossible to alter and by replicating the data to other nodes it strengthens the security of the full network.

Blockchain is a self-govern system because of consensus protocol which remedies the complications of depending on a third party and a central authority [52]. Subsequently, it renders the advent of technological evaluation by modulating the system depending on the prospective and the semantic of the system requirements. With this, different user groups can work concurrently and separately.

Nevertheless, attaining the same state for each participant node is convoluted because of the distributed nature of it. Because of this blockchain has adopted a mechanism called consensus. The consensus mechanism converges the full distributed network depending on the agreements of the participant nodes [53]. The consensus algorithms ensure the consistency and keep the states of blockchain congruous over the network [54].

Blockchain can be generally categorized in 2 types:

### **2.3.1 Public Blockchain:**

A public blockchain allows anyone to participate without any restrictions. Anyone can actively participate in validating blocks by installing a blockchain node in their devices [55]. Every participant node can read and write the public ledger. As there is no restriction, it is also called permissionless blockchain [56]. Only with a network connection, users can get access to the network and can start sending and validating transaction requests [50].

It is completely decentralized in nature, for this reason no one has the monopoly over the system [57]. For the convergence of blocks over the network special consensus algorithms are used, e.g., Proof of Work [58], Proof of Stake [59], and so on. The internal mining process of this type of consensus algorithms validates the users, blocks, and other requirements for the successful completion of a transaction [60]. In the digital public ledger, the blocks are validated and chained together by the miners in exchange for some monetary incentives [61]. Each transaction of a block is investigated for its authenticity. Moreover, the process checks the sender and receiver which also helps rendering a secure network as a whole.

No third party validation is needed which makes the system self-governed and autonomous [62]. Moreover, a public blockchain offers anonymity which provides the amenity to submit transaction requests anonymously. As a result, an adversary can-

not trace back the transaction to the requester. However, as participation is very easy, the number of participant nodes is comparatively high [63]. For this reason, the convergence takes more time than other types of blockchain so the throughput is limited and latency is very high. While, as the public ledger is distributed and shared with all the nodes, it renders higher security because of the higher distribution number [64].

Some well-known public blockchain systems are *Bitcoin* [65], *Litecoin* [66], *Ethereum* [67], and so on. There are also some notable public blockchain systems in the healthcare sector and they are *MedRec* [68], *FHIRChain* [69], and so on.

### 2.3.2 Private Blockchain:

A private blockchain is permissioned and only the permitted set of entities can take part in the system [64]. The mining process and consensus algorithms are maintained by only a handful of authorities or system admins. The new participants are also accepted by the admins as a result there is no need for anonymity in the private blockchain [70]. The read, write, and validation privileges are also dependent on the permissions which are pre-defined by the authorities. As the permission and data access is fixed, there is less computation compared to public blockchain [71].

There are circumstances where the privacy of data and restricted access control are primary, e.g., bank statements, healthcare data, and so on. In those cases, making this type of sensitive data public may cause a breach of data confidentiality [72]. Moreover, because of the less number of participants in the network for the block validation, the convergence of blocks over the network is very quick [73]. Only authorized entities can take part in the system which limits the adversarial activities.

All the decisions of a public blockchain are solved based on the majority of agreements which may deviate from the prime principle of the organization [74]. On the contrary, all the decisions of a private blockchain are fixed and controlled by the authorities which is favorable for the system's intention. Because of these reasons, we envisioned our proposed Healthcare DT to utilize a private blockchain platform.

Some notable private blockchain platforms are *Hyperledger Fabric* [75], *Hyperledger Sawtooth* [76], *Corda* [77], and so on. There are also some private blockchain based healthcare systems, e.g., *HealthChain* [78], *ModelChain* [79], *Ancile* [80], *MeDShare* [81], and so on.

## 2.4 Smart Contracts

Smart contracts are transaction protocols that can be executed on distributed networks among distrusted entities. It converts contractual conditions into computer readable and executable codes, and can run autonomously without any trusted authority [82]. It can change policies of digital assets by accessing ledger and can also write data on blockchain according to its definition [83]. A contract can encode a set of rules which are represented in the programming language and the correct execution of it is controlled by consensus protocol. When a transaction is addressed

with a smart contract, it executes with its defined manner without any enforcement. It is deterministic and can run concurrently so the same input at the same time in different places will always produce the same outputs. Also it renders proper distribution and automated workflow which is very apropos for blockchain which runs highly computational cryptographic algorithms [84]. Smart contracts can be a set of contracts where each contract can be assigned with different access permission so that the use of smart contracts can be restricted based on user type [85]. Smart contracts run in the scope of blockchain network for which it needs to be defined in a tidy manner as once deployed it cannot be edited except for redeploying it again which is time costly [86].

## 2.5 Hyperledger Fabric

A lot of unique features and easy adaptable deployment processes make *Hyperledger Fabric* one of the most popular permissioned blockchain platforms of the time. *Hyperledger Fabric* was developed by *Hyperledger* Foundation, an open source and global collaborative project on *Linux Foundation* [87]. The *Hyperledger* Foundation provides a mechanism where other developers can work in collaboration to develop more reliable, scalable, and efficient private blockchain systems. Many industry partners are stakeholders in the advancement of the *Hyperledger* initiative [88]. However, among all the other *Hyperledger* projects, the *Hyperledger Fabric* is the most prominent and mature one.

*Hyperledger Fabric* has a modular and configurable architecture, which facilitates optimizations of necessary tools based on the use cases. Consequently, it enables a fast and cost efficient system with great innovation and versatility [89]. One important aspect of *Hyperledger Fabric* is that it supports general-purpose programming languages like *Java*, *Go*, and *Node.js* for constructing Smart Contract [90]. Traditional blockchain frameworks confine the programming language for developing smart contracts compared to that *Hyperledger Fabric* gives more flexibility at this as no additional training is needed to get involved.

*Hyperledger Fabric* is an open source enterprise-grade permissioned distributed ledger technology platform which means the users or participants are not anonymous as a result fully trusted [91]. As mentioned earlier, *Hyperledger Fabric* is modular which supports pluggable consensus protocol [92]. For this reason, based on the use case if there is involvement of only one organization which means the network is operated by fully trusted authorities and the requirement of using byzantine fault tolerant is diminished. Consequently, the throughput and performance of the system increase.

There are a few key concepts in *Hyperledger Fabric*. In the following, we briefly present the functionalities of its different components. To articulate, the Figure 2.1 presents the connention of components of a *Hyperledger Fabric* network.

### 2.5.1 Nodes

There are a few types of nodes in *Hyperledger Fabric*. They have been described below:

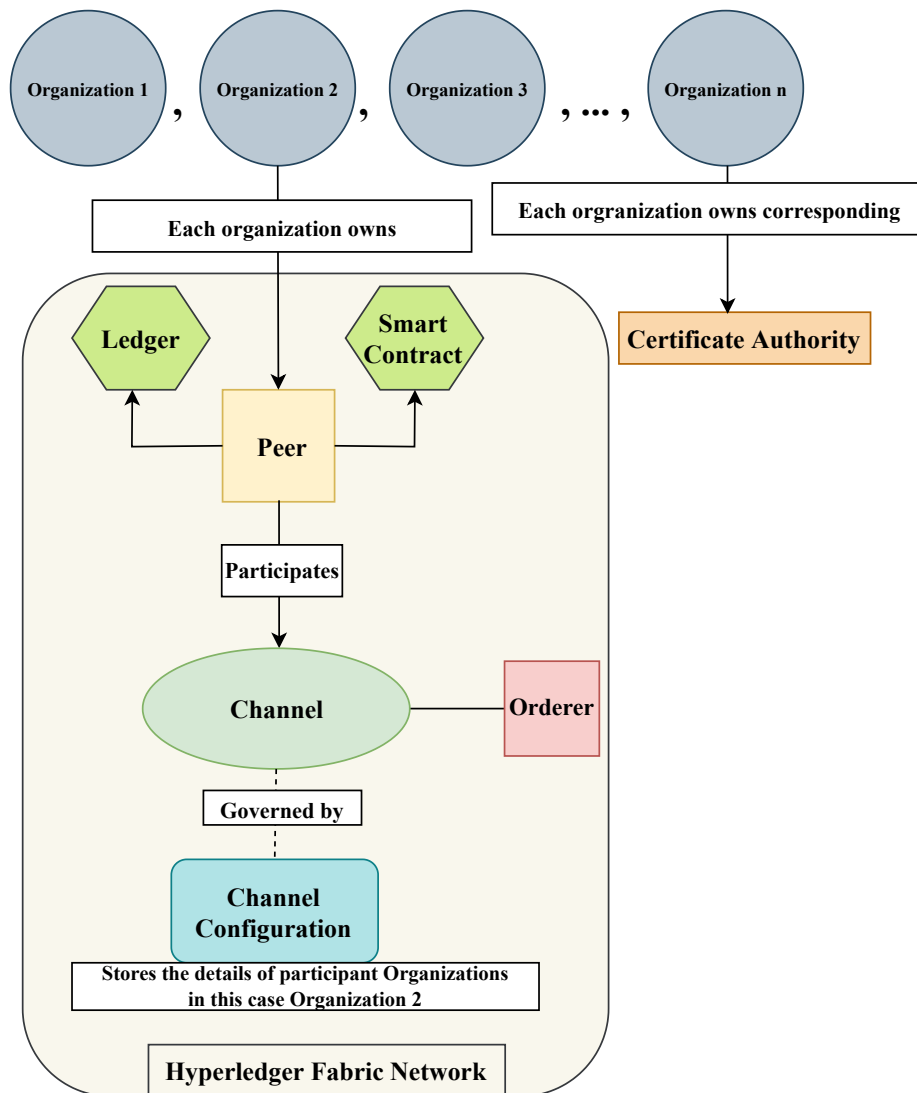


Figure 2.1: Components of *Hyperledger Fabric*.

### Organization:

To participate in a blockchain network of *Hyperledger Fabric*, an entity needs to be a member of an organization. Let us say, for a simple exchange business, the buyers will be from one organization, the seller will be from one organization, and the system admins will be from another organization. The organization defines a user group so that the network can easily identify the participants during ledger invocation [93]. From Figure 2.1, it can be perceived that the details of participant organization in a channel are stored in channel configuration.

### Certificate Authority (CA):

Whenever a new entity is registered to the blockchain network, the CA provides a corresponding identity consisting of private key, public key, and a digital certificate by which the entity can be uniquely identified in the *Hyperledger Fabric* network [94]. Each organization has its own CA which has been depicted in Figure 2.1. The public key of the entity is shared to other component in the network. So when the entity uses its private key for endorsing or signing a transaction, there needs to be



a mechanism to validate it. This is where the Membership Service Provider (MSP) comes in. The MSP has the public key of the entity (all of the entities' public keys) and can validate the authenticity of the signature or endorsement and can uniquely identify the entity and its certificate [95]. As a result, the permission and access control of each entity will prevail all over the network.

### **Peer:**

The blockchain network consists of peer nodes. Each organization needs to have at least one peer as the peers hold the copies of ledgers and smart contracts [96]. As the ledger and smart contract are necessary to hold and share data in the network which makes peers the most important component. Figure 2.1 illustrates how peer is connected to a channel.

### **Endorser:**

The endorsing peer is responsible for validation of each transaction by providing a digitally signed transaction response [97]. However, the endorsing peer needs to have smart contracts installed for conducting the validation process. Depending on the requirement, the transactions need to be validated by a set of endorsing peers and an endorsement policy identifies the organization whose peers need to digitally sign the transaction before committing it in the peer's copy of the ledger [98]. This full process is checked by an endorser according to the defined policy.

### **Orderer:**

The orderers are responsible for collecting all the transactions and endorsements from the network for combining them in a block [99]. After that, they send the blocks to all the peer nodes in the network for validating each block so that each corresponding ledger can be updated correctly with the same order and state [100]. A comprehensive understanding of the orderer node can be achieved from Figure 2.1. A system can have single or multiple orderers. The policy for the attributes of orderer are defined by the authorities or system admins.

## **2.5.2 Ledger:**

The ledger represents the blockchain and is a sequential and immutable record of all transactions in the form of blocks [101]. Whenever a new block is generated, each network component updates their own copy of ledger by which the replication of data prevails all over the network. Figure 2.1 presents how a ledger is connected to a channel with the means of a peer node.

## **2.5.3 Channel**

A channel creates a subdomain in the network to facilitate a set of organizations to communicate in private [102]. There can be multiple channels in a network and an organization can be part of more than one channel. The channel associates have their own sovereignty of defining the configuration and rules for the important factors like channel participants, data availability, endorsement policies, and so on



[103]. Channels are very efficient in isolating a set of organizations for private communication and private data which are accessible to the channel members. Let us assume a scenario where *Hyperledger Fabric* integrated system is deployed for 4 banks for financial transactions. However, 2 of the banks want to conduct some exchange where other 2 banks are not relevant. In this state, creating a channel with these 2 banks will be convenient and cost efficient as other banks do not have to participate in validating transactions for updating their own copy of ledger. There is also another type of channel called System Channel for regulating orderer services however that is not in the scope of our research work. A good visual understanding of channel can be achieved from Figure 2.1.

#### 2.5.4 Chaincode

In *Hyperledger Fabric*, the smart contract is known as Chaincode. A set of smart contracts can be Docker containerized and then will be installed on a peer as a Chaincode [104]. To perform activities on the ledger the chaincode needs to be installed. Chaincode outputs in a set of key-value pairs that can be submitted in the network and applied to the ledgers on all peers [105]. Figure 2.1 presents how a ledger is connected to a channel with the means of a peer node.

## 2.6 BigchainDB

*BigchainDB* is well known for having the characteristics of both traditional databases and blockchain [106]. Because of its dependency on *MongoDB*, the transaction size has some limitations [107]. Any type of data can be stored in *BigchainDB*, however it is very efficient to store a particular data that does not need alteration. It is not highly advisable for incorporating large organizations because it has only two types of transaction requests: “CREATE” and “TRANSFER” [108]. The Transfer transaction is out of scope for our research purpose.

### 2.6.1 CREATE transaction

The Create transaction enables generating a digital asset with additional metadata section [109]. The digital asset can have one or many owners. The asset can be anything so in our case, we can store the encrypted raw data as an asset creation transaction in *BigchainDB*. Each asset will be considered as a transaction and will be provided with a unique transaction ID. There are a myriad of built-in functions for formulating and querying stored data. Primarily, the function “getTransaction” will be used for accessing the asset or stored data. The mentioned function facilitates extracting a particular stored data against a transaction ID.

### 2.6.2 TRANSFER transaction

The transfer transaction is only applicable on another “CREATE” and “TRANSFER” transaction. As, the prime motivation of *BigchainDB* is to create asset and transfer the asset. By invoking a transfer transaction, the asset can be passed down to other owner or divided among many owner [110].

# Chapter 3

## Related Work

The developments of the digital twin have drawn research interest because of its volatile nature and successful practical applications. Here, we have given a summary of a few recent and noteworthy studies.

Peng et al. [111], in their article have presented a construction case on hospital DT in China, which had already been built. The authors have explained how the Continuous Lifecycle Integration method was used to construct the hospital twin. The real time acquired data is being used constantly to replicate the physical hospital in virtual space. Numerous sensors were installed during the ongoing construction to collect real-time data from the healthcare facility, and a digital twin allows the amenity to remotely control the entire system. However, the procedures for access control and encryption for the acquired data are not disclosed. There is also nothing mentioned about how this exorbitant amount of data will be utilized to generate rational decisions.

Liu et al. [112] have proposed a cloud based framework with healthcare DT. The idea was started since many elderly individuals don't seek medical attention because they don't care about illnesses. There are some people, on those conventional medicines that do not work, for them constant supervision is cardinal however it is not feasible for someone to look after another person for good. In these conditions, digital twin can customize medicines for elderly patients. Physical object, virtual object, cloud healthcare service platform, and healthcare data are the main components of the system. Although certain significant elements have been discussed relevant to system users and system components, no algorithm for predicting measurements has been mentioned.

With the help of edge computing, in this article [113], the authors have developed a healthcare Twin to alleviate heart diseases. Real-time data will be gathered through IoT devices with the means of smartphones, and after undergoing data fusion transformation techniques, the resulting data will be stored in a central database. They will be using edge computation so that the patient can be served real time with the help of their nearest device such as a smartphone. Usually the central data point systems detect and store the states of patients. There are a plethora of ECG collecting devices which can be connected via Bluetooth and real time patient data can be accumulated. Cardio Twin collects data through sensors (body area network),

medical records, social networks, and external sensors. Cardio twin will have access to the phone's Bluetooth communication to collect the sensor and the social network data. All the data will be fed to ML algorithms to take necessary actions. Cardio Twin is organized in three structures: Data source, AI-Inference Engine, and Multi-model Interaction and Smart Service. Their main incentive is to train Convolutional Neural Network (CNN), though any proper mentioning of data storage and security concerns are missing.

A similar type of work has been presented by Shamanna et al. [114], introducing Precision Nutrition to DT. The article is about Twin Precision Nutrition which monitors a group of 64 year old type 2 diabetic patients to reduce HbA1c in blood. Patient age, gender, duration of diabetes, and body mass index were recorded at enrollment. The platform collects data from body sensors and a mobile app to track and analyze the body's health signals in order to personalize the patient's treatment. Patients were asked to wear a sensor watch to continuously record sleep parameters, heart rate, step count, and other fitness parameters. Patients were also asked to record their blood pressure daily using a digital Bluetooth-enabled blood pressure meter. Patients measured their weight each morning along with blood beta hydroxybutyrate levels by means of finger prick. To create ambulatory glucose profiles CGM was performed daily throughout the study using a Libre Pro CGM Diabetes Sensor. All these data were transmitted securely through cellular networks via the mobile app. Patients were also asked to record their food intake. On a daily basis the patients were instructed what to eat by analyzing the received data respectively. Different factors were accounted for in devising decisions. Although the system is devising results based on real time data, the authors have not provided any mechanism by which they have conducted the analysis.

Barbiero et al. [22], in their article have proposed an architecture combining the qualities of a generative model with a graph-based representation of patho-physiological conditions. Using synthetic data with augmented explorable states of the underlying biological system, their proposed model can simulate intricate clinical situations which would have been hard to analyze otherwise. They have used numerous data models to collect data in a structural way. Moreover, they have utilized graph neural networks for deep learning and produced predictions about the evolution of the physiological state of the patient. They worked on different tissues of cardiovascular functions. There is not much about digital twin without some pertinent stuff. They have used graph neural networks for deep learning and produced predictions about the evolution of the physiological state of the patient.

In [115], Petrova et al. have proposed a DT platform for exploring the behavioral changes in patients with proven cognitive disorders with a focus on multiple sclerosis. This article proposes a platform for the exploration of behavioral changes in patients with proven cognitive disorders with a focus on Multiple Sclerosis. The platform has 2 main components: one provides functionality for diagnostics and rehabilitation and collects data for the next stages and another component, advanced analytical application provides services for data aggregation, enrichment, analysis and visualization that will be used to produce a new knowledge and support decision. According to the authors, the patients' data will be gathered from Electronic

Health Records (EHRs), open clinical databases, information from social networks, and other external applications. They have not elaborated any solutions to prevent data integrity and confidentiality violations.

A risk diagnosis digital twin system has been proposed in [116], to enhance the decision makings for liver disease with explainable artificial intelligence. The article is about using decision support systems in order to enhance decision making pertinent to liver disease risk diagnosis with explainable AI in Digital Twin. For patients' uniqueness factor they will be counting medical conditions, response to drugs, therapy, ecosystem, and many more. The main dataset which was used is the Indian Liver Patient Dataset. They will take the Random Forest model, they developed and use a state-of-the-art Explainable AI library called LIME (Local Interpretable Model-Agnostic Explanations). Because normal AI algorithms give the decisions but not depict which factors are taken into account. As the healthcare sector is a delicate matter for this reason explainable AI is used. The authors have provided sufficient information about the algorithms but have not mentioned anything about storage facilities and security.

One of the most notable and recent works in Healthcare DT can be found in [6]. In this work, the authors have proposed and implemented a framework which is beneficial to digital healthcare and to improve healthcare operations. The article is about diagnosing heart problems and detecting heart disease by classifying ECG heart rhythms with Digital Twin. DT combines Artificial Intelligence, Data Analytics, IoT, Virtual and Augmented Reality paired with digital and physical objects. This integration allows real-time data analysis, status monitoring to mitigate problems before they even occur, risk management, cost reduction, and future opportunities prediction. The proposed framework has 3 phases: Processing and Prediction, Monitoring and Correction, and Comparison. In the first phase the data will be stored after going through cleansing as raw data. The data they have used was MIT-BIH Arrhythmia Database. It contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory. In other phases, they have talked about how the human metrics value needed for making decisions. Like other works discussed here, they have not taken any precaution to safeguard the stored data.

In article [117], the authors have provided a vision about how multi-agent systems can be integrated with DT in the healthcare domain. The article is about how multi-agent systems can be integrated with digital twin in the healthcare domain. Here multi-agent means a software agent which can give a response before taking any action. The whole digital twin system will be built upon software agents. From the digital twins' perspective, agents provide a blueprint for engineering intelligent systems embedding AI and Distributed AI techniques, featuring some level of autonomy on top of DT, so that digital twin features could be exploited, for example: personal assistant agents supporting medics in doing their work and cooperating. The root of this research work is based on the Mirror World concept, introduced by Gelernter. Anything in the physical form can have a digital twin, it does not need to have a structure, let's say a patient management process can have a digital twin. Just like this, they discussed how agent based digital twin can be used for trauma

management. Trauma management can have two phases: pre-hospital phase where pre-medics are provided at the accident spot and general information data for digital twin are recorded like ambulance no, accident place, and time. Software agents work as a personal assistant for the head physician by providing forthcoming analysis and mandatory information. But, the authors have not provided any empirical process or analysis for the proposal.

Given these advancements, it makes sense that the DT based healthcare systems are advancing at a significant degree. Unfortunately, the serious problem is how this learning and analysis will go forward if complex and sophisticated data cannot be acquired from the physical environment which has not been solved yet. Also how this large volume of information and insights can be securely and privately recorded, is a grievous issue. With these in mind, we will try to assuage the collection of ambiguous data by providing a conceptual data model mathematically in Section 4.2. The corresponding threat modeling and requirement analysis for the Healthcare Digital Twin system are discussed in Section 4.3 and Section 4.4 respectively. Moreover, we will solve the insecure storing problem by representing a full system architecture for blockchain based Healthcare Digital Twin in Chapter 5.

# Chapter 4

## Proposal

### 4.1 Methodology

Because of the extant nuances in the healthcare sector, defining the data sources is an important task before defining the system architecture. Arbitrarily collecting patient data will cause redundant use of data transformation and filtering processes in the system. For this reason, we start our system proposal with the mathematical data model which is presented in Chapter 4. With each passing time, the data model can be changed for acquiring the required data for a patient. The data model stages have been collected through researching numerous healthcare relevant articles. It is to be noted that DT can be represented in a simple or in a more detailed and complex way. However, in our thesis, we have defined the data model in a simpler way, mostly focusing within the scope of our thesis. Towards this aim, our prime objective is to define the patient clinical data and develop a patient centric system.

Additionally to tackle uncertain threats and for the system to be autonomous proper threat modeling and requirement analysis has been conducted in Chapter 4. Figure 4.1 illustrates the preparation for the system proposal.

A blockchain integrated system cannot be invoked with a traditional web application. Decentralized application will be used from the client side for exploiting blockchain services. The architecture of the proposed system will be elaborated along with the considered assumptions in Chapter 5.

The proposed system will be implemented for all the primary requests. The applied tools and other configuration processes will be elaborated. After that, the protocol flow of all primary requests will be presented with proper delineation in harmony with the implementation in Chapter 6.

In Chapter 7, we will discuss how our proposed system has successfully integrated all the requirements and will critically analyse our proposed system.

Figure 4.2 illustrates the methodology for the proposed system.

Next we will discuss the preparation for the proposal of the system architecture starting with the mathematical data model.

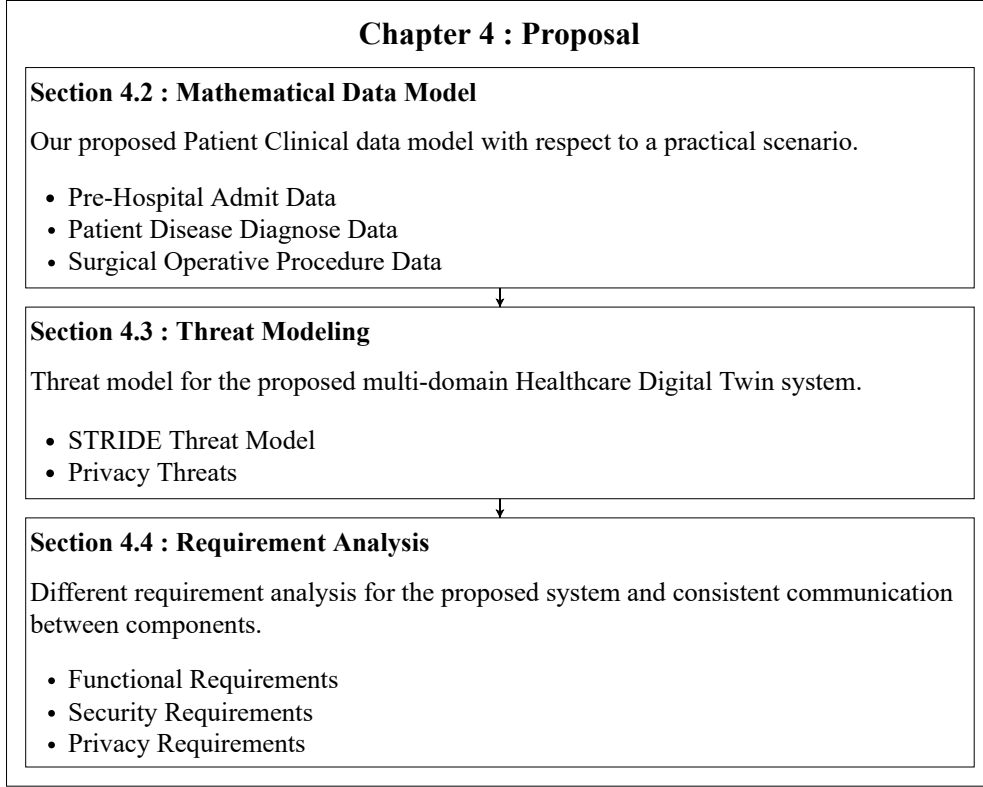


Figure 4.1: Methodology for the preparation of the system proposal.

## 4.2 Mathematical Data Model

Let us consider the whole Healthcare DT system as  $HDT$  and moreover  $H$  and  $P$  denote the set of all hospitals and patients respectively under  $HDT$ . The set of external hospitals not under the  $HDT$ , can be denoted as  $HP$ . To have an empirical understanding of the use case, let us consider a scenario where Brad is a patient who have been receiving healthcare from some hospitals,  $hp \in HP$ , outside of  $HDT$ . Consequently, there is no data available for Brad in  $HDT$  and he is now motivated to take healthcare services form a hospital which is under our proposed  $HDT$ .

So, he will go through the registration process or other pre-registration processes in a hospital,  $h \in H$  under  $HDT$ . His patient virtual profile will be stored with a unique ID or userName which can be worked as an identifier and subsequently all his patient relevant previous data will be acquired from  $hp$  with the means of a system,  $s \in S$  under  $HDT$ .  $S$  provides the amenity to acquire data from outside the jurisdiction of  $HDT$  with the help of some special APIs. The elaborated analysis of System  $S$  has been given in Chapter 5 and  $HDT = \{H, S\}$ .

Human body works differently from person to person, so which factors affect the body in which way is very difficult to determine [118]. For this reason, from social status to health conditions, all data regarding Brad, will be accumulated to have a comprehensive understanding of disease and the surroundings. Social data will contain Brad's age, daily routine, workplace, and so on [119]. It can be possible that Brad's condition is already severe and he has been taking care-giving services from exxternal hospitals, so Brad's caregivers' information will be collected from  $hp$ .

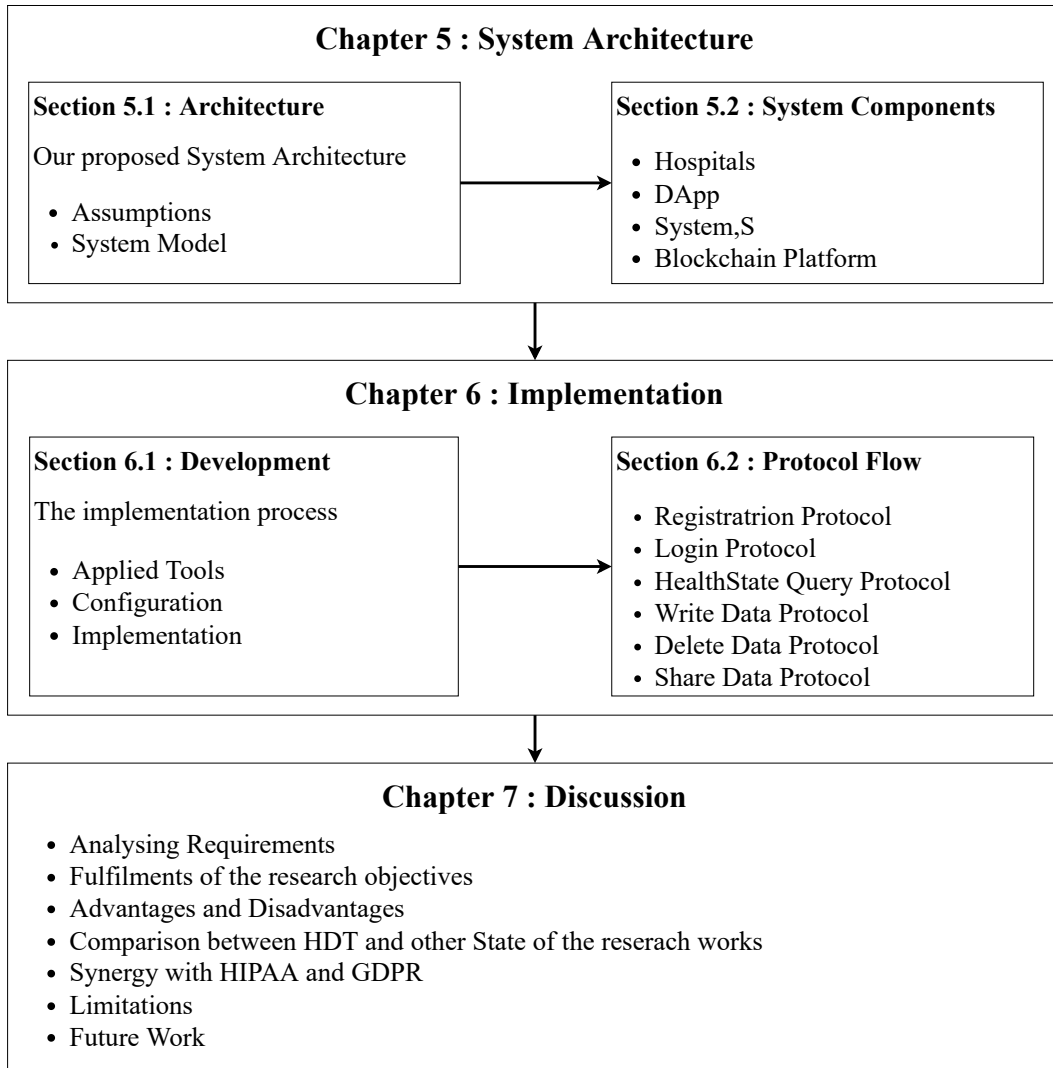


Figure 4.2: Methodology for the proposed system.

There will be other homogeneous data, e.g., previous tests, surgery data, medication, and so on, which will be accumulated succinctly [120].

There are some special types of data like food habit, regular activities, daily strenuous workout, and so on, which will be collected with the consent of Brad as self reported data, assuming Brad will provide the correct data on a daily basis [121].

With all this data, before admitting Brad into  $h$ , an aggregated Pre-Hospital Admit Data can be achieved which will enhance the chances of remedying the disease and will portray a better holistic representation of Brad as a patient. Most of the Pre-Hospital Admit Data can be collected with the help of  $s$  and the process is illustrated in Figure 4.3. All the necessary semantics and notations for Pre-Hospital Admit Data are elaborated in Table 4.1.



Notation	Meaning
PHA	Pre-Hospital Admit Data
SD	Social Data
VS	Vital Sign
MD	Medication
SR	Self Reported Data
PTD	Previous Test Data
PSG	Previous Surgery Data
CGD	Caregiver Data
CG	Caregiver
TS	Set of Timestamps
AA	Set of Attributes (Names)
AV	Set of Attribute Values

Table 4.1: Notation & semantics for Pre-Hospital Admit data

### 4.2.1 Pre-Hospital Admit Data

Our proposed healthcare DT system,  $HDT$  has an explicit intention of remedying the extant nuance and ambiguity for the care of patients. For this reason, the corresponding surrounded patient data will play the prominent part for a DT system in the healthcare sector. So, while introducing a patient,  $p \in P$ , like Brad who is new to the  $HDT$ , it would be common that there are a lot of external Pre-Hospital Admit Data of  $p$  that has to be collected through  $s \in S$  from other extant systems succinctly with proper pre-defined structure.

The subscript of any data representative set notation will represent the domain or system of that data. Then,  $p_h$  denotes the set of all patients in a hospital  $h \in H$ . The superscript will represent the entity. So,  $T_h^p$  will stand for a data set  $T$  which has been provided by an entity  $p$  under the system or domain  $h$ .

A special kind of set  $AA$  will subsume the attributes for the data set. The subscript and superscript of  $AA$  will represent the data set and the entity providing the data respectively. In the same manner, a special kind set  $AV$  will subsume the respective values for  $AA$ . So, Social Data of Pre-Hospital Admit Data for  $p$  under  $s$  with respect to time can be defined as following notation in Equation (4.1):

$$SD_s^p = \{ (AA_{SD}^p \times AV_{SD}^p) \times t \mid AA_{SD}^p \& AV_{SD}^p \text{ is defined} \wedge t \in TS \} \quad (4.1)$$

For convenience let us consider all the data was accumulated at the same time. As a patient, Brad's Social Data can be considered as following where  $SD_s^{Brad}$  represents all the Social Data for Brad under system  $s \in S$  according to Equation (4.1):

$$SD_s^{Brad} = \{ (Age,38,1625240415), (Sex,male,1625240415), (Home-Average-Temperature,33,1625240415), (Home-Address,Dhaka-Bangladesh,1625240415), (Job-Type,Student,1625240415), (Job-Environment,outdoor,1625240415) \}$$

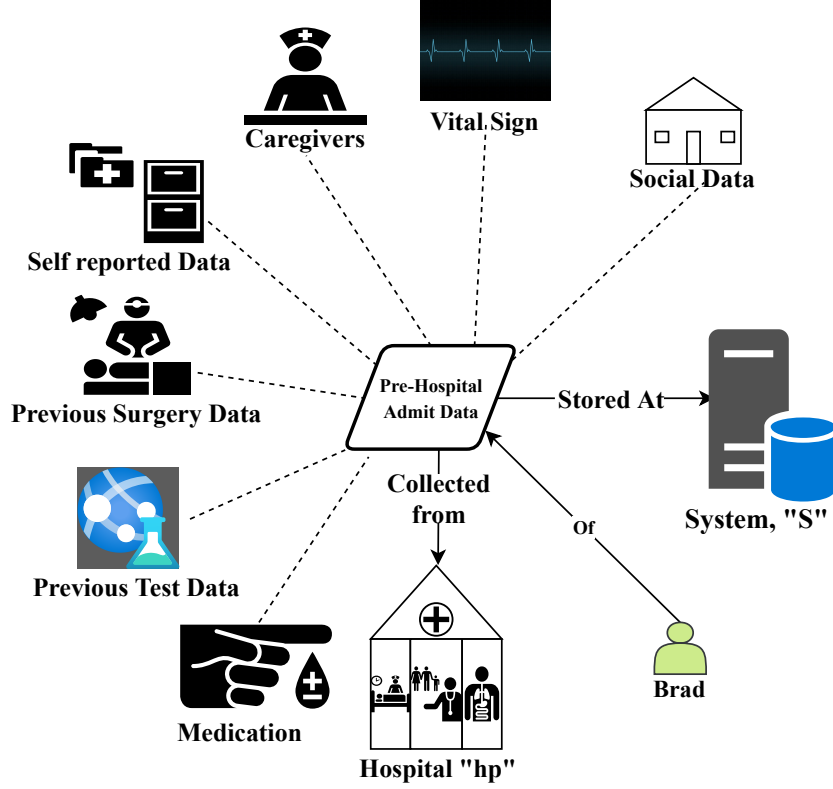


Figure 4.3: Pre-Hospital Admit Data (PHA): The necessary data required before introducing a patient to the system.

In the same manner,  $VS_s^p$ ,  $MD_s^p$ ,  $SR_s^p$ ,  $PSG_s^p$ , and  $PTD_s^p$  can be defined.

Let  $CG_S$  be the set of all caregivers in  $s$  and  $CGD_S$  be the set of data for caregivers in  $s$ . Like before,  $CGD_S$  can be represented as shown in Equation (4.2):

$$CGD_S = \{ (AA_{CGD}^{cg} \times AV_{CGD}^{cg}) \mid AA_{CGD}^{cg} \& AV_{CGD}^{cg} \text{ is defined} \wedge cg \in CG_S \} \quad (4.2)$$

**Definition 1** Let,  $patientToCaregiver: P \times S \rightarrow \mathcal{P}(CG_S \times CGD_S)$  be the function which returns the set of caregivers for a particular patient within system  $S$ .

For brevity, we denote such a set of caregivers with the notation  $CG_s^p$  for a patient  $p \in P$  in a system  $s \in S$ . That is,

$$patientToCaregiver(p, s) = CG_s^p$$

Let us say, the system  $s$  can extract the data from where Brad had been taking services and has all the Caregivers' data represented in the following way:

$$CG_S \times CGD_S = \{ \{ (Caregiver - Name, Bob), (Patient, Brad), (Duration, 213 \text{ days}), (Timestamp, 1625240415) \}, \{ (Caregiver-Name,Bob), (Patient,Brad), (Duration,21 \text{ days}), (Timestamp, 1625240415) \}, \{ (Caregiver-Name,Mark), (Patient,Selim), (Duration,80 \text{ days}), (Timestamp, 1625240415) \}, \dots \}$$

$$patientToCaregiver (Brad, s) = \{ \{ (Caregiver - Name, Bob), (Duration, 213 \text{ days}), (Timestamp, 1625240415) \}, \dots \}$$

So,  $patientToCaregiver (Brad, s) = CG_s^{Brad}$ . All the accessible previously stored data regarding Brad's caregivers from other systems can be amassed with the help of Definition 1.

At this point, all the external data before admitting into the hospital for patient  $p$  can be accumulated as Pre-Hospital Admit Data and can be defined according to Equation (4.3):

$$PHA_s^p = \{ SD_s^p \cup VS_s^p \cup MD_s^p \cup SR_s^p \cup PSG_s^p \cup PTD_s^p \cup CG_s^p \} \quad (4.3)$$

## 4.2.2 Patient Disease Diagnose Data

Now, Brad has become a patient of the hospital  $h \in H$  under  $HDT$  with all the necessary data. Before starting new treatment, Brad's disease needs to be determined. So, to determine Brad's disease, a specialist physician will be assigned [122]. The physician will provide a Check-Up Prescription containing numerous tests and current health condition. After getting the Check-Up Prescription, Brad will go through the tests. The tests data will be recorded as Diagnose Test Data [123]. There may need of some continuous monitoring of vital sign, so the necessary sensor data will also be collected as Patient Sensor Data. After having all this data, the physician can provide an accurate evaluation for Brad's disease and the whole data can be represented as Patient Disease Diagnose Data. At this point, all the Patient Disease Diagnose Data will be stored under domain hospital  $h$  and the process is illustrated in Figure 4.4. The necessary semantics and notations for Patient Disease Diagnose Data are described in Table 4.2.

Notation	Meaning
PDD	Patient Disease Diagnose Data
DTD	Diagnose Test Data
PS	Patient Sensor Data
CUP	Check-Up Prescription
TR	Test Result Data
TS	Set of Timestamps
AA	Set of Attributes (Names)
AV	Set of Attribute Values

Table 4.2: Notation & Semantics for Patient Disease Diagnose Data.

Let,  $PH_h$  denote the set of all physicians and surgeons under hospital  $h$ . After admitting into a hospital  $h \in H$ , a patient  $p \in P$  will be examined by a physician  $ph \in PH_h$ . The  $ph$  provided Check-Up Prescription can be defined according to Equation (4.4):

$$CUP_h^p = \{ (AA_{CUP}^{ph} \times AV_{CUP}^{ph}) \times t \mid AA_{CUP}^{ph} \& AV_{CUP}^{ph} \text{ is defined} \wedge ph \in PH_h \wedge t \in TS \} \quad (4.4)$$

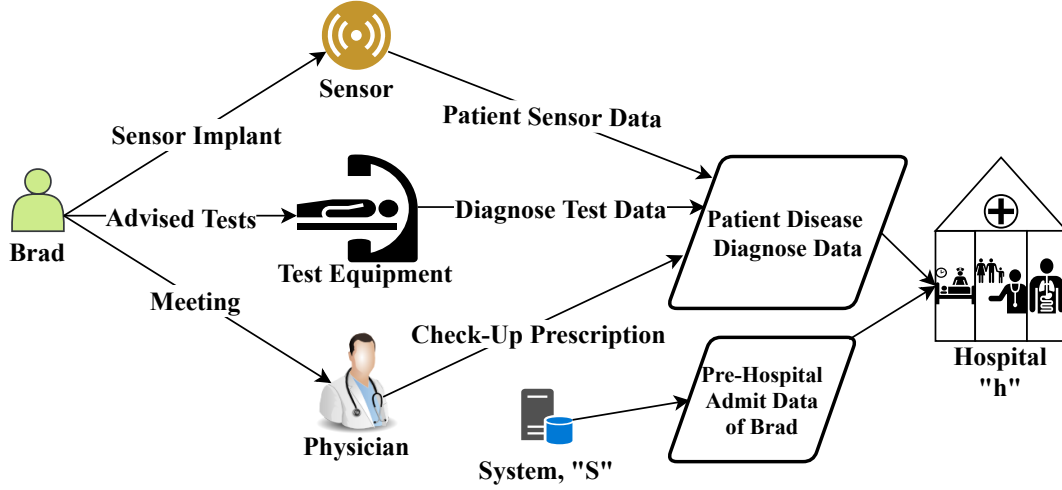


Figure 4.4: Patient Disease Diagnose Data (PDD): The data accumulated while patient goes through disease assessment phase.

Let  $E_h$  denote the set of all equipment under hospital  $h \in H$ . Different tests will be conducted to have a compendium knowledge of the disease, advised by  $ph \in PH_h$ . For conducting the tests, different equipment will be used. The Diagnose Test Data set for patient  $p \in P$  within the system  $h$  can be depicted according to Equation (4.5):

$$DTD_h^p = \{ (AA_{DTD}^e \times AV_{DTD}^e) \times ph \times t \mid AA_{DTD}^e \& AV_{DTD}^e \text{ is defined} \wedge t \in TS \wedge e \in E_h \} \quad (4.5)$$

All the results of the tests will be aggregated under hospital  $h$  as  $TR_h$  and an element of that can be represented as  $tr \in TR_h$ . Let us assume, Brad has conducted two advised tests so Diagnose Test Data for Brad can be represented as follows:

$$DTD_h^{Brad} = \{ \begin{aligned} & \{ (\text{Physician-Name, Selim}), (\text{Equipment-Type, diagnostic}), (\text{Test-Data, tr}), \\ & \quad (\text{Timestamp, 1625240415}) \}, \\ & \{ (\text{Physician-Name, Mark}), (\text{Equipment-Type, medical-laboratory}), (\text{Test-Data, tr}), \\ & \quad (\text{Timestamp, 1625240415}) \} \end{aligned} \}$$

Let us denote the set of all sensors  $SN_h$  under system  $h$ . The sensor data for  $p$ , perceived from various sensors can be depicted according to Equation (4.6):

$$PS_h^p = \{ (AA_{PS}^{sn} \times AV_{PS}^{sn}) \times t \mid AA_{PS}^{sn} \& AV_{PS}^{sn} \text{ is defined} \wedge t \in TS \wedge sn \in SN_h \} \quad (4.6)$$

Now, the Patient Disease Diagnose Data aggregation set can be defined according to Equation (4.7):

$$PDD_h^p = \{ CUP_h^p \cup DTD_h^p \cup PS_h^p \} \quad (4.7)$$

### 4.2.3 Surgical Operative Procedure Data

At this point, as a patient, Brad can be defined by the set of Pre-Hospital Admit and Patient Disease Diagnose Data. But, there may be a need for surgery for the betterment of Brad's health. If a surgery will be conducted under the jurisdiction of *HDT*, then this surgery data needs to be collected with proper structure according to the defined way. Before having the surgery, Brad will again go through some surgery pre-requirement tests known as Pre-Operative Assessment.

Depending on the Patient Disease Diagnose Data and Pre-Operative Assessment Data and with the consent of the designated physician's statement a surgery will be conducted by a specialist surgery team in a preferable hospital operating room [124]. There are various types of surgeries and depending on the type and special circumstances, a surgery will have some defined number of steps known as the Sequence of Surgery. For accepting the surgery data as a precedent for prognostication, these Sequences of Surgery data need to be collected in a sophisticated way by a surgery team member.

Some important or exceptional notable criteria will also be recorded by a surgeon called Surgeon Specific Factor [125]. After the surgery, Brad will be under complete observation for a specific time with the consent of the surgeon. During that time, the monitoring data will be collected as Post-Operative Follow-up data [126]. The necessary Notation and semantics for Surgical Operative Procedure are described in Table 4.3 and illustrated in Figure 4.5.

Notation	Meaning
SOP	Surgical Operative Procedure
PD	Patient Data
POA	Pre-Operative Assessment
POF	Post-Operative Follow-up
ST	Surgery Team
STD	Surgery Team Data
SOS	Sequence of Surgery
SSF	Surgeon Specific Factor
ORF	Operating Room Factor
TS	Set of Timestamps
AA	Set of Attributes (Names)
AV	Set of Attribute Values

Table 4.3: Notation & Semantics for Surgical Operative Procedure.

Surgeries will be conducted when the exigencies of the patients' conditions demand it. Before having a surgery,  $p \in P$  will go through Pre-Operative Assessments for the surgery specific pre-requirements according to  $PDD_h^p$ . If, the equipment used for the test are  $e \in E_h$ , then Pre-Operative Assessment can be defined according to

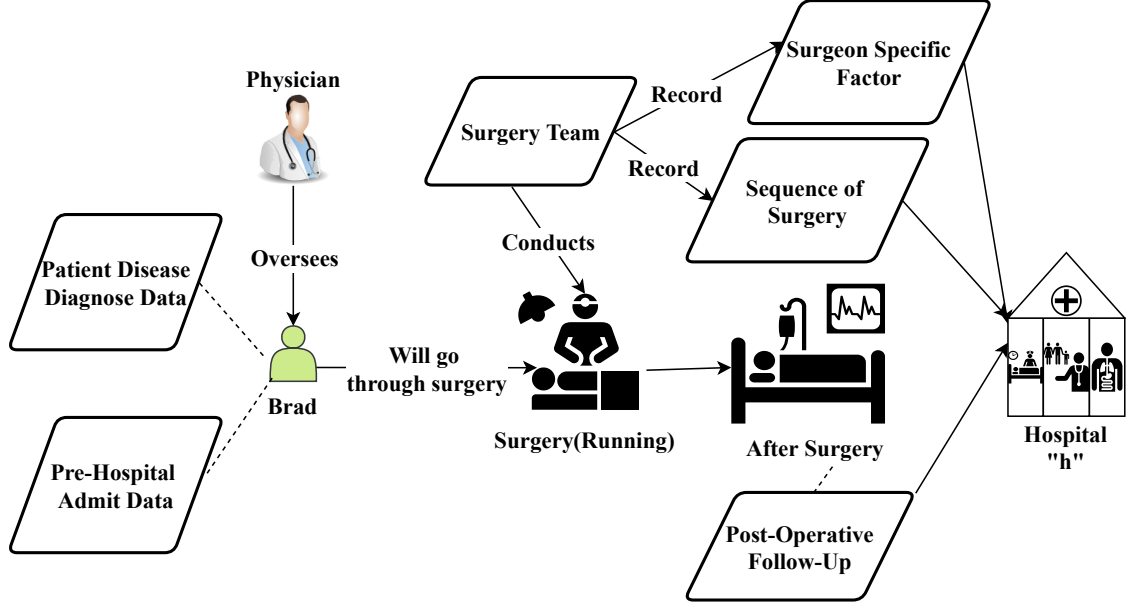


Figure 4.5: Surgical Operative Procedure (SOP): All the surgery pertinent data.

Equation (4.8):

$$POA_h^p = \{ (AA_{POA}^e \times AV_{POA}^e) \times t \mid AA_{POA}^e \& AV_{POA}^e \text{ is defined} \wedge t \in TS \wedge e \in E_h \} \quad (4.8)$$

Let us say, Brad needs a surgery. So, according to physician Selim's Check-Up Prescription,  $cp \in CUP_h^{Brad}$ , Brad will go through some pre-surgery tests and the results will fall under previously mentioned test result set,  $TR_h$  and  $tr \in TR_h$ . So, for Brad, Pre-Operative Assessment can be represented according to Equation (4.8):

$$POA_h^{Brad} = \{ (Advised-CheckUp, cp), (Presurgery-Test-Data, tr), (Surgery- Number, 1), (Timestamp, 1625240415) \}$$

There will be some surgery pertinent data, in the timeline from  $p$  entering into the operative room to leaving, will need to be recorded. Depending on the surgery type, a special surgery team will operate the surgery procedures. Let us consider,  $SG_h$  as the set of all surgeries in  $h$ .  $ST_h$  can be denoted as the set of all surgery teams and  $STD_h$  be the set of data for the surgery teams in  $h$ . Then,  $STD_h$  can be represented according to Equation (4.9):

$$STD_h = \{ (AA_{STD}^{stt} \times AV_{STD}^{stt}) \times SG_h \times t \mid AA_{STD}^{stt} \& AV_{STD}^{stt} \text{ is defined} \wedge stt \in ST_h \wedge t \in TS \} \quad (4.9)$$

**Definition 2** Let,  $patientToSurgeryTeam: P \times SG_h \rightarrow \mathcal{P}(ST_h \times STD_h)$  be the function which returns a set of Surgery Team for a particular patient within system  $H$  for a surgery.

For brevity, we denote such a set of Surgery Team for a surgery,  $sg \in SG_h$ , with the notation  $ST_h^p$  for a patient,  $p \in P$ , in a hospital,  $h \in H$ . That is,

$$patientToSurgeryTeam(p, sg) = ST_h^p$$

There is a sequence of surgery procedures depending on surgery type and the relevant data regarding different sequences will be recorded by a member of  $ST_h^p$  usually surgeon's assistant. The Sequence of Surgery can be defined according to Equation (4.10):

$$SOS_h^p = \{ (AA_{SOS}^{st} \times AV_{SOS}^{st}) \times sg \times t \mid AA_{SOS}^{st} \& AV_{SOS}^{st} \text{ is defined} \\ \wedge st \in ST_h^p \wedge sg \in SG_h \wedge t \in TS \} \quad (4.10)$$

Some special notable or exceptional attributes of the surgery will be recorded by surgeon and will be stored as Surgeon Specific Factor and can be defined according to Equation (4.11):

$$SSF_h^p = \{ (AA_{SSF}^{st} \times AV_{SSF}^{st}) \times sg \times t \mid AA_{SSF}^{st} \& AV_{SSF}^{st} \text{ is defined} \\ \wedge st \in ST_h^p \wedge sg \in SG_h \wedge t \in TS \} \quad (4.11)$$

A compendium knowledge about Operating Room Factor regarding surgery will be recorded. It can be represented according to Equation (4.12):

$$ORF_h^p = \{ (AA_{ORF}^{st} \times AV_{ORF}^{st}) \times sg \times t \mid AA_{ORF}^{st} \& AV_{ORF}^{st} \text{ is defined} \\ \wedge st \in ST_h^p \wedge sg \in SG_h \wedge t \in TS \} \quad (4.12)$$

After the surgery, patient  $p$  will be under complete observation in hospital  $h$ . The Post-Operative Follow-up data for  $p$  can be represented according to Equation (4.13):

$$POF_h^p = \{ ((AA_{POF}^{ph} \times AV_{POF}^{ph}) \cup (AA_{POF}^{sn} \times AV_{POF}^{sn}) \cup (AA_{POF}^e \times AV_{POF}^e)) \times t \mid AA_{POF}^{ph}, AV_{POF}^{ph}, AA_{POF}^{sn}, AV_{POF}^{sn}, AA_{POF}^e \& AV_{POF}^e \text{ is defined} \\ \wedge sn \in SN_h \wedge e \in E_h \wedge ph \in PH_h \wedge t \in TS \} \quad (4.13)$$

Based on Patient Disease Diagnose Data and Pre-Operative Assessment data ( $PDD_h^p \wedge POA_h^p$ ), a surgery  $sg \in SG_h$  will be conducted on a patient  $p$  by a surgery team ( $ST_h^p$ ) in a operating room ( $ORF_h^p$ ) and during the surgery  $sg$ , Sequence of Surgery and Surgeon Specific Factor ( $SOS_h^p \wedge SSF_h^p$ ) will be recorded and after completing the  $sg$  the  $p$  will be under complete observation ( $POF_h^p$ ) for a definite time. This whole process can be structurally defined as Surgical Operative Procedure and this action can be represented according to Equation (4.14):

$$SOP_h^p = ST_h^p \frac{(PDD_h^p \wedge POA_h^p) \prec ORF_h^p}{(SOS_h^p \wedge SSF_h^p) \models POF_h^p} \quad (4.14)$$

Now after having Brad's Pre-Hospital Admit Data, Patient Disease Diagnose Data, and Surgical Operative Procedure data, Brad as a Patient can be properly depicted. For a patient  $p \in P$  under the Healthcare DT system  $hdt \in HDT$ ,  $HDT$  is comprised of system  $H$  and  $S$ ,  $hdt = \{h, s\}$  where,  $h \in H$  and  $s \in S$ , a compendium representation of Patient Data can be defined by according to Equation (4.15):

$$PD_{hdt}^p = \{ PHA_s^p \cup PDD_h^p \cup SOP_h^p \} \quad (4.15)$$

## 4.3 Threat Modeling

Predictable desirable, predictable undesirable, unpredictable desirable, and unpredictable undesirable behaviors are the four emergent behaviors of a complex system [2]. To minimize the system's unpredictable behaviors, necessary measurements should be taken. Digital Twin is without a doubt a complex system. There will be numerous obstacles for various causes in the Digital Twin, where data will be acquired in real time from the physical space while simultaneously being analyzed, processed, and updated upon in the virtual space [127]. Threat modeling makes it easier to comprehend numerous problems and threats before deploying the system for empirical use in the real world. We have selected the well-known Microsoft threat model, STRIDE [128], which encompasses the various security threats which are listed below:

- **T1-Spoofing Identity:** The act of spoofing refers to an adversary using the identity of an authorized entity (e.g., as a patient or sensor) to illegally participate in activities. If the system generated certificate or unique identity gets stolen, this kind of adversary attack can happen and proper system architecture can defend it.
- **T2-Tampering with Data:** An attacker may attempt to alter effective decisions to debase patients' condition or hospital management processes (e.g., by increasing medicine dosage, an adversary can impinge on a patient's health condition). If a system has low security precautions where control access and active participants are not modulated periodically, this type of threat increases.
- **T3-Repudiation:** An attacker can repudiate after altering data. For this reason, digital signature of the corresponding request toward the system has to be attached from the user. Which can be validated later for defending the repudiation threat.
- **T4-Information Disclosure:** Restricted data can be disclosed or made public (e.g., a leak of medical data of an illustrious personal can bring significant ramifications in his/her health security). Encryption and proper access control of entities will minimize this threat.
- **T5-Denial of Service (DoS):** The system will be impeded to do the tasks incumbent on it.
- **T6-Elevation of Privilege:** An attacker might get elevated privileges having higher access amenity. Sometimes because of the lack of required protocol flow, an user can registered or for other reasons may receive higher privileges. For example, a patient get registered in the system as a physician or somehow has received the elevation of privilege which facilitate to make patient data altering request.

In addition to these, we have considered some additional threats which are crucial for the Hospital DT system.

- **T7-Replaying Transactions:** An attacker might capture an old transaction and submit it afterwards, thus launching a replay attack.



- **T8-Misuse of System Resources:** Without any concrete reason overuse of system’s calculation power. e.g., naive or with bad intentions, users may create multiple query requests for different purposes which will cause the system to do high cost calculations.

Different forms of data will be sent from entities to entities or from the system to the outside when the system interacts with a large number of entities. This poses a number of privacy risks to the system due to the lack of user privacy controls. Based on this assumption, the identified privacy threats are presented below:

- **T9-Lack of consent:** A transaction is being carried out without the consent of a user. e.g., a read or write operation has been conducted on a personal private data which the user is unaware of or does not know the identity of the person who has done it. It may happen in a decentralized network where geographically separate entities make transaction or data relevant operations on the same asset or data.
- **T10-Lack of control:** Data will be accessed by different parties from diverse domains having different trusts. So, an error may occur because of this byzantine access relation. For this reason each transaction or request needs to be recorded strictly.

## 4.4 Requirement Analysis

Every new service or system is developed in response to a demand from the necessary requirements [129]. The needed product or system sometimes does not coincide with the plan, even after investing a lot of time and money in development. Therefore, in order to prevent significant issues down the road, a focused and in-depth requirements analysis is required early on in any project.

To evaluate the requirements and expectations of a new model, a technique called requirements analysis or requirements engineering is utilized [130]. It entails regular communication with the product’s stakeholders and end users to clarify expectations, settle disputes, and record all essential requirements.

### 4.4.1 Functional Requirements (FR)

- F1. At any instance, the system should provide all the necessary data that will be needed to create a digital twin of an extant entity. e.g., when a doctor needs to create a digital twin of a patient’s heart, the system must supply all the up-to-date information regarding that patient’s sensor, diagnosis test, surgery, checkup prescription, and other relevant heart-related information up until that point.
- F2. Users can share their private information with organizations under the system’s control, which needs to be verified.
- F3. The system must verify the consistency of the saving of dynamic data from IoT devices.

- F4. The system should be integrated with a private blockchain infrastructure for the implementation of Digital Twin functionalities so that clinical transactions can be carried out satisfying different security requirements.
- F5. Each entity will be introduced with a unique ID in the system to accumulate all the pertinent data throughout the system.

#### **4.4.2 Security Requirements (SR)**

- S1. The system should ensure that only the authenticated and authorized users can access the corresponding data and participate in an activity. This mitigates T1 and T6.
- S2. Data needs to be managed and distributed securely to ensure the integrity, authenticity, and confidentiality of that data. This can mitigate T2, T3, and T4.
- S3. The system should take protective measures against any DoS attack so that it can deter T5.
- S4. The system must take protective measures against any replay attack in order to mitigate the T7 threat.
- S5. The system must be monitoring the misuse of resources and will restrict the users from overusing services. This will obviate T8.

#### **4.4.3 Privacy Requirements (PR)**

To remedy the privacy threats, privacy requirements play an important role. We present these requirements below:

- P1. The system must ensure that each transaction must be carried out only with the user's consent. This mitigates T9 threat.
- P2. The system must provide selective disclosure attribute privileges to its users so that the users can choose which fraction of data is needed to be shared. This mitigates T10 threat.

# Chapter 5

## System Architecture

### 5.1 Architecture

Before diving into the system architecture let us discuss some of the assumptions we have considered to successfully deploy the proposed system.

**Assumption 1** *The hospitals, which will be following the protocols of HDT, need to adopt adequate technologies to be under the support of HDT, otherwise, it would not be possible to get the services of DT.*

**Assumption 2** *The peripheral hospitals which are not under HDT need to have at least proper storage facilities, compatible servers, and government ordinance to comply with the proposed system so that HDT can extract patient's data from the peripheral hospitals.*

**Assumption 3** *To acquire the patient data from outside the system, there are a set of APIs available within the outside system which can provide a patient's data if all requirements are met.*

**Assumption 4** *The system can only be deployed in alignment with the government of the territory or nationwide for better access and data portability.*

The architecture of the proposed system is illustrated in Figure 5.1. In the architecture, the connection among the users and the equipment with the private blockchain platform for a hospital  $h \in H$  has been shown where  $H$  is a set of hospitals using the proposed Healthcare DT system,  $HDT$ , as per Assumption 1. Physicians, patients, and all other stakeholders affiliated with the healthcare services are considered as users and have the amenity (only the authorized one) to interact with the blockchain.

In the same manner, sensors and other IoT devices will send data pertinent to users to different components of the architecture, consequently the raw data will be stored in an off-chain database and their hash values and metadata will be stored as transactions in blockchain. To extract patient and other important data from peripheral hospitals, there will be a System,  $S$ .  $HP$  is the set of peripheral hospitals which are not under  $HDT$  and matches Assumption 2. With the help of  $S$ , hospitals  $H$  can extract data from the outer hospitals  $HP$ , which has been depicted in the architecture in Figure 5.1. By holistically looking at the architecture, it can be

perceived that there are 4 main components: Hospitals, Decentralized Application (DApp), System  $S$ , and Blockchain platform. The components will be explained elaborately in Section 5.2.

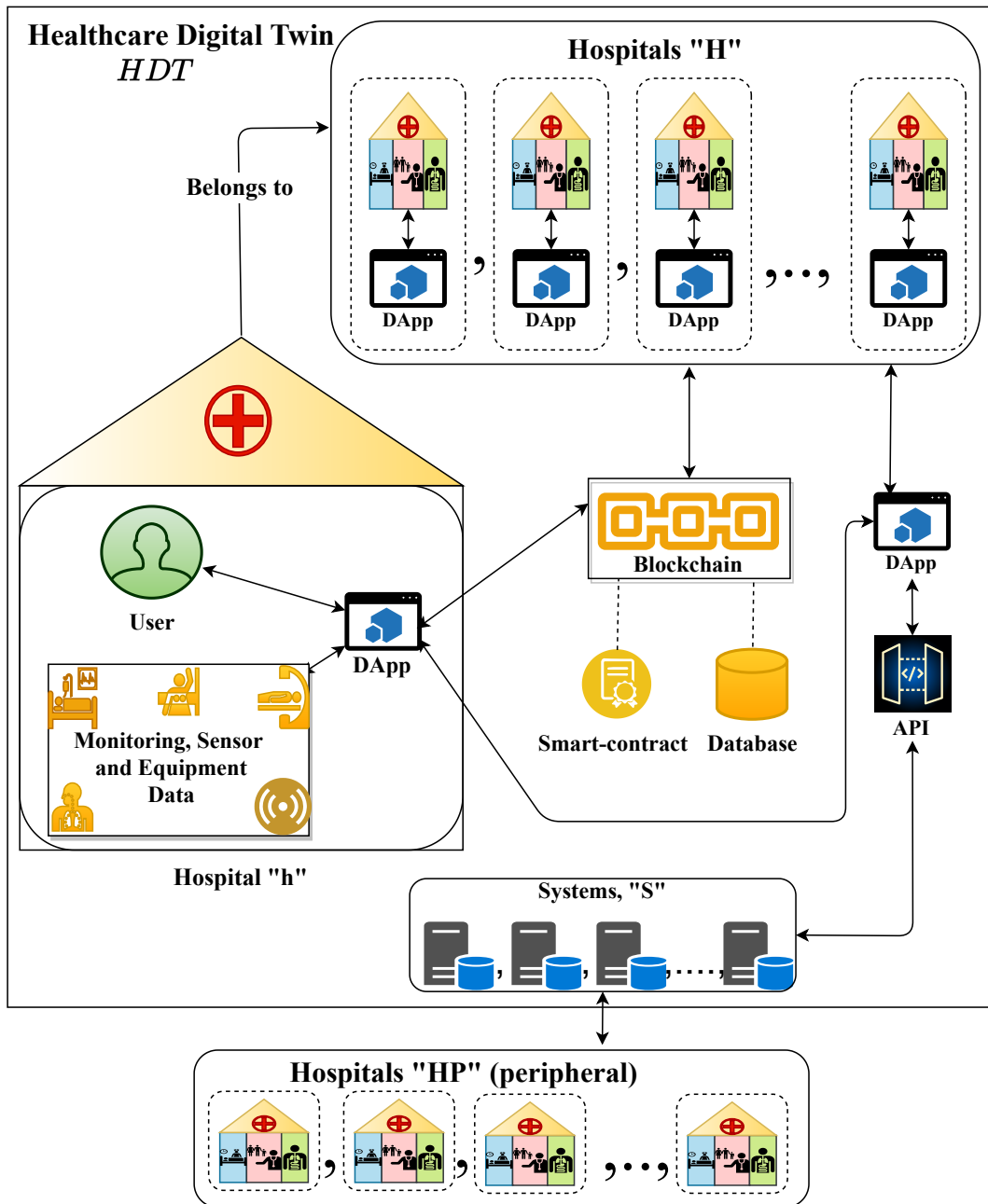


Figure 5.1: High-level Architecture and System Components

## 5.2 System Components

### 5.2.1 Hospitals

There are two types of hospitals in our architecture: one is governed by Healthcare DT ( $HDT$ ) and the other one is not.  $H$  is denoted as the set of hospitals which are

under the jurisdiction of  $HDT$  and connected to the blockchain. Now a hospital,  $h \in H$ , has two properties. They are:

- **User:** According to our provided data model, patients, physicians, surgeons, anesthetists, assistants, nurses, and administrators are the users of a hospital. With special permissions which are conferred to the users by the administrators, users can query or write data on the blockchain. They can also request external data from outside of  $HDT$ .
- **Equipment:** There are a myriad of IoT devices, which are test, monitoring, and sensor devices in a hospital. By interacting with blockchain, this equipment sends their dynamic or periodical data.

There is another type of hospital called peripheral hospitals denoted as  $HP$ , which are not under the jurisdiction of  $HDT$ . But, hospitals' ( $H$ ) users of  $HDT$  can extract data from  $HP$ . When the data is needed, it can be accumulated with proper structure through the system,  $S$  under  $HDT$ .

### 5.2.2 DApp

All the hospital ( $H$ ) users in our proposed architecture constantly need to interact with blockchain for querying and handling data. Normal web applications do not contain necessary tools or APIs to connect with blockchain and also blockchain runs on a distributed environment so the applications need to be apt enough to function on the same environment. For this reason, DApp, a decentralized application, can assist the users to interact with the blockchain. A DApp can run as a web server and by doing so it can render APIs to web applications which leads to a connection with the blockchain platform. In Figure 5.1, all the users of the hospital ( $h$ ) are connected to the blockchain via a DApp. There are a raft of IoT devices in the hospital generating dynamic and periodical data. This cluster of data will also be transferred to the blockchain through DApp, for this reason, the pieces of equipment of hospitals are also connected through DApp to the blockchain. DApp can also interact with smart contracts by transactions but a peer node will be the middleman. So, with the help of blockchain APIs, the DApp transactions will run on the blockchain platform and all the physical entities will be converged with the blockchain platform.

### 5.2.3 System, $S$

In general, patients will have former healthcare data stored in other hospitals ( $HP$ ) and to access that data, there will be a set of systems,  $S$  corresponding to each peripheral hospital. This  $S$  will render the chance to access the data from previously mentioned peripheral hospitals  $HP$  with the consent of specific data owners and the bridge between  $HDT$  and the external systems. There will be a DApp for each hospital in  $H$ , by which users can connect to each system in  $S$  with the help of APIs according to Assumption 3.

The forthcoming APIs facilitate  $HDT$  to acquire data form the respective systems of outside hospitals. For this, these outside hospitals could expose an API and our system will collect data from those APIs. These APIs traditionally are hosted in web servers and therefore, they will not be under blockchain network. Also, these

outside systems would store data in different formats. It will be the responsibility of these systems to convert those data according to our model before they are collected by our system. We are assuming that these systems utilise a biomedical data translator (e.g. [131]) for this purpose.

As the data is already stored in some data storage, when the data is necessary it can be brought back to *HDT*. For this reason, there is no need to store the data again in the blockchain. With our proposed data model from Section 4.2, most of the Pre-Hospital Admit Data (PHA) from Equation (4.3) can be collected in a codified way via a system in *S*.

#### 5.2.4 Blockchain Platform

Blockchain platform is one of the core components in our proposed Healthcare DT system (*HDT*). As we do not want our system to be public, moreover, the system will have only permitted users, we incorporate a private blockchain system. Hospital user and equipment are clients according to the blockchain network and are linked to the private blockchain platform through DApp. When a client wants to read, write or delete data with the means of a transaction request, DApp invokes the peer node to get a response against the transaction request. After that, the transaction response gets distributed to all the nodes in the blockchain network for validation. All the transaction requests, responses, and endorsements are stored as a block in blockchain in an immutable way. After adding a block, the blockchain also updates the records of current state for all the entities in the network according to the transactions in that specific block for faster data query. In terms of equipment generated dynamic and periodical data and user generated large amount of data, constantly creating transactions will be a burden for blockchain and will lag the system. Additionally, immutably storing patient healthcare data in the blockchain is not advisable and against some of the well-established regulations [132]. For this reason, the collected raw data will be stored in an off-chain database. But to keep the integrity of the data flow, a transaction will be compiled with the hash of the collected data, metadata, and the index of the off-chain database in the blockchain and the current state will be updated. Moreover, an amenity to delete or update data is available for *HDT*. After updating or deleting the raw data in the off-chain database, a new transaction will be compiled with the new hash, metadata, and indexes. In *HDT*, all the clients of the private blockchain network will be provided with certificates. With this any type of client like patients or sensors can be recognized by the system and assessed consequently. There will be a raft of users in the system and the data owners need to share their data among entities. By making policies, data access control can be defined among entities. The access control is not only confined in the scope of only one hospital, data can be shared through blockchain with other hospitals which are under *HDT*'s support.

# Chapter 6

## Implementation

### 6.1 Development

As we mentioned earlier, the data will be stored in 2 ways: on-chain and off-chain. The *Hyperledger Fabric* blockchain release version 2.2 has been used as the on-chain storage. Except for the extracted file data or raw data, all other types of data like metadata, user information, and so on, will be stored on-chain in *Hyperledger Fabric* blockchain. All the encrypted raw data will be stored off-chain in *BigchainDB* blockchain release version 4.2.1.

For showcasing and evaluating the output, we implement the algorithms presented in Section 6.2.2 in JavaScript. We have used Node.js which is a server-side JavaScript platform actively used in Blockchain development [133]. The experiment has been carried out in a PC with a Ubuntu 22.04.1 LTS Operating System, 5.15.0-58-generic kernel version, and hardware configurations of Intel(R) Core(TM) i5-7200U CPU @2.50GHz, 8 GB DDR4 RAM, 100 GB SSD, 2 TB HDD, and Intel(R) HD Graphics 620 GPU.

The *Hyperledger Fabric* provides a few NPM modules. Among them, the “fabric-network” facilitates developers to provide users’ wallet and gateways for the DApp to connect with the *Fabric* network. Moreover, the other *Fabric* module “fabric-client” provides all the necessary certificates and key-pairs to the participant entities so that each participants can be uniquely identified during transactions.

The set of smart contracts or in other word Chaincode has to be developed separately. The chaincode was developed using JavaScript. With the help of *Fabric* module “fabric-contract-api”, the developers can create a child class inheriting all the properties of *Hyperledger Fabric* provided “Contract” class. After developing the set of smart contracts, the Chaincode will be deployed on the designated *Fabric* Channel.

With the help of “bigchaindb-driver” module, we exploits the *BigchainDB* blockchain amenities inside our implementation. The *BigchainDB* is only used for storing the raw data off-chain, which will mitigate the burden from the *Hyperledger Fabric*.

To implement our proposed *HDT* system, we have considered the two most im-

portant type of user: patient and physician. So, there are two organization to divide the type of users and both of the organization have their own certificate authority (CA). Moreover, both of the user type have their own peer node as depicted in Figure 6.1. All the transactions are handled by one orderer node and the configuration of the orderer node is not part of the experiment. There is one channel called “mychannel” for incorporating the patient and physician peer nodes. Each of the peer node will be running with the accepted set of smart contracts, chaincode called “HDT” and will contain a copy of the ledger according to the Figure 6.1. *CouchDB* has been used for storing the ledger data.

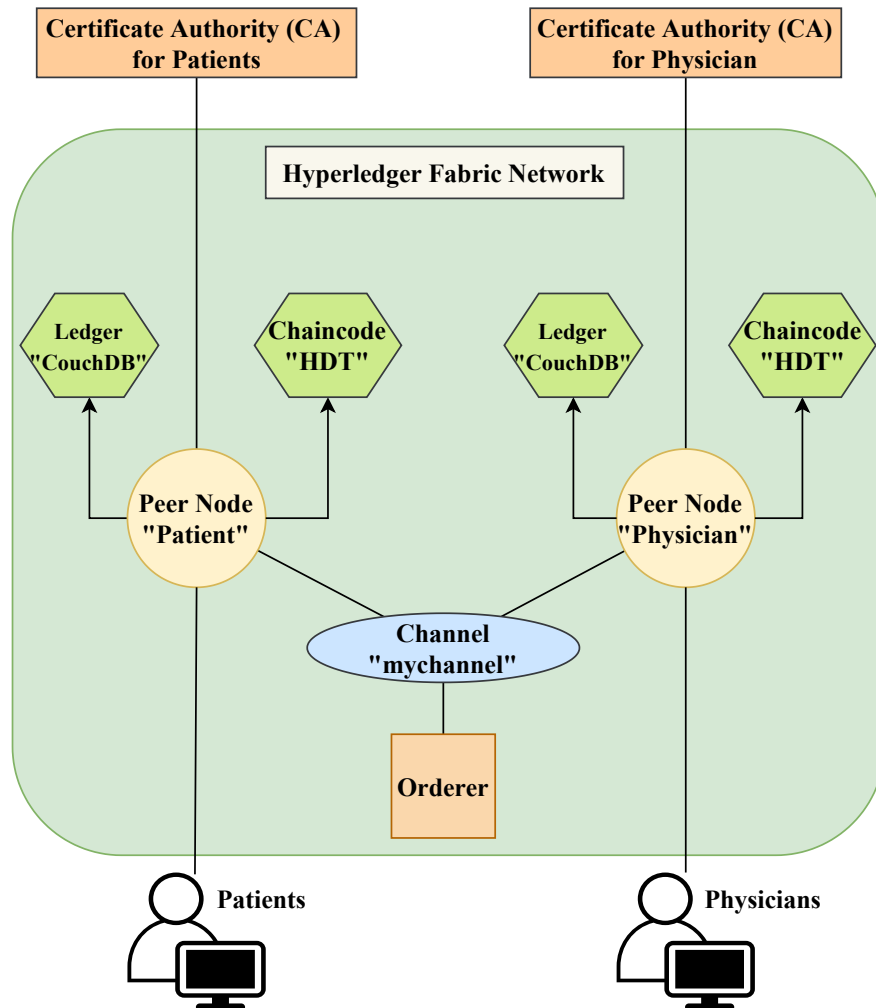


Figure 6.1: The Implemented *Hyperledger Fabric* Network for *HDT*.

Next we present the protocol flow in Section 6.2 in harmony with the conducted experiment.

## 6.2 Protocol Flow

As we are using private blockchain for *HDT*, no one can participate in the system without administrators’ consent. So, all the data flow will be considered between the legit members who are or will be accepted by the administrators. In this section, we will provide the protocol flow of different components in *HDT*. Before illustrating



the protocol flow we introduce the necessary notations in Table 6.1 and data model in Table 6.2.

Notation	Meaning
$U$	Users
$D$	DApp
$AD$	Administrator
$K_e$	Public key for entity $e$
$K_e^{-1}$	Private key for entity $e$
$N_i$	A fresh nonce
$\{\}_{K_e}$	Encryption operation using a public key $K_e$
$\{\}_{K_e^{-1}}$	Signature using a private key $K_e^{-1}$
$H(I)$	SHA-256 hashing operation of message $I$
$\square_{https}$	Communication over <i>HTTPS</i> Channel
$req$	Request
$resp$	Response
$response_{req}$	Response status according to $req$
$returnedData$	The additional or requested data according to $req$
$TYPE$	Request type
$DATA$	Data corresponds to $TYPE$
$regisData$	Collected data from a new user
$hsquery$	A type of $req$ to get health state
$hsQData$	Data corresponds to $hsquery$
$hdatawrite$	A type of $req$ to update or write health data
$writeData$	Data corresponds to $hdatawrite$
$delete$	A type of $req$ to delete data
$deleteData$	Data corresponds to $delete$
$share$	A type of $req$ to update or add policy
$shareData$	Data corresponds to $share$
$certificate$	The authenticity of user
$permissions$	Containing write or read operation access for a user
$c_n$	The series of given conditions to make a query request
$ZZ_{hdt}^{p,t}$	The aggregated returned data set against $hsquery$
$ZZ_h^{p,t}$	The resultant data set off-chain database under <i>HDT</i>
$ZZ_s^{p,t}$	The resultant data set queried from outside <i>HDT</i>
$WDT$	New unassigned generated data
$\emptyset$	null

Table 6.1: Cryptographic and other Notations.

### 6.2.1 Data Model

All the activities of the users in the system can be aggregated as request which is denoted as  $req$  in Table 6.2.  $req$  consists of  $type$  and  $data$ .  $TYPE$  denotes different request types from users to the system and  $type \in TYPE$ . Moreover,  $DATA$  represents the corresponding data of  $TYPE$  and  $data \in DATA$ . Whenever a  $req$  will be completed, a corresponding  $response_{req}$ , a natural response, will be provided by

$req \triangleq \langle type, data \rangle$
$resp \triangleq \langle response_{req}, returnedData \rangle$
$TYPE \triangleq \langle registration, login, hsquery, hdatawrite, delete, share \rangle$
$DATA \triangleq \langle regisData, loginData, hsQData, writeData, deleteData, shareData \rangle$
$regisData \triangleq \langle userName, userType, ha, constituentID \rangle$
$loginData \triangleq \langle userName, ha \rangle$
$certificate \triangleq \langle K_u, \{H(userName, userType, K_u)\}_{K_u^{-1}} \rangle$
$permissions \triangleq \langle \{(AA_q^{ad} \times AV_q^{ad}) \times t \mid AA_q^{ad} \& AV_q^{ad} \text{ is defined} \wedge t \in TS\} \rangle$
$hsQData \triangleq \langle userName, c_n, t \rangle$
$writeData \triangleq \langle userName, q, \{wdt\}_{K_u}, \{H(\{wdt\}_{K_u})\}_{K_u^{-1}}, \dots \rangle$
$WDT \triangleq \langle \{(AA_q^u \times AV_q^u) \times xx \times t \mid AA_q^u \& AV_q^u \text{ is defined} \wedge t \in TS\} \rangle$
$deleteData \triangleq \langle userName, q, \{H(q)\}_{K_u^{-1}} \rangle$
$shareData \triangleq \langle reqr, appr, q, \{H(q)\}_{K_u^{-1}} \rangle$

Table 6.2: Data Model.

the system. Additionally, there may be some system generated data which need to be returned to the user denoting *returnedData*. The *resp* will contain necessary response data (*response<sub>req</sub>*) and other additional data (*returnedData*) against the *req*. There are few types of *req* for which there will be no *returnedData* except *response<sub>req</sub>* for *resp*, and for these types of *resp*, *returnedData* will contain null ( $\emptyset$ ) value. *TYPE*, *DATA*, and *resp* are defined in Table 6.2.

In *TYPE*, *registration* denotes the action when a new user tries to join the system, consequently *regisData* in *DATA* contains the provided data by the user which is defined in Table 6.2. *ha* in *regisData* is the hash of the provided password,  $ha = H(password)$  and the *userName* is the provided name by the entity which will work as an identifier for the entity as a user. There are different types of entities like physician or patient, so entities need to apply in a category, *userType*, it wants to join. When a user  $u \in U$  will be accepted as an entity in the blockchain network, *response<sub>req</sub>* and *returnedData* will be generated. The *returnedData* for *registration* type *req* will mainly contain  $K_u$  (public key),  $K_u^{-1}$  (private key), *permissions*, and *certificate* for the new user *u*. One legit user may use others' public key for some malicious purposes for that reason administrator  $ad \in AD$  will give a signature with the hash value of *userName*, *userType*, and  $K_u$  to entitle them for the designated user *u*. By doing this, it will signify that the  $K_u$  is corresponded to that *userName* and *userType*. In *certificate*, user will be provided with this signature and  $K_u$  and it is defined in Table 6.2. There are different types of users with various degrees of access control. To codify this access control, each user will be provided with definite *permissions*. *permissions* is a set subsuming attributes and values regarding the data set, mentioned in Section 4.2, to do read, write, and write operation. In the definition of *permissions* in Table 6.2,  $q \in PD_{hdt}^p$  from Equation (4.15) and  $ad \in AD$ .

During *login* type of *req*, *userName* and *ha* will be taken from *u* and matched with *u*'s *userName* and *ha* from *regisData*. A patient type user can request to check the current health status of a patient, denoted as *hsquery* and the corre-

sponding data is  $hsQData$ . According to the scope of our system model, a user can only query the data available according to the data model presented in Section 4.2. By providing  $userName$ , conditions ( $c_n$ , here  $n = 1, 2, \dots, k$  or 0, where,  $k$  is the number of available conditions to make the request for data), and time ( $t$ ) for  $hsQData$ , a  $hsquery$  type  $req$  can be made. After the successful compilation of the  $req$ , an automatic response  $response_{req}$  with the resultant  $returnedData$  will be sent back to  $u$ .  $loginData$  and  $hsQData$  are defined in Table 6.2.

Whenever a user wants to write or update health data in blockchain, there will be a transaction to convey the request  $hdatawrite$ . The user needs to provide its  $userName$  as identity, the data ( $q \in \{PDD_h^p \cup SOP_h^p\}$  from Equation (4.7) and (4.14)) it wants to write, the new data  $wdt \in WDT$  in encrypted form  $\{wdt\}_{K_u}$ , and the user signature ( $\{H(\{wdt\}_{K_u})\}_{K_u^{-1}}$ ) of the hash value of new encrypted data, collectively can be represented as  $writeData$  which corresponds to the  $req$  type  $hdatawrite$ .  $WDT$  is the set of all newly unassigned generated data, where  $xx$  is the necessary data to complete the data set according to the data model defined in Section 4.2. For this  $req$  in the  $resp$ , there will be no  $returnedData$  except a  $response_{req}$  corresponds to the  $req$ . The  $writeData$  is defined in Table 6.2. In the same manner for the  $delete$  request, the data owner can discard the association of data from  $BigchainDB$  by updating  $Fabric$ 's transaction log, subsequently removing the world state which was storing the associative metadata. For  $delete$  type  $req$  there will be only  $response_{req}$ . The  $deleteData$  is defined in Table 6.2. There will be another type of request, sharing certain personal data with another user denoted as  $share$ . To make this request  $shareData$  contains requester's  $userName$  ( $reqr$ ), data owner's  $userName$  ( $appr$ ), the data properties ( $q \in PD_{hdt}^p$ ), and the signature of the requester ( $\{H(q)\}_{K_R^{-1}}$ ) to make the  $share$  type  $req$ . The  $shareData$  is defined in Table 6.2.

## 6.2.2 Algorithm

We present the algorithms of  $HDT$  smart contracts in Algorithm 1 and Algorithm 2. The prime functionalities of our system are registration, login, health state query, write data, and data share operation. The  $start$  function is the starting point for smart contracts in both Algorithm 1 and Algorithm 2. Depending on the type of the request ( $req$ ), different functions are called in smart contracts (Line 1 to Line 8 in Algorithm 1 and Line 1 to Line 10 in Algorithm 2). In terms of Algorithm 1, after retrieving the data (Line 2), any of the two functions,  $regisFunc$  or  $loginFunc$  will execute if nothing goes wrong. Here, the  $loginFunc$  encodes the logic for the login functionality whereas the  $regisFunc$  encodes the registration functionality. After the completion of executing a function,  $resp$  will be returned to DApp (Line 10). On the other hand, the smart contract in Algorithm 2 will handle three functions denoting  $hsqueryFunc$ ,  $writeFunc$ , and  $dataShareFunc$ .  $hsqueryFunc$  deals with the retrieval of health state query data from blockchain as well as external systems. The  $writeFunc$  encodes the functionalities for writing data and the  $dataShareFunc$  deals with recording data sharing properties. When the  $start$  receives a  $req$  it will retrieve data the same way before (Line 2). Depending on the  $req$  type, the corresponding functions will be called and after the execution  $resp$  will be sent back to DApp (Line 12).

---

**Algorithm 1** Smart Contracts of Healthcare Digital Twin: Registration and Login

---

**Input:**  $req$  : Request from the user

**Output:**  $resp$  : Response against the req

```
1: function start( $req$ )
2:    $this.data = req.data$ ;
3:   if ( $req.type == registration$ ) then
4:      $resp = regisFunc(data)$ ;
5:   else if ( $req.type == login$ ) then
6:      $resp = loginFunc(data)$ ;
7:   else
8:     return  $error$ ;
9:   end if
10:  return  $resp$  to  $DApp$ ;
11: end function
12: function regisFunc( $data$ )
13:   $uName = data.userName$ ;
14:   $uType = data.userType$ ;
15:   $uHashPassword = data.ha$ ;
16:   $uglobalId = data.constituentID$ ;
17:   $ujson = \{Type : uName, Password : uHashPassword, NationalID : uglobalId\}$ ;
18:   $putState(uName, ujson)$ ; ▷ Store into blockchain
19:   $permissions = System\ provided$ ;
20:   $response_{req} = Successfully\ Registered$ ;
21:   $returnedData = permissions$ ;
22:   $resp = response_{req} + returnedData$ ;
23:  return  $resp$ ;
24: end function
25: function loginFunc( $data$ )
26:   $uName = data.userName$ ;
27:   $uHashPassword = getState(uName)$ ; ▷ Retrieve form blockchain
28:  if ( $uHashPassword == data.ha$ ) then
29:     $response_{req} = Successfully\ Logged\ In$ ;
30:  else
31:     $response_{req} = Error$ ;
32:  end if
33:  return  $response_{req}$ ;
34: end function
```

---

### 6.2.3 Protocol flow

Now, we depict the protocol flow illustrating user interactions with different functions of *HDT*.

**Registration Protocol:** To participate, a user must register following the protocol presented in Table 6.3 and illustrated in Figure 6.3. The  $req$  for the registration request is comprised of *registration* type and *regisData*. The  $userName$ ,  $userType$ , the hash of password ( $ha$ ), and  $constituentID$  are provided in the *regisData*. New users, who will register in a hospital  $h \in H$  under  $hdt \in HDT$  and was involved with

---

**Algorithm 2** Smart Contracts of Healthcare Digital Twin: Health State Query, Health Data Write, and Sharing Data

---

**Input:**  $req$  : Request from the user

**Output:**  $resp$  : Response against the req

```
1: function start( $req$ )
2:    $this.data = req.data$ ;
3:   if ( $req.type == hsquery$ ) then
4:      $resp = hsqueryFunc(data)$ ;
5:   else if ( $req.type == hdatawrite$ ) then
6:      $resp = writeFunc(data)$ ;
7:   else if ( $req.type == share$ ) then
8:      $resp = shareFunc(data)$ ;
9:   else
10:    return error;
11:  end if
12:  return  $resp$  to  $DApp$ ;
13: end function
14: function hsqueryFunc( $data$ )
15:    $uName = data.userName$ ;
16:    $uconditions = data.cn$ ;
17:    $utime = data.t$ ;
18:    $ujson = getState(uName)$ ;  $\triangleright$  Retrieve the metadata from blockchain
19:    $TxID = \text{Sorting the } BigchainDB \text{ Transactions from } ujson$ ;
20:   return  $BigchainDB.TxID$ ;
21: end function
22: function writeFunc( $data$ )
23:    $uName = data.userName$ ;
24:    $uDataProperty = data.q$ ;
25:    $uTxID = data.BigchainDB.TransactionID$ ;
26:    $boolean\ x = \text{Store new data with a transaction and return a Boolean}$ ;
27:   if ( $x == TRUE$ ) then
28:      $response_{req} = \text{Transaction Successful}$ ;
29:   else
30:      $response_{req} = \text{Transaction Failed}$ ;
31:   end if
32:   return  $response_{req}$ ;
33: end function
34: function shareRecordFunc( $data$ )
35:    $urequester = data.repr$ ;
36:    $uapprover = data.appr$ ;
37:    $umetadata = data.q$ ;
38:    $usignature = data.\{H(q)\}_{K_R^{-1}}$ ;
39:    $ujson = getState(uapprover)$ ;  $\triangleright$  Retrieve the metadata from blockchain
40:    $BigchainDB.TxID = \text{Transactions from } ujson$ ;
41:    $shareRecordjson = \{Sender : uapprover, Receiver : urequester, Data :$ 
42:      $umetadata, Sign : usignature\}$ ;
43:    $\text{Store shareRecordjson with a transaction and returns a Boolean}$ ;
44:   return  $BigchainDB.TxID$ ;
45: end function
```

---

the services under a peripheral hospital,  $hp \in HP$ , for them  $h$  needs to create a gateway through system,  $s \in S$ , to access external data specially  $PHA_s^p$  (Equation (4.3)) of the patients from  $hp$ . For this reason, users need to provide their national digital constituent number with which the users can be recognizable to the outer systems according to Assumption 4. With this digital constituent number ( $constituentID$ ) hospital  $H$  can request data for respective users from peripheral hospitals  $HP$  by the means of system  $S$ .

- **Step 1:** According to the registration protocol, the first message  $M1$  defined in Table 6.3, a new user sends a nonce ( $N_1$ ) and the  $req$  encrypted with the public key ( $K_D$ ) of DApp (D) to the DApp (D) over an  $HTTPS$  channel. The request can be made form the user interface as illustrated in Figure 6.2.

The image shows a web form titled "Register New User:". It contains the following fields and values:

- Name:** Sadman Sakib Akash
- Constituent ID:** 5084170405
- User Type:** Patient (selected from a dropdown menu)
- Email:** sadmansakibakash@gmail.com
- Password:** [Masked with 10 dots]

A blue "Register" button is located below the password field.

Figure 6.2: The User Interface for Registration Request

- **Step 2:** Receiving the message, D decrypts the request with its private key ( $K_D^{-1}$ ) and forwards the request to smart contracts (SC) ( $M2$  in Table 6.3). Now,  $regisFunc$  function in Algorithm 1 handles the provided  $regisData$  firstly by extracting it (from Line 13 to Line 16).
- **Step 3:** After that, SC aggregates the user data in Line 17 and stores it into blockchain (BCH) according to message  $M3$  in Table 6.3 (Line 18).
- **Step 4:** Later, BCH creates a transaction and returns  $TRUE$  for a successful completion of registration ( $M4$  in Table 6.3). Then, SC provides an access control set entitled as  $permissions$  in Line 19 and the  $returnedData$  contains it (Line 21). A general response according to the  $req$  which is "Successfully Registered" is being provided as  $response_{req}$  (Line 20).
- **Step 5:** Then a final response  $resp$  containing  $response_{req}$  and  $returnedData$  is being returned to D in Line 23 and it is defined as message  $M5$  in Table 6.3. Next, D generates a pair of public and private key ( $K_u$  and  $K_u^{-1}$ ) for user  $u$  which is illustrated in Figure 6.3. Moreover, for storing the raw data off-chain in  $BigchainDB$ , another pair of public and private key will be generated for a new user. Additionally, a key pair of RSA (Rivest–Shamir–Adleman) will be generated for the corresponding user to encrypt and decrypt AES (Advanced Encryption Standard) key. All the created key-pairs will be stored

in the local machine and the security of the stored key-pairs are not the scope of this reserach work. An administrator,  $ad \in AD$  signs the hash of  $K_u$ ,  $userName$ , and  $userType$ , collectively denoted as *certificate*. Now, *returnedData* includes  $K_u^{-1}$ , *certificate*, and *permissions* (illustrated in Figure 6.3).

- **Step 6:** Finally, the updated *resp* and hash of *resp* signed ( $\{H(resp)\}_{K_D^{-1}}$ ) by D is being returned to  $u$  over an *HTTPS* channel (M6 in Table 6.3). The user stores its public and private keys in device for any future correspondence.

M1	$u \rightarrow D :$	$[N_1, \{req\}_{K_D}]_{https}$
M2	$D \rightarrow SC :$	$N_2, req$
M3	$SC \rightarrow BCH :$	$N_3, \{uName, ujson\}$
M4	$BCH \rightarrow SC :$	$N_3, TRUE$
M5	$SC \rightarrow D :$	$N_2, resp$
M6	$D \rightarrow u :$	$[N_1, resp, \{H(resp)\}_{K_D^{-1}}]_{https}$

Table 6.3: Registration Protocol.

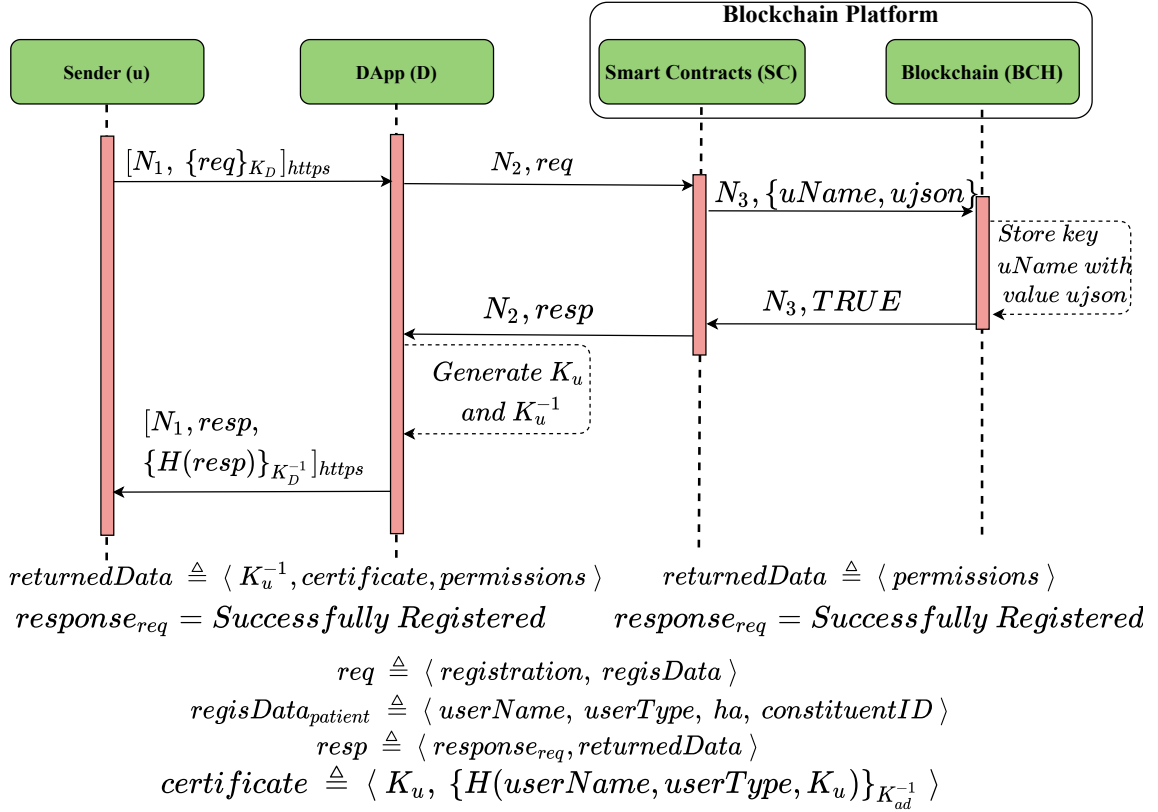


Figure 6.3: Registration Flow.

**Login Protocol:** Every user must log in before accessing the system. The user  $u$  will provide *loginData* incorporating *userName* and hash password (*ha*) in a *login* type *req*. Figure 6.4 depicts a login request. With this request, *start* relays the data to *loginFunc* where the request is being handled (from Line 26 to Line 33



in Algorithm 1). A successful validation will sign in the user to the system. For security, every request and response between the user and the DApp are transmitted over an *HTTPS* channel.

Figure 6.4: The User Interface for Login Request

**Write Data Protocol:** Now we present the protocol flow for writing data on blockchain. The process follows the protocol flow of Table 6.4 and Figure 6.5.

$M1$	$u \rightarrow D:$	$[N_1, \{req\}_{K_D}]_{https}$
$M2$	$D \rightarrow BNB:$	$N_2, \{wdt\}_{K_u}$
$M3$	$BNB \rightarrow D:$	$N_2, Index$
$M4$	$D \rightarrow SC:$	$N_3, req$
$M5$	$SC \rightarrow BCH:$	$N_4, \{uName, uDataProperty, uIndex\}$
$M6$	$BCH \rightarrow SC:$	$N_4, TRUE$
$M7$	$SC \rightarrow D:$	$N_3, resp$
$M8$	$D \rightarrow u:$	$[N_1, resp, \{H(resp)\}_{K_D^{-1}}]_{https}$

Table 6.4: Write Data Protocol.

- **Step 1:** User  $u$  sends  $req$  encrypted with the public key ( $K_D$ ) of DApp (D) as well as a nonce ( $N_1$ ) to DApp (D) according to the message  $M1$  in Table 6.4 over an *HTTPS* channel. Figure 6.6 depicts a write data request user interface.
- **Step 2:** Firstly, D decrypts the  $req$  subsuming  $hdatawrite$  and  $writeData$ .  $writeData$  contains  $userName$ , data property ( $z$ ), the encrypted new data  $\{wdt\}_{K_u}$ , and the signature  $\{H(\{wdt\}_{K_u})\}_{K_u^{-1}}$ . Here the data encryption has been conducted using a random AES key (32 Bytes). One notable point is that user cannot change the data under System,  $S$  as that data has only read operation access to all users. To write or update, user needs to specify which data ( $q \in \{PD_{hdt}^p \setminus PHA_s^p\}$ ) from data model he wants to write or update with the new data. After that, D sends the encrypted data ( $\{wdt\}_{K_u}$ ) to off-chain database (BNB) according to  $M2$  in Table 6.4. According to our implementation the encrypted data will be stored in *BigchainDB* and the corresponding transaction is presented in Figure 6.7.
- **Step 3:** BNB stores the encrypted data and returns the transaction ID (TxID) of the stored data to D ( $M3$  in Table 6.4) as illustrated in Figure 6.5.



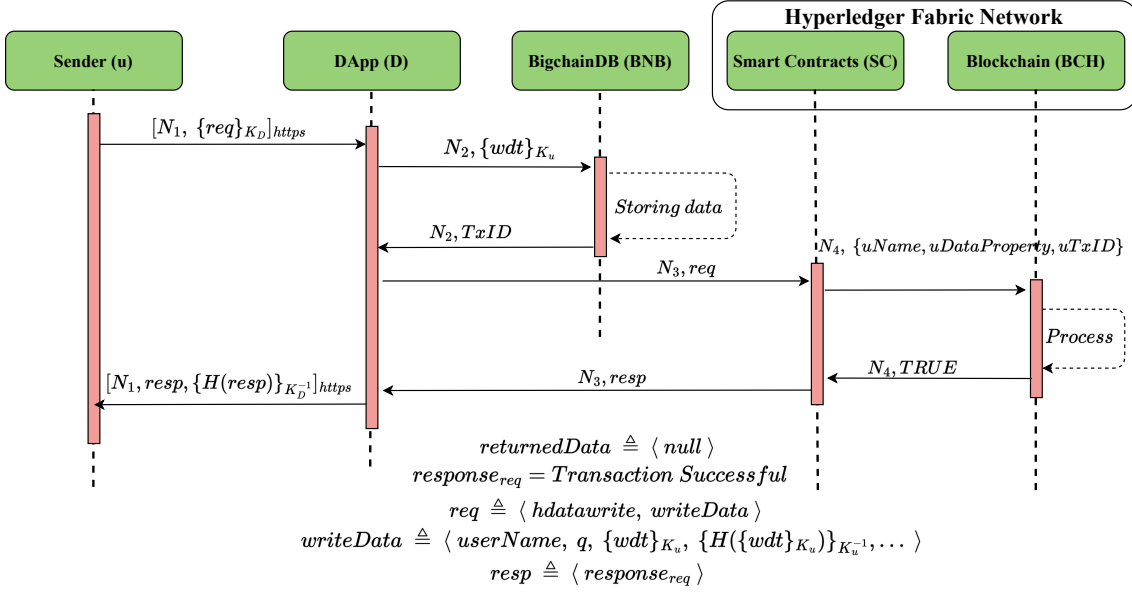


Figure 6.5: Write Data Flow.

- **Step 4:** After receiving the confirmation of the stored data from BNB, D encrypts the AES key with the RSA public key of the corresponding user. Now D replaces the encrypted data property of  $writeData$  with  $TxID$  and AES key Cypher then forwards the  $req$  to smart contracts (SC) (M4 in Table 6.4).
- **Step 5:** The  $hdatawrite$  type  $req$  will be handled by function  $writeFunc$  in Algorithm 2. At first, SC extracts the data from  $writeData$  (from Line 23 to Line 25). Then SC stores the data into blockchain (BCH) by providing the desideratum data according to M5 in Table 6.4 (Line 26).
- **Step 6:** BCH creates a transaction for writing the healthcare data on BNB and for a successful transaction operation a Boolean response is being returned to SC (M6 in Table 6.4). The corresponding transaction is presented in Figure 6.8.
- **Step 7:** A general response ( $response_{req}$ ) “Transaction Successful” otherwise “Transaction Failed” will be provided based on the Boolean result (from Line 27 to Line 30). There is no  $returnedData$  because there is no additional data for this  $req$ . The  $resp$  encapsulating  $response_{req}$  is being returned to D from SC according to M7 in Table 6.4.
- **Step 8:** Now, after signing the hash of  $resp$ , D returns  $\{N_1, resp, \{H(resp)\}_{K_D^{-1}}\}$  to user in congruous with M8 in Table 6.4 over an  $HTTPS$  channel.

**HealthState Query Protocol:** The health state query follows the protocol represented in Table 6.5 and Table 6.6. For this request, the  $req$  consists of type  $hsquery$  and  $hsQData$ , containing the data corresponds to  $hsquery$ . As we are developing a patient centric system, so for this research scope only the patient type  $u$  can make this type of query.  $hsQData$  contains the  $userName$  of the patient type user who is making the request for checking health state, conditions ( $c_n$ ), and the time frame ( $t$ )

**Upload your file and Metadata:**

\* Patient Stage:

\* Data Type:

\* Patient Email ID:

Upload the File:  CaregiverList.pdf  
 Filename: CaregiverList.pdf  
 Filetype: application/pdf  
 Size in bytes: 70254  
 lastModifiedDate: 7/25/2022

Figure 6.6: The User Interface for Write Data Request

of the requested data.  $c_n$  contains the available predefined conditions. For example, if a user  $u$ , who is a patient, wants to query ( $hsquery$ ) about surgery data then  $u$  will provide  $userName$ ,  $c_n$  ( $c_1 = Surgery - Information$ ,  $c_2 = Surgery - Type$ ), and tentative time period  $t$  of the requested data as the  $hsQData$ .

M1	$u \rightarrow D$ :	$[N_1, \{req\}_{K_D}]_{https}$
M2	$D \rightarrow SC$ :	$N_2, req$
M3	$SC \rightarrow BCH$ :	$N_3, userName$
M4	$BCH \rightarrow SC$ :	$N_3, ujson$
M5	$SC \rightarrow D$ :	$N_2, TxID$
M6	$D \rightarrow BNB$ :	$N_4, TxID$
M7	$BNB \rightarrow D$ :	$N_4, \{ZZ_h^{p,t}\}_{K_u}$
M8	$D \rightarrow u$ :	$[N_1, resp, \{H(resp)\}_{K_D^{-1}}]_{https}$

Table 6.5: HealthState Query Protocol under system  $H$ .

After the successful compilation of the  $req$ , automated  $response_{req}$  is being generated with  $returnedData$ . Here, the  $returnedData$  comprises of a set of data regarding the request, denoting  $ZZ_{hdt}^{p,t}$ , which is defined in Table 6.1. Here,  $ZZ_{hdt}^{p,t}$  represents an aggregated data set  $ZZ$  under system  $hdt \in HDT$  for patient  $p \in P$  (here,  $p = u$ ) where the  $c_n$  conditions are evaluated for the time period  $t$  and  $ZZ_{hdt}^{p,t} \subseteq PD_{hdt}^p$  from Equation (4.15). The resultant data,  $ZZ_{hdt}^{p,t} = \{ZZ_h^{p,t} \cup ZZ_s^{p,t}\}$  where  $ZZ_h^{p,t}$  is the result data set coming from off-chain database under system  $h \in H$  and  $ZZ_s^{p,t}$  is coming from outside  $HDT$  which falls under system  $s \in S$ . According to the mentioned scenario,  $ZZ_{hdt}^{p,t}$  will contain all the relevant data regarding that specific  $Surgery - Type$  for  $p$  under system  $h$  and  $s$ :  $\{POA_h^{p,t} \cup ST_h^{p,t} \cup SOS_h^{p,t} \cup SSF_h^{p,t} \cup PSG_s^{p,t}\}$ . Here,  $PSG_s^{p,t}$  comes from  $ZZ_s^{p,t}$  and other data sets come from  $ZZ_h^{p,t}$ . Though, the data will come in encrypted form so without the designated user no one can read the resultant data. By not providing any  $c_n$ , patients can query all the

```

{
  asset: {
    data: {
      FileHash: '3WnQiUXYY/x9mdYoDUAf1hN61kR2Z/UAjfMEiG3CXko=',
      FileName: 'CaregiverList.pdf',
      FileType: 'application/pdf',
      RawFileData: [Object],
      Size: 70254
    }
  },
  id: '2a00cafacc83fb71d94501bba65c2f57e750ecd7b4e993594d95c914654208989',
  inputs: [
    {
      fulfillment: 'pGSAIK1BQcxecgrc-gLrEiJTulMqyyUud0ugAwS5F15XzyKfgUB4BJYlpEsq1UBoNtQTfGy-z_9PoD',
      fulfills: null,
      owners_before: [Array]
    }
  ],
  metadata: { 'Time of Submitting Transaction': '2023-01-29T18:34:56.847Z' },
  operation: 'CREATE',
  outputs: [ { amount: '1', condition: [Object], public_keys: [Array] } ],
  version: '2.0'
}

```

Figure 6.7: A *BigchainDB* Transaction for storing a file’s encrypted raw data.

```

{
  Key: 'file_sadmansakibakash@gmail.com_2a00cafacc83fb71d94501bba65c2f57e750ecd7b4e993594d95c914654208989',
  DataOwner: 'sadmansakibakash@gmail.com',
  PatientStage: 'Patient Disease Diagnose Data',
  DataType: 'Caregiver Data',
  FileName: 'CaregiverList.pdf',
  BigchainDBTransactionID: '2a00cafacc83fb71d94501bba65c2f57e750ecd7b4e993594d95c914654208989',
  AESCipher: '7780b6e96fe884469e55c289feb307398f9e3da37df041e0d3e6f20ef209cc28ded1fe46fb839844f894acd1f011ae7dffdb55d756b0e0fcc82212e221255aafd96087034a895ea14c1de87df54fc1a4d4e1b902cda53fb3ac9dcca64a56431f7b710c6964c803e1e6f991feb8a6b420dce81387e91e8414e3e18b97116d7059ba9ecd2493345c55d88fb90d5adb69cf940d444943e961f2fb1cb069dbc3ff224d46ea54ef556ef0eb1273b4d62becabaa0c3aca35cd9b23085185bc88f14846f7771e185052c36ad02bb577c',
  FileHash: '3WnQiUXYY/x9mdYoDUAf1hN61kR2Z/UAjfMEiG3CXko=',
  UploaderEmail: 'sadmansakibakash@gmail.com',
  UploadTime: 'Mon Jan 30 2023 00:34:58 GMT+0600 (Bangladesh Standard Time)',
  DocType: 'File'
}

```

Figure 6.8: A *Hyperledger Fabric* Transaction for Write Data Request.

data. Now it is understandable that, the actions will conduct under  $h$  and outside  $HDT$  under  $s$ , for this reason, the flow has been divided into two parts (Table 6.5 and Table 6.6 for system  $h$  and  $s$  respectively). Assuming user  $u$  is logged in and will compile the request disregard of which system data is coming from.

- **Step 1 for both  $h$  and  $s$ :** According to the same message  $M1$  defined in both Table 6.5 and Table 6.6,  $u$  sends the  $req$  encrypted with the public key ( $K_D$ ) of DApp (D) to the DApp (D) under hospital  $h$ . The request can be made form the user interface as illustrated in Figure 6.9.
- **Step 2 for both  $h$  and  $s$ :** After completion of the decryption process,  $hsQData$  containing  $userName$ ,  $c_n$ , and  $t$  will be assessed in D and depending on the required data sources D will forward it to either smart contracts (SC) ( $M2$  in Table 6.5) or DApp ( $D_s$ ) connected to system,  $S$  ( $M2$  in Table 6.6) or both.
- **Step 3 for  $h$ :** The  $req$  that is passed on to SC is being handled by  $hsqueryFunc$  starting with the retrieval of  $hsQData$  data (from Line 15 to Line 17) in Algorithm 2. Then SC retrieves all the data pertinent to  $uName$  from Blockchain

M1	$u \rightarrow D:$	$[N_1, \{req\}_{K_D}]_{https}$
M2	$D \rightarrow D_s:$	$N_2, req$
M3	$D_s \rightarrow hp_s:$	$N_3, req$
M4	$hp_s \rightarrow D_s:$	$N_3, \{ZZ_s^{p.t}\}_{K_u}$
M5	$D_s \rightarrow D:$	$N_2, \{ZZ_s^{p.t}\}_{K_u}$
M6	$D \rightarrow u:$	$[N_1, resp, \{H(resp)\}_{K_D^{-1}}]_{https}$

Table 6.6: HealthState Query Protocol under system  $S$ .

**Please! Provide All the Conditions for Query Processing:**

\* Patient Email ID:

\* Patient Stage:

\* Data Type:

\* Start Date:

\* End Date:

Figure 6.9: The User Interface for Health State Query Request

(BCH) in Line 18 ( $M3$  in Table 6.5) which returns the  $ujson$  data set according to  $M4$  in Table 6.5.

- **Step 4 for  $h$ :** After finding out the  $BigchainDB.TxID$  from the  $ujson$  for the queried data in Line 19, SC returns the  $TxID$  to  $D$  in congruous with  $M5$  in Table 6.5 (Line 20).
- **Step 5 for  $h$ :** Without changing anything,  $D$  also relays the  $TxIDs$  toward  $BigchainDB$  (BNB) according to  $M6$  in Table 6.5.
- **Step 6 for  $h$ :** Then, BNB retrieves the  $TxIDs$ ' encrypted data ( $\{ZZ_h^{p.t}\}_{K_u}$ ) as depicted in Figure 6.10 and returns it to  $D$  ( $M7$  in Table 6.5).
- **Step 3 for  $s$ :** On the other hand, in terms of  $D_s$ , after receiving the  $req$  from Step 2,  $D_s$  forwards it to peripheral hospital ( $hp_s$ ) with the help of APIs ( $M3$  in Table 6.6).
- **Step 4 for  $s$ :** The  $hp_s$  conducts the necessary processing and sends back the resultant encrypted data  $\{ZZ_s^{p.t}\}_{K_u}$  to  $D_s$  in congruous with  $M4$  in Table 6.6 as depicted in Figure 6.11.
- **Step 5 for  $s$ :** Without any alteration,  $D_s$  relays back the data to  $D$  according to message  $M5$  in Table 6.6.

The DApp ( $D$ ) aggregates both received data ( $\{ZZ_h^{p.t}\}_{K_u}$  and  $\{ZZ_s^{p.t}\}_{K_u}$ ) and stores it as  $returnedData$  ( $\{ZZ_h^{p.t}\}_{K_u} \cup \{ZZ_s^{p.t}\}_{K_u}$ ). Moreover, depending on the result of the query request a general  $response_{req}$  is created and after that  $D$  returns the

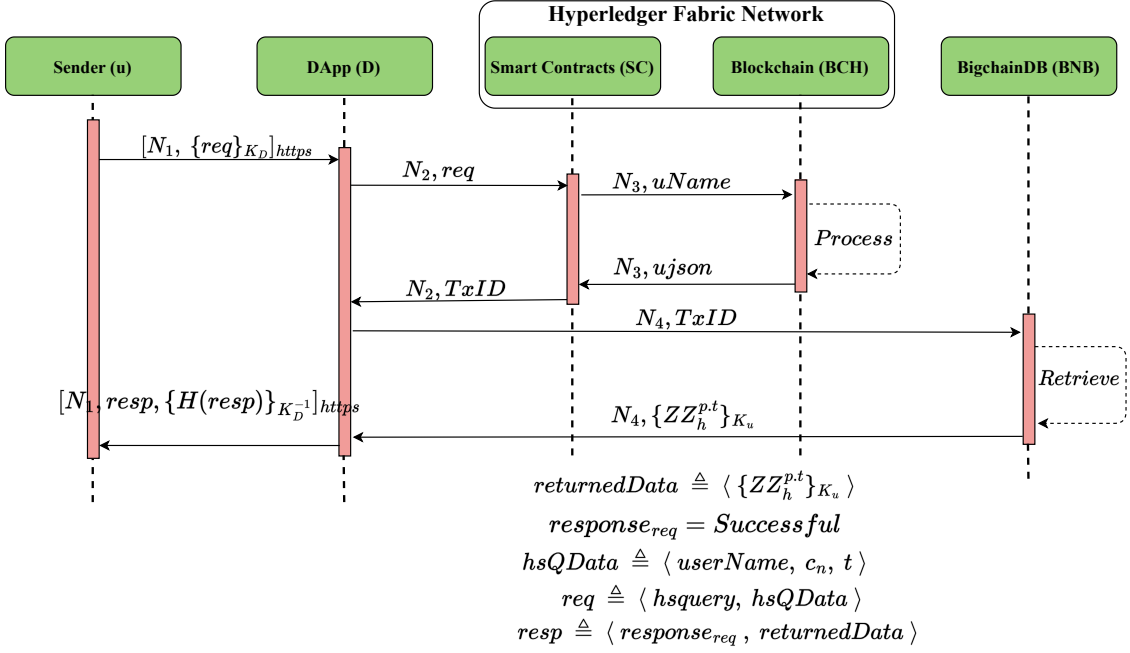


Figure 6.10: HealthState Query Flow under system  $H$ .

encapsulated  $resp$  with its signature ( $\{H(resp)\}_{K_D^{-1}}$ ) to  $u$  ( $M8$  in Table 6.5 and  $M6$  in Table 6.6 concurrently). For security concerns all the data interactions from user to DApp are over an  $HTTPS$  channel. After decrypting all the queried data, the mentioned resultant data can be achieved,  $ZZ_{hdt}^{p,t} = \{ZZ_h^{p,t} \cup ZZ_s^{p,t}\}$ . However for this thesis, we have not considered the data from System,  $S$ . So only the retrieved data ( $ZZ_h^{p,t}$ ) from internal sources can be queried as depicted in Figure 6.12. At first the AES key Cipher will be decrypted with the corresponding user's RSA private key. Then the AES key will be used to decrypt the returned data ( $ZZ_h^{p,t}$ ) according our implementation.

**Delete Data Protocol:** The protocol for deleting data follows the protocol flow of Table 6.7 and is illustrated in Figure 6.13.

$M1$	$u \rightarrow D:$	$[N_1, \{req\}_{K_D}]_{https}$
$M2$	$D \rightarrow SC:$	$N_2, req$
$M3$	$SC \rightarrow BCH:$	$N_3, userName, DataProperty$
$M4$	$BCH \rightarrow SC:$	$N_3, TRUE$
$M5$	$SC \rightarrow D:$	$N_2, resp$
$M6$	$D \rightarrow u:$	$[N_1, resp, \{H(resp)\}_{K_D^{-1}}]_{https}$

Table 6.7: Delete Data Protocol.

- **Step 1:** User and data owner  $u$  sends the  $req$  encrypted with the public key ( $K_D$ ) of DApp (D) and a nonce ( $N_1$ ) to the D in accordance with  $M1$  in Table 6.7 over an  $HTTPS$  channel.
- **Step 2:** After decrypting the  $req$ , D forwards it to SC according to  $M2$  in Table 6.7.

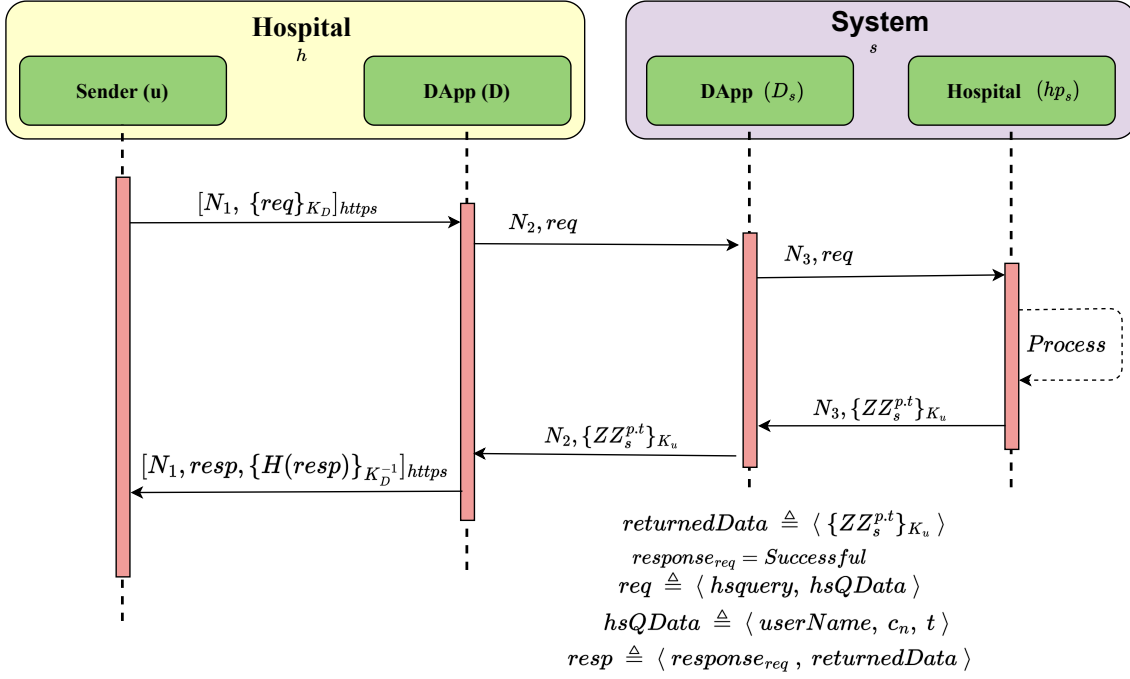


Figure 6.11: HealthState Query Flow under system  $S$ .

- **Step 3:** Now, SC sends the `userName` and `Data Property` to blockchain (BCH) in accordance with  $M3$  which is illustrated in Figure 6.13.
- **Step 4:** Then, SC retrieves the `BigchainDB` transaction ID ( $TxID$ ) of the data from BCH ( $M4$  in Table 6.7) and deletes the corresponding transaction from `Fabric`'s world state concurrently by creating a transaction. As a result the association of the stored data in `BigchainBD` with the `Fabric` network will be severed (Figure 6.13).
- **Step 5:** SC returns a Boolean value to D ( $M5$  in Table 6.7).
- **Step 6:** Finally, after signing the hash of `resp`, D returns it to  $u$  in congruous with  $M6$  in Table 6.7 over an `HTTPS` channel.

**Share Data Protocol:** The protocol flow for sharing data follows the protocol flow of Table 6.8 and is illustrated in Figure 6.14. To share data with other users, the protocol starts with the requester  $R$ , creating `share` type `req` and `shareData` contains all the corresponding data of `share` which are requester  $R$ 's `userName` (`reqr`), approver  $A$ 's `userName` (`appr`) who will provide the data, data property or metadata ( $q \in PD_{hdt}^p$ ), and signature  $\{H(q)\}_{K_R^{-1}}$ . The data under system,  $S$  can also be accessible in the same way which is not shown.

- **Step 1:** User  $R$  sends the `req` encrypted with the public key ( $K_D$ ) of DApp (D) and a nonce ( $N_1$ ) to the DApp (D) in congruous with  $M1$  in Table 6.8 over an `HTTPS` channel. The request can be made form the user interface as illustrated in Figure 6.15.
- **Step 2:** After decrypting the `req`, D forwards it to the data owner or approver (A) according to  $M2$  in Table 6.8.

FileName	Patient Stage	Patinet Data Type	Uploader Email	Upload Time	Action	
file1Body.json	Patient Disease Diagnose Data	Test Result Data	sadmansakibakash@gmail.com	Sun Jan 29 2023 23:07:18 GMT+0600 (Bangladesh Standard Time)	Show Data	Delete
exercise.js	Pre-Hospital Admit Data	Social Data	sadmansakibakash@gmail.com	Sat Jan 15 2022 06:00:00 GMT+0600 (Bangladesh Standard Time)	Show Data	Delete
FoodIntake.txt	Pre-Hospital Admit Data	Self-Reported Data	sadmansakibakash@gmail.com	Mon Jun 20 2022 06:00:00 GMT+0600 (Bangladesh Standard Time)	Show Data	Delete
SurgeryTeam.js	Surgical Operative Procedure	Previous Test Data	sadmansakibakash@gmail.com	Sat Dec 10 2022 06:00:00 GMT+0600 (Bangladesh Standard Time)	Show Data	Delete
file3File.json	Surgical Operative Procedure	Sequence of Surgery	sadmansakibakash@gmail.com	Sun Jan 29 2023 23:07:42 GMT+0600 (Bangladesh Standard Time)	Show Data	Delete

Figure 6.12: The Output of the Health State Query Request.

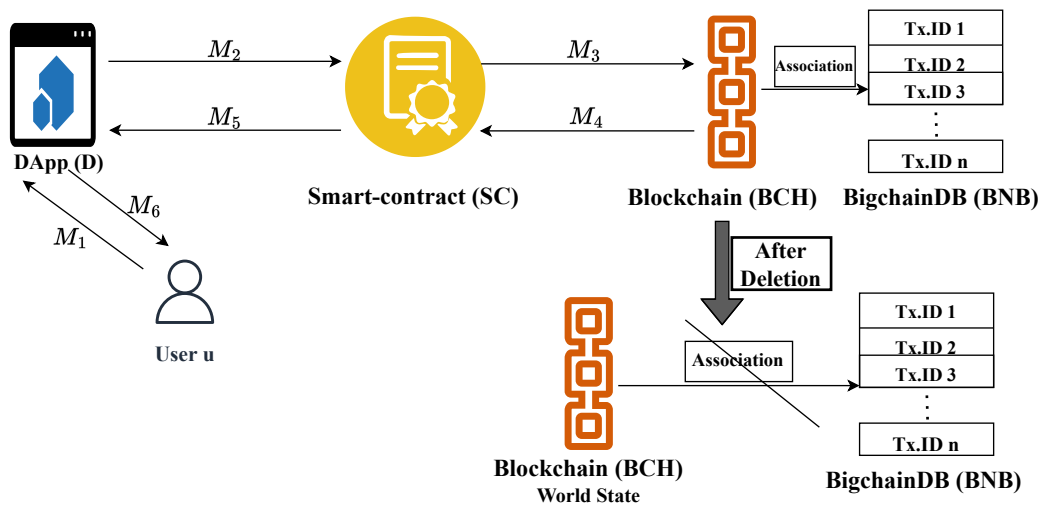


Figure 6.13: Delete Data Protocol.

- **Step 3:** After reaching the *req* to A, it can be dropped or gone forward with the consent of A. If, A wants to share the requested data, it sends the *req* to D encrypting it with D's public key according to  $M3$  in Table 6.8 and it is also illustrated in Figure 6.14. The request can be accepted or canceled from the user interface as illustrated in Figure 6.16.
- **Step 4:** After that, D relays the *req* to smart contracts (SC) ( $M4$  in Table 6.8), subsequently it is being handled by *shareRecordFunc* in Algorithm 2, starting with retrieving data from *shareData* (from Line 35 to Line 38).
- **Step 5:** Now, SC sends the approver's *userName* (*uapprover*) to blockchain (BCH) in accordance with  $M5$  in Table 6.8.
- **Step 6:** Then, SC retrieves the relevant data for A from BCH which is *ujson* in Line 39 ( $M6$  in Table 6.8).
- **Step 7:** The *TxID* of the *BigchainDB* for the mentioned data have been processed in Line 40. To record the share information, the necessary data is being aggregated as *shareRecordjson* and SC sends the data again to BCH



M1	$R \rightarrow D$ :	$[N_1, \{req\}_{K_D}]_{https}$
M2	$D \rightarrow A$ :	$[N_2, req, \{H(req)\}_{K_D^{-1}}]_{https}$
M3	$A \rightarrow D$ :	$[N_3, \{req\}_{K_D}]_{https}$
M4	$D \rightarrow SC$ :	$N_4, req$
M5	$SC \rightarrow BCH$ :	$N_5, uapprover$
M6	$BCH \rightarrow SC$ :	$N_5, ujson$
M7	$SC \rightarrow BCH$ :	$N_6, \{shareRecordjson\}$
M8	$BCH \rightarrow SC$ :	$N_6, TRUE$
M9	$SC \rightarrow D$ :	$N_4, TxID$
M10	$D \rightarrow BNB$ :	$N_7, TxID$
M11	$BNB \rightarrow D$ :	$N_7, \{ZZ_h^{p,t}\}_{K_A}$
M12	$D \rightarrow A$ :	$[N_3, resp', \{H(resp')\}_{K_D^{-1}}]_{https}$
M13	$A \rightarrow D$ :	$[N_2, resp, \{H(resp)\}_{K_A^{-1}}]_{https}$
M14	$D \rightarrow R$ :	$[N_1, resp, \{H(resp)\}_{K_D^{-1}}]_{https}$

Table 6.8: Share Data Protocol.

to create a transaction according to M7 in Table 6.8. The corresponding transaction is presented in Figure 6.17.

- **Step 8:** After the compilation of transaction, BCH returns a Boolean result in Line 42 (M8 in Table 6.8).
- **Step 9:** Then SC returns the  $TxID$  to D in Line 43 (M9 in Table 6.8).
- **Step 10:** Without changing anything, D sends the  $TxID$  number to *BigchainDB* (BNB) in congruous with (M10 in Table 6.8).
- **Step 11:** BNB retrieves the specified data  $\{ZZ_h^{p,t}\}_{K_A}$  which is encrypted by A's public key (depicted in Figure 6.14) and returns it to D in accordance with M11 in Table 6.8.
- **Step 12:** Now D stores the acquired data as *returnedData'* and provides a "Transaction Successful" response as  $response'_{req}$  (illustrated in Figure 6.14). After that, D sends the encapsulated  $resp'$  to the data owner signing the hash of the  $resp'$  over an *HTTPS* channel (M12 in Table 6.8).
- **Step 13:** A decrypts the *returnedData'* and re-encrypts ( $\{ZZ_h^{p,t}\}_{K_R}$ ) with the requester R's public key before sending it back to D with a signature of the new formed  $resp$ 's hash over an *HTTPS* channel in congruous with M13 in Table 6.8.
- **Step 14:** Lastly, the requested data is being sent to requester R by D over an *HTTPS* channel (M14 in Table 6.8). Now, R can decrypt the data with its private key ( $K_R^{-1}$ ) and can access it.



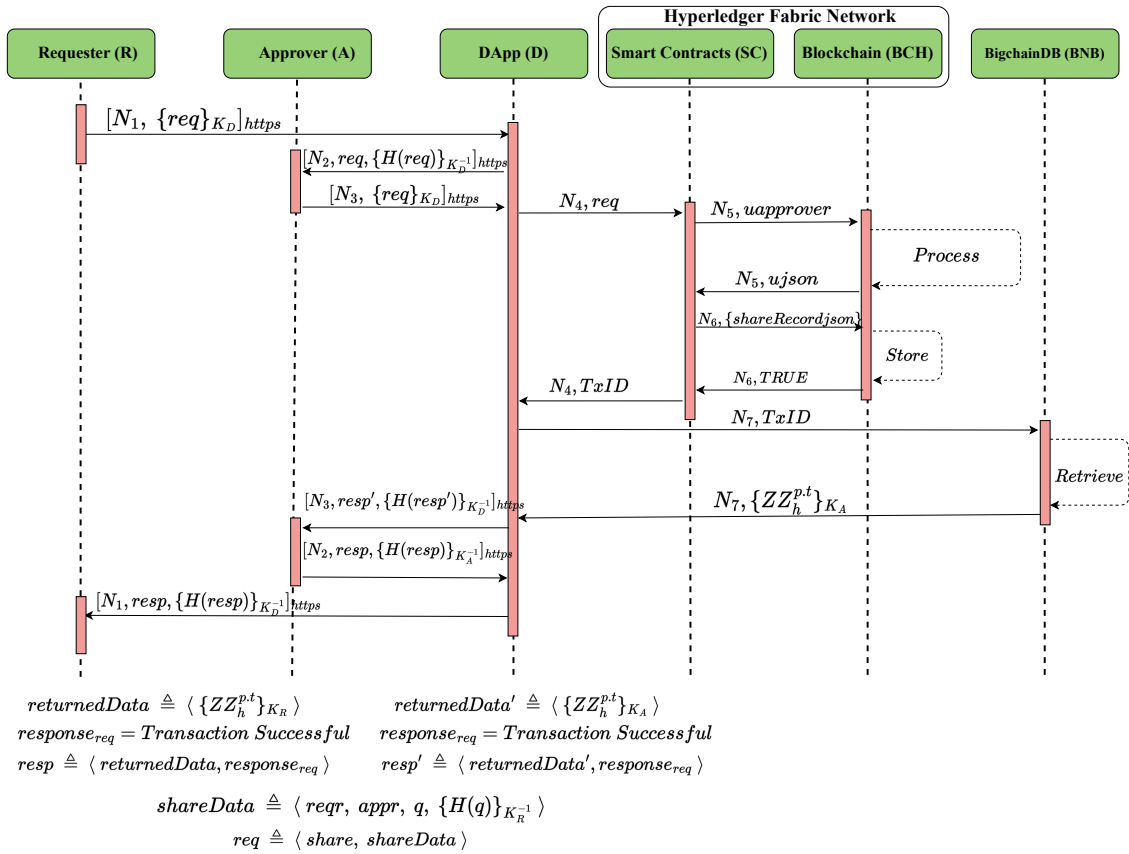


Figure 6.14: Share Data Flow.

Patient Name	File Name	Patient Stage	Patinet Data Type	Action
sadmansakibakash@gmail.com	file1Body.json	Patient Disease Diagnose Data	Test Result Data	<a href="#">Request Access</a>
sadmansakibakash@gmail.com	exercise.js	Pre-Hospital Admit Data	Social Data	<a href="#">Request Access</a>
sadmansakibakash@gmail.com	FoodIntake.txt	Pre-Hospital Admit Data	Self-Reported Data	<a href="#">Request Access</a>
sadmansakibakash@gmail.com	CaregiverList.pdf	Patient Disease Diagnose Data	Caregiver Data	<a href="#">Request Access</a>
sadmansakibakash@gmail.com	CaregiverList.pdf	Patient Disease Diagnose Data	Caregiver Data	<a href="#">Request Access</a>
sadmansakibakash@gmail.com	SurgeryTeam.js	Surgical Operative Procedure	Previous Test Data	<a href="#">Request Access</a>
sadmansakibakash@gmail.com	HealthStateQuery.png	Surgical Operative Procedure	Caregiver Data	<a href="#">Request Access</a>
sadmansakibakash@gmail.com	file3File.json	Surgical Operative Procedure	Sequence of Surgery	<a href="#">Request Access</a>

Figure 6.15: The User Interface for Share Data Request.

FileName	Requester	Patient Data Type	RequesterEmail	RequesterType	Action
FoodIntake.txt	Dr. Raihan Ahsan	Self-Reported Data	raihanahsan@gmail.com	Physician	<a href="#">Share</a> <a href="#">Cancel</a>

Figure 6.16: The User Interface for the action of Share Data Request.

```
{
  Key: 'fileShare_file_sadmansakibakash@gmail.com_236c57805d813d71326eb8d82373ff451c7c9399312c
m',
  DataOwner: 'sadmansakibakash@gmail.com',
  DataType: 'Self-Reported Data',
  PatientStage: 'Pre-Hospital Admit Data',
  RequesterEmail: 'raihanahsan@gmail.com',
  FileName: 'FoodIntake.txt',
  BigchainDBTransactionID: '49d1723ef97ea9ba474fc184845cd3d74ab1531e6a17e6d096f8a448e9f3447e',
  AESCipher: '11f81b6935bb9010b9985bdd625d09feff9252c8203e43e2d41a75f1f3acb798121bc908f2f915cb
128ddb3a80050c14c481dc01200754105e1232cb072272042c8517436c2e462c23407f8d566e7486602cb4ec3966c
e51e50da1cfc0b59681e6241d55619e02612253aaaf5035baa7cc76c02df54da73fadbc096e566bbcd3b3cec87ca08
4fa001ac6422c74deef6b1c66880f8d48195d97e2eb5b1d22daaff6d61e3904da8d9786bb33b6c4055d5489e77709d
DocType: 'fileShare'
}
```

Figure 6.17: A *Hyperledger Fabric* Transaction for Share Data Request.

# Chapter 7

## Discussion

### 7.1 Analysing Requirements

We have discussed mandatory requirements for the consistency among the proposed system's components in Section 4.4. Now we analyze how the proposed system *HDT* has satisfied the defined requirements.

#### 7.1.1 Functional Requirements

By correctly making a *hsquery* request (Section 6.2.3), *HDT* facilitates a user to gather all the necessary data to create a digital twin, consequently satisfying F1. The *share* request (Section 6.2.3) enables the sharing of personal data among entities and hence satisfies F2. By keeping the hash of each off-chain data as a transaction on the on-chain storage, F3 is satisfied. *HDT* is based on a private blockchain platform which processes all the transactions in an immutable manner which satisfies F4. During the *registration* request (Section 6.2.3), a *userName* attribute will be accepted by both the user and the system as a unique ID for that specific user. This satisfies F5.

#### 7.1.2 Security Requirements

According to the defined protocol flow of Section 6.2, a user needs to be registered and then authenticated to access the system. However, the administrator of the system will need to enforce adequate access control rules so that only authorized users can access a service or data. By doing these, S1 requirement can be satisfied.

All data between the user and the DApp, are transmitted over secure *HTTPS* channels which ensures the confidentiality of the data. Even though we propose to store data in an off-chain database, the hash of each data is stored in the blockchain, which ensures the integrity of the data. In addition, every request from the user is digitally signed to ensure authenticity. All these combinedly can satisfy S2. *HDT* is based on a private blockchain platform which does not exhibit any single point of failure, however, individual nodes can be the victim of a DoS attack rather than the full network [134]. Indeed, DoS attacks such as Transaction flooding and Spam requests can be tackled, as the system will be governed by a blockchain consensus algorithm where each transaction and user requests will go thorough a filtering

process. This can minimize the threat of a DoS attack. This can partially satisfy S3. However, to fully deter DoS attacks or reduce it to a minimum level, additional measures might be required. We propose to use nonces in every step of our protocol to guard against any replay attack, consequently satisfying S4. A rigorous access control mechanism can be used to monitor the resource consumption of every single user which in turn can satisfy S5.

### 7.1.3 Privacy Requirements

The requests that will generate transactions in the blockchain, e.g., *hdatawrite* or *share*, require a user to sign the transaction with their private key. Transactions without the signature will be considered as an invalid transaction and will not be recorded. This signing mechanism will imply the user's consent for any particular request and hence, satisfies P1. For a *share* request (Section 6.2.3), users do not need to give access to all the personal data. Users can define the range of data that need to be shared with the help of data property parameters of *share* request. As a result, it will fulfil P2.

## 7.2 Fulfilment of Research Objectives

Here, we present how our proposed model completes different research objectives (R.O.) described in Section 1.2.

1. According to R.O. 1, a mathematical data model needs to be developed for accumulating defined and structured clinical data. We have fulfilled this research objective by developing and presenting a mathematical data model in Section 4.2
2. R.O. 2 presents a problem: when a novel concept is deployed for empirical use there are some precautions for handling the uncertain mishaps. Section 4.3 and Section 4.4 respectively describe the threat modeling and requirement analysis for our proposed system which ameliorate this problem to some degree.
3. Before deploying a blockchain network for practical use, all the use cases and plausible stakeholders need to be estimated as blockchain is an autonomous system. R.O. 3 highlighted this important factor. The proposed system architecture in Chapter 5, has been delineated properly with all the system components which fulfils the objective.
4. Section 6.2 provides how the system components will communicate during user request with a proper protocol flow for the proposed system. This fulfils the R.O. 4.
5. Section 6.1 presents the experiment environment and tools. Moreover, Section 6.2 illustrates the protocol flow in harmony with the implementation where the *Hyperledger Fabric* and the *BigchainDB* have been used as on-chain and off-chain storage. So Chapter 6 fulfils the R.O. 5.
6. We have analysed requirements where we have elaborated how the system is consistent with the pre-defined requirements in Section 7.1. It fulfils the R.O. 6.

## 7.3 Comparison

Here we will compare our proposed model with other recent research works. The comparison criteria and the comparison are presented in Table 7.1. The notation “○” implies a research work is out of scope for corresponding criterion while the notation “✕” indicates the corresponding criterion is not needed for the research work. The notation “⊗” represents there is nothing explicitly described about the criterion for that corresponding research work. The notation “▶” means the criterion is necessary but not considered for the that research work. Moreover, we have used the notation “◐” to denote a research work has considered the criterion while developing the work but not provided any explicit information, whereas, the notation “●” represents that the research work has fully considered the corresponding criterion and has been implemented.

Only for the “Privacy” evaluation criterion, the access control of the systems, the encryption of private data, and the Security of the systems are considered as the sub-criteria respectively in the comparison Table 7.1.

A few observations can be made from Table 7.1:

- The research works mostly avoids any concern for user identity. Most of them stores the user identities in the same storage where the system data resides which is significantly prone to multiple threats. Anyone with the higher access control can stole or breach identity relevant data. Our propose *HDT* is based on blockchain technology where each user is provided with certificate and public-private key-pair and they are stored in local wallet. The presented identity facility during *registration* protocol in Section 6.2.3 ensures better security.
- From the perspective of user authentication, it can be perceived from Table 7.1 that most of the research works have used traditional authentication procedure without any mapping mechanism, whereas our proposed system has proper authentication method with the help of defined protocol flow in Section 6.2.
- For concocting a solid Digital Twin system for an instance, collecting data from physical space for the product is the most difficult endeavor. A defined and planned data model can conveniently collect data from the necessary sources. Our proposed system has its own concrete mathematical data model with a practical use case which has been elaborated in Section 4.2. Additionally, among the mentioned research works in Table 7.1, [112] and [22] have presented a few models for perceiving data in a predefined way, however proper structure and any elaboration of the use of it was missing.
- Data conversion technique (e.g. filter, cleaning, consistency checking, and so on) is a fundamental process before utilizing the data for knowledge generation, the input for Machine Learning algorithms or some other purposes. Some of the selected works have used data pre-processing techniques for achieving the required data. In our case, the data will collected according to the predefined data model. Assuming all the collected data is not required to go through any

data transformation techniques, we have not discussed anything in this regard for our proposed model.

- Data sharing is an important feature for a system specially when there are a myriad of stakeholders involved. From the perspective of data sharing, there are no solid information is mentioned in the research works from Table 7.1. With our presented share data protocol in Section 6.2.3, all the participant entities in our system can share their data and can stop sharing it also. Because of it, data can be shared among entities with proper integrity and confidentiality.
- Privacy is a prominent factor for a system to work efficiently. For the comparisons in Table 7.1, for evaluating privacy of a system we have considered three factors as mentioned before (access control, encryption, and security). All the mentioned research works have avoided or remotely motioned anything about privacy concern.
- Additionally, privacy is a prominent factor for handling data inside or outside the system environment. From the stated comparison in Table 7.1, we have considered access control, encryption, and security of data for evaluating privacy of a system. However, all the mentioned research works have not used all these three important criteria while considering privacy issue.
- Except for [22], all the mentioned research works in Table 7.1 have presented the fact that for research purposes previous data was used. For many reasons, previous data is very useful and it can be used for generating knowledge, isolating important features, and so on. This is why previous data has to be collected or extracted from previous systems or projects. Our proposed system *HDT* can extract data from external system with the means of system *S* as depicted in health state query from Section 6.2.3. Other research works have mentioned what dataset were used for the research purpose however did not mention how they accumulated it or is there any way to dynamically collect the data.

From the Table 7.1, we can come to the conclusion that our proposed system *HDT* can provide better security and privacy compare to other mentioned research work. Collecting data in a structured and predefined way help faster data control. Moreover, extracting data from other system with the consent of user is a unique feature and it can be evolved to exploit other services for the betterment of the system. These important distinctions of our proposed *HDT* model are the reasons which let our system to stand out among other state-of-the-art research works in the health-care sector from the scope of DT.

## 7.4 Synergy with HIPAA and GDPR

The developed countries of the world have their own regulations to control the distribution of patient data. Because health is more important than any monetary asset. There are some well known and established regulations and protocols for the safety and security of patient data. Health Insurance Portability and Accountability

Research Work	Criteria to compare									
	Identity	Authen-tication	Data Share	Storage System	Data model	Data Preprocessing	Privacy	Previous Data collection		
Cardio Twin [113]	○	○	⊗	⊗	⊗	Data fusion	○, ○, ○	EKG for recent patient update		
Hospital Buildings [111]	⊗	⊗	○	Data warehouse	BIM Technology	Booklet of interior data standards	○, ○, ●	Genotype Tissue Expression		
CloudDTH [112]	Cloud services	Cloud services	⊗	Cloud	5 types	Data cleaning and fusion	●, ⊗, ●	Patient Health Record		
Reducing HbA1c [114]	▶	▶	⊗	●	⊗	Data fusion	▶, ▶, ●	Vitals, Biothesiometry		
Graph [22]	○	○	○	*	4 types	*	○, ○, ○	●		
Diagnostics and Rehabilitation [115]	Anony-mous	▶	▶	PostgreSQL Relational Database	Patinet Information Model	Cleaning, consistency checking, linking	●, ⊗, ●	Patient and clinician personal record		
DT Clinical DSS [116]	⊗	⊗	○	⊗	⊗	*	○, ○, ○	Indian Liver Patient Dataset		
Intelligent Healthcare [6]	Not secure	▶	⊗	Cloud	Model-Building	Cleansed, prepro-cessed, transformed	▶, ▶, ▶	MIT-BIH Arrhythmia		
Agents and DT [117]	⊗	⊗	⊗	Cloud infras-tructure	⊗	⊗	▶, ▶, ●	DT data from its lifecycle		
Our proposed <i>HDT</i>	●	●	●	Private Block-chain	Mathematical model	○	●, ●, ●	Separate system, S		

**Symbol Legends:** Out of scope: ○, No need: \*, Not mentioned: ⊗, Needed but not available: ▶, Mentioned but not stated: ●, Applied: ●

Table 7.1: Comparison among some recent digital twin research works.

Act (*HIPAA*) [28] is one of the most prominent one. There are a lot of clauses that are described by *HIPAA*. But, our main concern will be with the clauses that are pertinent to patients' data. There is another regulation known as the General Data Protection Regulation (*GDPR*) [29] in European Union (EU). Here we will discuss some of the selective regulations relevant to patients' perspective from *HIPAA* and *GDPR*.

We will start with the similarities between *HIPAA* and *HDT*.

#### 7.4.1 Similarities between *HIPAA* and *HDT*:

- *HIPAA* has a clause explicitly mentioning that each participant patient should be identifiable by unique identity [135]. There should never be any confusion about identifying a patient. Each user in our proposed system *HDT*, has to go through registration procedure as described in Section 6.2.3. Each user will receive a unique *userName* and the corresponding *userType* will isolate a cluster of user. Moreover, system provided *certificate* will be used for identifying a user in different components. Moreover, not only user all the entities in the system can be uniquely identifiable.
- *HIPAA* clause 45 CFR 164.502 (a) states all systems in the field of healthcare should provide the opportunities for the physicians or doctors to get access of the patient data or can request to get the access. Our system provided *share* request can get this task done proficiently. Physicians can request patients to get access of the health data with the means of share data protocol defined in Section 6.2.3.
- According to *HIPAA* administrative rule 45 CFR 164.304, patient data created by other entities, e.g., physicians, sensors, testing devices, and so on, need to be recorded with integrity, confidentiality, and availability [136]. In our case, the *HDT* system is governed by private blockchain so repudiation can be easily evaded. Moreover, the data will be stored alongside with the hash value as a result data integrity can be ensured. Additionally, the patient encrypted raw data is stored off-chain so confidentiality breach can be avoided. Similar way, patient data is accessible without the consent or endorsement from the corresponding patient consequently the system is comparatively more secure in many ways.

#### 7.4.2 Dissimilarities between *HIPAA* and *HDT*:

- There are some special conditions when rules can be overridden. At these situations conventional rules are not applicable. *HIPAA* ACT 45 CFR 164.502 (a) (2) dictates that on certain situations, e.g., investigation, government ordinance, and so on, it is obligatory to disclose patients' private data to authoritative entities. Our proposed system does not follow or has no amenity for these special situations. The patient data can only be shared with other entities by the corresponding patients through share data protocol defined in Section 6.2.3.



### 7.4.3 Similarities between *GDPR* and *HDT*:

- *GDPR* Article 20 (1 (a, b), 2) dictates that patients have to have the full authority over the patient health data. *HDT* has the defined structure where the patients are the owners of their own data and share the data with other participants of the system.
- Article 16 and Article 17 of *GDPR* states that patients have the authority to update or delete their health data [137]. In our case, the system is designed primarily for the benefit of the patients, subsequently they have the amenities to delete and write data. The *delete* and *hdatawrite* protocol flows are designated to those tasks respectively which are elaborated in Section 6.2.3.
- *GDPR* Article 7 restricts the use of patient health data without the consent of corresponding patients. There are many scenarios where the system authorities use the system data for other purposes. But patient health data is a different case as it is considered private and very sensitive. *HDT* provides the patients full ownership and access control authority which coincided with this Article.
- Article 15 (1(b, c, g), 2, 3) of *GDPR* collectively states that patients have to be notified whenever the data is altered, shared, and disclosed [138]. The *HDT* system works according to the regulations mentioned in these Articles. The system is based on blockchain so every requests are being recorded as transactions with total consent of the initiators.

### 7.4.4 Dissimilarities between *GDPR* and *HDT*:

- According to *GDPR* Article 21, patients have the full authority to stop use of its' data. The *HDT* system does not render any opportunity to compliant with this regulation. There is no procedure or step the on-going use of patient data.
- Article 24, Article 26, and Article 28 of *GDPR* collectively states that a designated entity controller will control all the policies, data processing, requests, and so on, for a set of patients [139]. It means a personnel will overlook the requests and activities of a certain number of patients. On the contrary our proposed *HDT* is a autonomous system based on blockchain technology. The full system is governed by consensus algorithm, defined smart contracts, and policies. So this regulation does not match with our proposed system from the perspective of patients.

One of the crucial design goals of *HDT* was that no healthcare data would be stored on-chain, rather an off-chain database would be utilized for this purpose. The implication is that the healthcare data of a patient thus can be updated or removed if required to enforce a corresponding regulation (e.g. *GDPR*'s 'right to be forgotten') [31]. However, the immutability of a blockchain transaction will ensure that the request to read/update/remove is recorded in the blockchain, thus creating a secure audit trail. Interestingly, there are many on going researches to facilitate the delete and edit operations in blockchain [140]–[142].

## 7.5 Advantages and Disadvantages

Healthcare DT (*HDT*) provides some advantages as bellow:

- *HDT* can pull external data outside of the system. The data will be extracted and then it will be stored according to the described mathematical data model in Section 4.2. In this way, the accumulated data does not have to go through data conversion techniques, is an important feature of our proposed system.
- The proposed *HDT* system is dependent on a concrete mathematical data model. The mathematical data model ensures the collection of structured data from different dimensions. According to the proposed system, the data can be used as it is, for creating DT instance for a patient.
- The *HDT* system is volatile because of its adaptable nature. The system could be deployed nationwide and at first it could start with a set of hospitals, later on, more hospitals can register under our proposed system and mathematical data model will be dynamically changed to perceive more varieties of patient centric data.
- The *HDT* system does not primarily depend on the data available from the internal system. Based on requirement, the system can extract data from outside systems with the consent of the data owner, in our case which is patient. Because of this aggregation of data, the system will evolve with time and the inclusion of more entities, e.g., hospitals, stakeholder, and so on, will be facilitated.
- With the help of Share Data protocol, hospital or other stakeholders can share and stop sharing data in a consistent way. The defined sharing requests are being recorded on-chain which can stop repudiation and other threats.
- Storing all the raw data (e.g., file data, patient test data, physicians' prescriptions, and so on) in an off-chain database, in our case BigchainDB, mitigates the overburden of insurmountable data. It makes the system more fast and easy to iterate.
- A private blockchain based system gives the amenity to exploit all the features of blockchain which helps building a synchronized environment. Decentralization, transparency, and other benefits of using the blockchain technology, can be achieved, would have been more complicated other wise.

However, the current development has some disadvantages as presented below:

- All the patient relevant data has been stored in an off-chain database, in our case BigchainDB as according to *HIPPA* and *GDPR* patient should have the amenity to delete the data which opportunity blockchain technology does not provide. Because of this, all the data are being stored encrypted in BigchainDB with the owners' public key as a result the data is not accessible to other entities. However, the amenity of deleting data is not being properly accomplished.

- According to our proposed model, the users of *HDT* are not eligible to access the external patient data except for the authorized entities. After developing some proper access control mechanism and APIs, this problem can be solved. Future work
- There are some scenarios where patient is not in the condition to share their data beforehand, e.g., accident, severe health condition, and so on. However, this issue is not introduced in this model. Some other security threats can arise from this amenity if it is incorporated without due considerations.

## 7.6 Limitations

There are some limitations of the proposed system *HDT* which are presented as bellow:

- In our system, we have not discussed anything about data analysis. Forecasting, health state estimation, and so on, are some results of incorporating data analysis. With due consideration, it will be taken into account in our future research.
- Usually, after deployment of a successful DT system, a raft of data are generated which are later used for data analysis. To fathom the system generated data for *HDT*, there has to be some proper services and data analysis techniques for generating knowledge. However, it was not in the scope of our research incentives.
- There are some exceptional cases where the patient data has to be disclosed to some authoritative entity. In desperate situations e.g., death, government ordinance, emergency, and so on, the data has to be shared or disclosed which was not accounted for developing the *HDT* system.

## 7.7 Future Work

Now we will present some future plan regarding extending this research.

- The system performance is not analysed and not accounted as a scope of this research work. In future we will examine the empirical influence and performance.
- The *HDT* system can be used for simulation, forecasting, and estimation of health state. We will extend our research work later to investigate these mentioned issues. It would be very efficacious if a predictive model can be built on top of *HDT*. There are many other facets of DT that has to be accounted, we will consider those in our next research.
- *HIPAA* and *GDPR* are top healthcare regulations introduced by developed countries. As previously elaborated in Section 7.4, there are some dissimilarities between *HDT* and *HIPAA* as well as *GDPR*. In future, we will try to mitigate the mentioned dissimilarities.

- In order to deploy our system in a real-life setting, it would be important to understand the different perspectives of different users of the proposed system. Also, the presented mathematical data model in Section 4.2 needs to be evaluated according to the views of different users. To accomplish these objectives, we would like to have a focus group discussion with some hospital stakeholders, (e.g. hospitals, patients, physicians, nurses, relevant government organisations, and so on). In future we will conduct a user-study covering the hospital as well from the users' perspectives toward our proposed data model and will consider different privacy aspects of our proposed system.

# Chapter 8

## Conclusions

At present, many developments are going on in order to subside the uncertain health mishaps. Artificial Intelligence, Big data, and many more techniques are being used without any due consideration of how this vast and diverse data can be accumulated from the real world conveniently and store them securely. The digital twin technology can enable an effective way for collecting data and generating insight through analysis. But this data, being generated through numerous processes, needs to be systematically stored with proper security and handled by a compact system, which can also render all the requirements to create a digital twin in the healthcare sector. With these motivations in mind, our article presents a concrete mathematical model of Digital Twin for healthcare, proposes the Healthcare Digital Twin (*HDT*) system and provides the protocol flow for the system to coincide with the mathematical model.

The main contributions of this article are the following. The *HDT* is proposed with the incentive of remedying the segregated data collection process by incorporating a defined mathematical data model with which patient relevant data can be collected in a regulated way. The model has emphasized three core stages: Pre-Hospital Admit, Patient Disease Diagnose, and Surgical Operative Procedure, as these stages present the three most important stages for a patient. Next, the architecture of the system, being integrated with blockchain, is constructed with the defined data model in consideration, so that users can use the data for other purposes without any conflicts. With proper protocol flows, there are some illustrations of how the system can be used for different use cases.

It is understandable that, even with the state-of-the-art technologies, a digital twin of a full patient body is still out of reach because of the extant nuances in the human body. There are a raft of opportunities to decrease this gap. We strongly believe that the proposed model and system in this article will be a step towards fulfilling this goal. In future, we will develop the proposed system and examine its applicability and performance.

# Bibliography

- [1] G. Zhou, C. Zhang, Z. Li, K. Ding, and C. Wang, “Knowledge-driven digital twin manufacturing cell towards intelligent manufacturing,” *International Journal of Production Research*, vol. 58, no. 4, pp. 1034–1051, 2020.
- [2] M. Grieves and J. Vickers, “Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems,” in *Transdisciplinary perspectives on complex systems*, Springer, 2017, pp. 85–113.
- [3] M. W. Grieves, “Product lifecycle management: The new paradigm for enterprises,” *International Journal of Product Development*, vol. 2, no. 1-2, pp. 71–84, 2005.
- [4] S. Suhail, R. Hussain, R. Jurdak, and C. S. Hong, “Trustworthy digital twins in the industrial internet of things with blockchain,” *IEEE Internet Computing*, 2021.
- [5] F. Tao, F. Sui, A. Liu, *et al.*, “Digital twin-driven product design framework,” *International Journal of Production Research*, vol. 57, no. 12, pp. 3935–3953, 2019.
- [6] H. Elayan, M. Aloqaily, and M. Guizani, “Digital twin for intelligent context-aware iot healthcare systems,” *IEEE Internet of Things Journal*, vol. 8, no. 23, pp. 16 749–16 757, 2021.
- [7] S. D. Okegbile, J. Cai, C. Yi, and D. Niyato, “Human digital twin for personalized healthcare: Vision, architecture and future directions,” *IEEE Network*, 2022.
- [8] J. Zhang, L. Li, G. Lin, D. Fang, Y. Tai, and J. Huang, “Cyber resilience in healthcare digital twin on lung cancer,” *IEEE Access*, vol. 8, pp. 201 900–201 913, 2020.
- [9] A. M. Madni, D. Erwin, and C. C. Madni, “Digital twin-enabled mbse testbed for prototyping and evaluating aerospace systems: Lessons learned,” in *2021 IEEE Aerospace Conference (50100)*, IEEE, 2021, pp. 1–8.
- [10] L. Zhao, C. Wang, K. Zhao, D. Tarchi, S. Wan, and N. Kumar, “Interlink: A digital twin-assisted storage strategy for satellite-terrestrial networks,” *IEEE Transactions on Aerospace and Electronic Systems*, 2022.
- [11] F.-Y. Wang, Y. Li, W. Zhang, G. Bennett, and N. Chen, “Digital twin and parallel intelligence based on location and transportation: A vision for new synergy between the ieeecrfid and itss in cyberphysical social systems [society news],” *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 1, pp. 249–252, 2021.

- [12] J. Han, Q. Hong, M. H. Syed, *et al.*, “Cloud-edge hosted digital twins for coordinated control of distributed energy resources,” *IEEE Transactions on Cloud Computing*, 2022.
- [13] G. Xie, K. Yang, C. Xu, R. Li, and S. Hu, “Digital twinning based adaptive development environment for automotive cyber-physical systems,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1387–1396, 2021.
- [14] S. Almeaibed, S. Al-Rubaye, A. Tsourdos, and N. P. Avdelidis, “Digital twin analysis to promote safety and security in autonomous vehicles,” *IEEE Communications Standards Magazine*, vol. 5, no. 1, pp. 40–46, 2021.
- [15] F. Alshehri and G. Muhammad, “A comprehensive survey of the internet of things (iot) and ai-based smart healthcare,” *IEEE Access*, vol. 9, pp. 3660–3678, 2020.
- [16] A. N. Navaz, M. A. Serhani, H. T. El Kassabi, N. Al-Qirim, and H. Ismail, “Trends, technologies, and key challenges in smart and connected healthcare,” *Ieee Access*, vol. 9, pp. 74 044–74 067, 2021.
- [17] U. Bharti, D. Bajaj, H. Batra, S. Lalit, S. Lalit, and A. Gangwani, “Medbot: Conversational artificial intelligence powered chatbot for delivering telehealth after covid-19,” in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, IEEE, 2020, pp. 870–875.
- [18] M. A. Bazel, F. Mohammed, and M. Ahmed, “Blockchain technology in healthcare big data management: Benefits, applications and challenges,” in *2021 1st International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, IEEE, 2021, pp. 1–8.
- [19] J. Stauffer and Q. Zhang, “S 2 cloud: A novel cloud system for mobile health big data management,” in *2021 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (Green-Com) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, IEEE, 2021, pp. 380–383.
- [20] O. C. Madubuike and C. J. Anumba, “Digital twin-based health care facilities management,” *Journal of Computing in Civil Engineering*, vol. 37, no. 2, p. 04 022 057, 2023.
- [21] C. Patrone, G. Galli, and R. Revetria, “A state of the art of digital twin and simulation supported by data mining in the healthcare sector,” *Advancing Technology Industrialization Through Intelligent Software Methodologies, Tools and Techniques*, pp. 605–615, 2019.
- [22] P. Barbiero, R. Viñas Torné, and P. Lió, “Graph representation forecasting of patient’s medical conditions: Toward a digital twin,” *Frontiers in genetics*, vol. 12, p. 652 907, 2021.
- [23] S. Huang, G. Wang, Y. Yan, and X. Fang, “Blockchain-based data management for digital twin of product,” *Journal of Manufacturing Systems*, vol. 54, pp. 361–371, 2020.
- [24] Y. Lu, B. Ai, Z. Zhong, and Y. Zhang, “Energy-efficient task transfer in wireless computing power networks,” *IEEE Internet of Things Journal*, 2022.

- [25] F. Mokhtari and A. Imanpour, “A digital twin-based framework for multi-element seismic hybrid simulation of structures,” *Mechanical Systems and Signal Processing*, vol. 186, p. 109 909, 2023.
- [26] H. Wang, L. Lv, X. Li, *et al.*, “A safety management approach for industry 5.0 s human-centered manufacturing based on digital twin,” *Journal of Manufacturing Systems*, vol. 66, pp. 1–12, 2023.
- [27] X. Jia, M. Luo, H. Wang, J. Shen, and D. He, “A blockchain-assisted privacy-aware authentication scheme for internet of medical things,” *IEEE Internet of Things Journal*, 2022.
- [28] HIPAA, *Health information privacy*, Accessed on: 29 November. [Online]. Available: <https://www.hhs.gov/hipaa/index.html>.
- [29] GDPR, *General data protection regulation*, Accessed on: 29 November. [Online]. Available: <https://gdpr-info.eu/>.
- [30] P. Chaudhari, C. Gangane, and A. Lahe, “Digital twin in industry 4.0 a real-time virtual replica of objects improves digital health monitoring system,” in *International Conference on Information Systems and Management Science*, Springer, 2023, pp. 506–517.
- [31] S. S. Akash and M. S. Ferdous, “A blockchain based system for healthcare digital twin,” *IEEE Access*, vol. 10, pp. 50 523–50 547, 2022. DOI: [10.1109/ACCESS.2022.3173617](https://doi.org/10.1109/ACCESS.2022.3173617).
- [32] M. Schluse, M. Priggemeyer, L. Atorf, and J. Rossmann, “Experimentable digital twins—streamlining simulation-based systems engineering for industry 4.0,” *IEEE Transactions on industrial informatics*, vol. 14, no. 4, pp. 1722–1731, 2018.
- [33] K. Ding and P. Jiang, “Rfid-based production data analysis in an iot-enabled smart job-shop,” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 128–138, 2017.
- [34] A. Karakra, F. Fontanili, E. Lamine, and J. Lamothe, “Hospit’win: A predictive simulation-based digital twin for patients pathways in hospital,” in *2019 IEEE EMBS international conference on biomedical & health informatics (BHI)*, IEEE, 2019, pp. 1–4.
- [35] K. Y. H. Lim, P. Zheng, and C.-H. Chen, “A state-of-the-art survey of digital twin: Techniques, engineering product lifecycle management and business innovation perspectives,” *Journal of Intelligent Manufacturing*, vol. 31, no. 6, pp. 1313–1337, 2020.
- [36] D. Kiritsis, A. Bufardi, and P. Xirouchakis, “Research issues on product lifecycle management and information tracking using smart embedded systems,” *Advanced engineering informatics*, vol. 17, no. 3-4, pp. 189–202, 2003.
- [37] R. Portela, C. Varsakelis, A. Richelle, *et al.*, “When is an in silico representation a digital twin? a biopharmaceutical industry approach to the digital twin concept,” *Digital Twins*, pp. 35–55, 2020.
- [38] A. Matsokis and D. Kiritsis, “An ontology-based approach for product lifecycle management,” *Computers in industry*, vol. 61, no. 8, pp. 787–797, 2010.



- [39] M. J. M. Chowdhury, M. S. Ferdous, K. Biswas, N. Chowdhury, and V. Muthukkumarasamy, “A survey on blockchain-based platforms for iot use-cases,” *The Knowledge Engineering Review*, vol. 35, 2020.
- [40] C. M. Ezhilarasu, Z. Skaf, and I. K. Jennions, “Understanding the role of a digital twin in integrated vehicle health management (ivhm),” in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, IEEE, 2019, pp. 1484–1491.
- [41] S. Ivanov, K. Nikolskaya, G. Radchenko, L. Sokolinsky, and M. Zymbler, “Digital twin of city: Concept overview,” in *2020 Global Smart Industry Conference (GloSIC)*, IEEE, 2020, pp. 178–186.
- [42] E. D’Auria, M. Abrahams, G. V. Zuccotti, and C. Venter, “Personalized nutrition approach in food allergy: Is it prime time yet?” *Nutrients*, vol. 11, no. 2, p. 359, 2019.
- [43] P. Jouan and P. Hallot, “Digital twin: Research framework to support preventive conservation policies,” *ISPRS International Journal of Geo-Information*, vol. 9, no. 4, p. 228, 2020.
- [44] K. Židek, J. Pitel’, M. Adámek, P. Lazorik, and A. Hošovský, “Digital twin of experimental smart manufacturing assembly system for industry 4.0 concept,” *Sustainability*, vol. 12, no. 9, p. 3658, 2020.
- [45] S. Tiwari, N. Dhanda, and H. Dev, “A real time secured medical management system based on blockchain and internet of things,” *Measurement: Sensors*, p. 100 630, 2022.
- [46] S. Gupta, H. K. Sharma, and M. Kapoor, “Introduction to blockchain and its application in smart healthcare system,” in *Blockchain for Secure Healthcare Using Internet of Medical Things (IoMT)*, Springer, 2023, pp. 55–65.
- [47] M. B. Hossain, S. S. Akash, and Haque, “Coinbd: An enhanced version of proof of work with less computational power,” Ph.D. dissertation, BRAC University, 2018.
- [48] L. Bian, R. Xiao, Y. Lu, and Z. Luo, “Construction and design of food traceability based on blockchain technology applying in the metaverse,” in *International Conference on Smart Multimedia*, Springer, 2022, pp. 294–305.
- [49] M. V. B. Reddy, R. Kumar, A. Bag, A. A. Hagar, G. Vaithecswaran, and V. Tripath, “The multi layer security network authentication system development through blockchain technology,” in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, IEEE, 2022, pp. 259–264.
- [50] L. Lu, Z. Wen, Y. Yuan, *et al.*, “Iquery: A trustworthy and scalable blockchain analytics platform,” *IEEE Transactions on Dependable and Secure Computing*, 2022.
- [51] S. Siddiqui, S. Hameed, S. A. Shah, A. K. Khan, and A. Aneiba, “Smart contract-based security architecture for collaborative services in municipal smart cities,” *Journal of Systems Architecture*, p. 102 802, 2022.
- [52] M. J. M. Chowdhury, M. S. Ferdous, K. Biswas, *et al.*, “A comparative analysis of distributed ledger technology platforms,” *IEEE Access*, vol. 7, pp. 167 930–167 943, 2019.

- [53] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, “A review on consensus algorithm of blockchain,” in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, IEEE, 2017, pp. 2567–2572.
- [54] M. S. Ferdous, M. J. M. Chowdhury, M. A. Hoque, and A. Colman, “Blockchain consensus algorithms: A survey,” *arXiv preprint arXiv:2001.07091*, 2020.
- [55] D. Li, R. Chen, Q. Wan, *et al.*, “Decentralized iot resource monitoring and scheduling framework based on blockchain,” *IEEE Internet of Things Journal*, 2022.
- [56] Z. Alhadhrami, S. Alghfeli, M. Alghfeli, J. A. Abedlla, and K. Shuaib, “Introducing blockchains for healthcare,” in *2017 international conference on electrical and computing technologies and applications (ICECTA)*, IEEE, 2017, pp. 1–4.
- [57] C. Lin, X. Huang, J. Ning, and D. He, “Aca: Anonymous, confidential and auditable transaction systems for blockchain,” *IEEE Transactions on Dependable and Secure Computing*, 2022.
- [58] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, “On the security and performance of proof of work blockchains,” in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 3–16.
- [59] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, “Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals, applications and opportunities,” *IEEE Access*, vol. 7, pp. 85 727–85 745, 2019.
- [60] H. Mao, T. Nie, H. Sun, D. Shen, and G. Yu, “A survey on cross-chain technology: Challenges, development, and prospect,” *IEEE Access*, 2022.
- [61] D. Kester, T. Li, and Z. Erkin, “Pride: A privacy-preserving decentralised key management system,” in *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, 2022, pp. 1–6.
- [62] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An overview of blockchain technology: Architecture, consensus, and future trends,” in *2017 IEEE international congress on big data (BigData congress)*, Ieee, 2017, pp. 557–564.
- [63] B. Zhang, H. Cui, Y. Chen, X. Liu, Z. Yu, and B. Guo, “Enabling secure deduplication in encrypted decentralized storage,” in *International Conference on Network and System Security*, Springer, 2022, pp. 459–475.
- [64] R. Yang, R. Wakefield, S. Lyu, *et al.*, “Public and private blockchain in construction business process and information integration,” *Automation in construction*, vol. 118, p. 103 276, 2020.
- [65] *Bitcoin: A peer-to-peer electronic cash system*, Accessed on: 03 January 2023. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>.
- [66] *Litecoin whitepaper*, Accessed on: 03 January 2023. [Online]. Available: <https://www.allcryptowhitepapers.com/litecoin-whitepaper/>.
- [67] *A next-generation smart contract and decentralized application platform*, Accessed on: 03 January 2023. [Online]. Available: <https://ethereum.org/en/whitepaper/>.

- [68] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, “Medrec: Using blockchain for medical data access and permission management,” in *2016 2nd international conference on open and big data (OBD)*, IEEE, 2016, pp. 25–30.
- [69] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, “Fhir-chain: Applying blockchain to securely and scalably share clinical data,” *Computational and structural biotechnology journal*, vol. 16, pp. 267–278, 2018.
- [70] Y. Hao, Y. Li, X. Dong, L. Fang, and P. Chen, “Performance analysis of consensus algorithm in private blockchain,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, pp. 280–285.
- [71] S. Pongnumkul, C. Siripanpornchana, and S. Thajchayapong, “Performance analysis of private blockchain platforms in varying workloads,” in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, IEEE, 2017, pp. 1–6.
- [72] I. Alom, R. M. Eshita, A. I. Harun, *et al.*, “Dynamic management of identity federations using blockchain,” in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, IEEE, 2021, pp. 1–9.
- [73] C. Mohan, “State of public and private blockchains: Myths and reality,” in *Proceedings of the 2019 international conference on management of data*, 2019, pp. 404–411.
- [74] A. Ismailisufi, T. Popović, N. Gligorić, S. Radonjic, and S. Šandi, “A private blockchain implementation using multichain open source platform,” in *2020 24th International Conference on Information Technology (IT)*, IEEE, 2020, pp. 1–4.
- [75] E. Androulaki, A. Barger, V. Bortnikov, *et al.*, “Hyperledger fabric: A distributed operating system for permissioned blockchains,” in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.
- [76] *Sawtooth*, Accessed on: 03 January 2023. [Online]. Available: [https://www.hyperledger.org/wp-content/uploads/2018/01/Hyperledger\\_Sawtooth\\_WhitePaper.pdf](https://www.hyperledger.org/wp-content/uploads/2018/01/Hyperledger_Sawtooth_WhitePaper.pdf).
- [77] *Corda*, Accessed on: 03 January 2023. [Online]. Available: <https://www.r3.com/reports/corda-technical-whitepaper/>.
- [78] J. Xu, K. Xue, S. Li, *et al.*, “Healthchain: A blockchain-based privacy preserving scheme for large-scale health data,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8770–8781, 2019.
- [79] T.-T. Kuo and L. Ohno-Machado, “Modelchain: Decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks,” *arXiv preprint arXiv:1802.01746*, 2018.
- [80] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, “Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology,” *Sustainable cities and society*, vol. 39, pp. 283–297, 2018.
- [81] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, “Med-share: Trust-less medical data sharing among cloud service providers via blockchain,” *IEEE Access*, vol. 5, pp. 14 757–14 767, 2017.

- [82] N. Szabo, “Formalizing and securing relationships on public networks,” *First monday*, 1997.
- [83] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, *et al.*, “Formal verification of smart contracts: Short paper,” in *Proceedings of the 2016 ACM workshop on programming languages and analysis for security*, 2016, pp. 91–96.
- [84] K. Christidis and M. Devetsikiotis, “Blockchains and smart contracts for the internet of things,” *Ieee Access*, vol. 4, pp. 2292–2303, 2016.
- [85] Z. Zheng, S. Xie, H.-N. Dai, *et al.*, “An overview on smart contracts: Challenges, advances and platforms,” *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2020.
- [86] D. Siegel, “Understanding the dao attack,” *Retrieved June*, vol. 13, p. 2018, 2016.
- [87] Q. Nasir, I. A. Qasse, M. Abu Talib, and A. B. Nassif, “Performance analysis of hyperledger fabric platforms,” *Security and Communication Networks*, vol. 2018, 2018.
- [88] R. Alotaibi, M. Alassafi, M. S. I. Bhuiyan, R. S. Raju, and M. S. Ferdous, “A reinforcement-learning-based model for resilient load balancing in hyperledger fabric,” *Processes*, vol. 10, no. 11, p. 2390, 2022.
- [89] I. Alom, M. S. Ferdous, and M. J. M. Chowdhury, “Blockmeter: An application agnostic performance measurement framework for private blockchain platforms,” *arXiv preprint arXiv:2202.05629*, 2022.
- [90] W.-S. Lee, A. John, H.-C. Hsu, and P.-A. Hsiung, “Spchain: A smart and private blockchain-enabled framework for combining gdpr-compliant digital assets management with ai models,” *IEEE Access*, 2022.
- [91] N. Andola, M. Gogoi, S. Venkatesan, S. Verma, *et al.*, “Vulnerabilities on hyperledger fabric,” *Pervasive and Mobile Computing*, vol. 59, p. 101 050, 2019.
- [92] H. Javaid, C. Hu, and G. Brebner, “Optimizing validation phase of hyperledger fabric,” in *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, IEEE, 2019, pp. 269–275.
- [93] A. Sharma, F. M. Schuhknecht, D. Agrawal, and J. Dittrich, “Blurring the lines between blockchains and database systems: The case of hyperledger fabric,” in *Proceedings of the 2019 International Conference on Management of Data*, 2019, pp. 105–122.
- [94] A. Dabholkar and V. Saraswat, “Ripping the fabric: Attacks and mitigations on hyperledger fabric,” in *International Conference on Applications and Techniques in Information Security*, Springer, 2019, pp. 300–311.
- [95] H. Sukhwani, N. Wang, K. S. Trivedi, and A. Rindos, “Performance modeling of hyperledger fabric (permissioned blockchain network),” in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, IEEE, 2018, pp. 1–8.

- [96] L. Jiang, X. Chang, Y. Liu, J. Mišić, and V. B. Mišić, “Performance analysis of hyperledger fabric platform: A hierarchical model approach,” *Peer-to-Peer Networking and Applications*, vol. 13, no. 3, pp. 1014–1025, 2020.
- [97] H. Sukhwani, J. M. Martmez, X. Chang, K. S. Trivedi, and A. Rindos, “Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric),” in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, IEEE, 2017, pp. 253–255.
- [98] F. Benhamouda, S. Halevi, and T. Halevi, “Supporting private data on hyperledger fabric with secure multiparty computation,” *IBM Journal of Research and Development*, vol. 63, no. 2/3, pp. 3–1, 2019.
- [99] H. Mukne, P. Pai, S. Raut, and D. Ambawade, “Land record management using hyperledger fabric and ipfs,” in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IEEE, 2019, pp. 1–8.
- [100] J. Sousa, A. Bessani, and M. Vukolic, “A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform,” in *2018 48th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, IEEE, 2018, pp. 51–58.
- [101] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, “Performance analysis of a hyperledger fabric blockchain framework: Throughput, latency and scalability,” in *2019 IEEE international conference on blockchain (Blockchain)*, IEEE, 2019, pp. 536–540.
- [102] L. Foschini, A. Gavagna, G. Martuscelli, and R. Montanari, “Hyperledger fabric blockchain: Chaincode performance analysis,” in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, IEEE, 2020, pp. 1–6.
- [103] C. Wang and X. Chu, “Performance characterization and bottleneck analysis of hyperledger fabric,” in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, 2020, pp. 1281–1286.
- [104] M. Q. Nguyen, D. Loghin, and T. T. A. Dinh, “Understanding the scalability of hyperledger fabric,” *arXiv preprint arXiv:2107.09886*, 2021.
- [105] H. Honar Pajoooh, M. Rashid, F. Alam, and S. Demidenko, “Hyperledger fabric blockchain for securing the edge internet of things,” *Sensors*, vol. 21, no. 2, p. 359, 2021.
- [106] Q. Luo, R. Liao, J. Li, X. Ye, and S. Chen, “Blockchain enabled credibility applications: Extant issues, frameworks and cases,” *IEEE Access*, vol. 10, pp. 45 759–45 771, 2022.
- [107] A. Alnuaimi, A. Alshehhi, K. Salah, R. Jayaraman, I. A. Omar, and A. Battah, “Blockchain-based processing of health insurance claims for prescription drugs,” *IEEE Access*, vol. 10, pp. 118 093–118 107, 2022.
- [108] *Bigchaindb 2.0 the blockchain database*, Accessed on: 10 May 2022. [Online]. Available: <https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf>.

- [109] I. G. B. B. Nugraha *et al.*, “A blockchain-based traceability system to achieve the quality objectives in the production process of a manufacturing industry,” in *2022 International Conference on Information Technology Systems and Innovation (ICITSI)*, IEEE, 2022, pp. 194–199.
- [110] F. Tian, “A supply chain traceability system for food safety based on haccp, blockchain & internet of things,” in *2017 International conference on service systems and service management*, IEEE, 2017, pp. 1–6.
- [111] Y. Peng, M. Zhang, F. Yu, J. Xu, and S. Gao, “Digital twin hospital buildings: An exemplary case study through continuous lifecycle integration,” *Advances in Civil Engineering*, vol. 2020, 2020.
- [112] Y. Liu, L. Zhang, Y. Yang, *et al.*, “A novel cloud-based framework for the elderly healthcare services using digital twin,” *IEEE Access*, vol. 7, pp. 49 088–49 101, 2019.
- [113] R. Martinez-Velazquez, R. Gamez, and A. El Saddik, “Cardio twin: A digital twin of the human heart running on the edge,” in *2019 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, IEEE, 2019, pp. 1–6.
- [114] P. Shamanna, B. Saboo, S. Damodharan, *et al.*, “Reducing hba1c in type 2 diabetes using digital twin technology-enabled precision nutrition: A retrospective analysis,” *Diabetes Therapy*, vol. 11, no. 11, pp. 2703–2714, 2020.
- [115] D. Petrova-Antonova, I. Spasov, I. Krasteva, I. Manova, and S. Ilieva, “A digital twin platform for diagnostics and rehabilitation of multiple sclerosis,” in *Computational Science and Its Applications – ICCSA 2020*, O. Gervasi, B. Murgante, S. Misra, *et al.*, Eds., Cham: Springer International Publishing, 2020, pp. 503–518, ISBN: 978-3-030-58799-4.
- [116] D. J. Rao and S. Mane, “Digital twin approach to clinical dss with explainable ai,” *arXiv preprint arXiv:1910.13520*, 2019.
- [117] A. Croatti, M. Gabellini, S. Montagna, and A. Ricci, “On the integration of agents and digital twins in healthcare,” *Journal of Medical Systems*, vol. 44, no. 9, pp. 1–8, 2020.
- [118] D. Chambers, P. Wilson, C. Thompson, and M. Harden, “Social network analysis in healthcare settings: A systematic scoping review,” 2012.
- [119] S. Hasavari and Y. T. Song, “A secure and scalable data source for emergency medical care using blockchain technology,” in *2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA)*, IEEE, 2019, pp. 71–75.
- [120] P. Pasricha, B. Singh, and P. Verma, “Ethical leadership, organic organizational cultures and corporate social responsibility: An empirical study in social enterprises,” *Journal of Business Ethics*, vol. 151, no. 4, pp. 941–958, 2018.
- [121] I. Azimi, A. Anzanpour, A. M. Rahmani, P. Liljeberg, and H. Tenhunen, “Self-aware early warning score system for iot-based personalized healthcare,” in *eHealth 360°*, Springer, 2017, pp. 49–55.

- [122] V. Patterson, S. Samant, M. B. Singh, P. Jain, V. Agavane, and Y. Jain, "Diagnosis of epileptic seizures by community health workers using a mobile app: A comparison with physicians and a neurologist," *Seizure*, vol. 55, pp. 4–8, 2018.
- [123] F. Guo, Y. Mai, X. Zhao, *et al.*, "Yanbao: A mobile app using the measurement of clinical parameters for glaucoma screening," *IEEE Access*, vol. 6, pp. 77 414–77 428, 2018. DOI: [10.1109/ACCESS.2018.2882946](https://doi.org/10.1109/ACCESS.2018.2882946).
- [124] M. A. Cassera, B. Zheng, D. V. Martinec, C. M. Dunst, and L. L. Swanström, "Surgical time independently affected by surgical team size," *The American journal of surgery*, vol. 198, no. 2, pp. 216–222, 2009.
- [125] J. Wang, J. Cabrera, K.-L. Tsui, H. Guo, M. Bakker, and J. B. Kostis, "Predicting surgery duration from a new perspective: Evaluation from a database on thoracic surgery," *arXiv preprint arXiv:1712.07809*, 2017.
- [126] A. Wheelock, A. Suliman, R. Wharton, *et al.*, "The impact of operating room distractions on stress, workload, and teamwork," *Annals of surgery*, vol. 261, no. 6, pp. 1079–1084, 2015.
- [127] C. Zhuang, J. Liu, and H. Xiong, "Digital twin-based smart production management and control framework for the complex product assembly shop-floor," *The International Journal of Advanced Manufacturing Technology*, vol. 96, no. 1, pp. 1149–1163, 2018.
- [128] A. Shostack, *Threat modeling: Designing for security*. John Wiley & Sons, 2014.
- [129] R. Dennis, G. Owenson, and B. Aziz, "A temporal blockchain: A formal analysis," in *2016 International Conference on Collaboration Technologies and Systems (CTS)*, IEEE, 2016, pp. 430–437.
- [130] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, "Pobt: A lightweight consensus algorithm for scalable iot business blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2343–2355, 2019.
- [131] B. D. T. Consortium *et al.*, "Toward a universal biomedical data translator," *Clinical and translational science*, vol. 12, no. 2, p. 86, 2019.
- [132] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, "Gdpr-compliant personal data management: A blockchain-based solution," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1746–1761, 2019.
- [133] S. Tilkov and S. Vinoski, "Node.js: Using javascript to build high-performance network programs," *IEEE Internet Computing*, vol. 14, no. 6, pp. 80–83, 2010.
- [134] Z. Abou El Houda, A. S. Hafid, and L. Khoukhi, "Cochain-sc: An intra-and inter-domain ddos mitigation scheme based on blockchain using sdn and smart contract," *IEEE Access*, vol. 7, pp. 98 893–98 907, 2019.
- [135] D. M. De Simone, "When is accessing medical records a hipaa breach?" *Journal of Nursing Regulation*, vol. 10, no. 3, pp. 34–36, 2019.
- [136] S. U. Gardazi and A. A. Shahid, "Compliance-driven architecture for health-care industry," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 5, pp. 568–577, 2017.

- [137] R. N. Zaeem and K. S. Barber, “The effect of the gdpr on privacy policies: Recent progress and future promise,” *ACM Transactions on Management Information Systems (TMIS)*, vol. 12, no. 1, pp. 1–20, 2020.
- [138] L. Bufalieri, M. La Morgia, A. Mei, and J. Stefa, “Gdpr: When the right to access personal data becomes a threat,” in *2020 IEEE International Conference on Web Services (ICWS)*, IEEE, 2020, pp. 75–83.
- [139] A. Mahindrakar and K. P. Joshi, “Automating gdpr compliance using policy integrated blockchain,” in *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, IEEE, 2020, pp. 86–93.
- [140] R. Herian, “Blockchain, gdpr, and fantasies of data sovereignty,” *Law, Innovation and Technology*, vol. 12, no. 1, pp. 156–174, 2020.
- [141] W.-C. Huang, L.-Y. Yeh, and J.-L. Huang, “A monitorable peer-to-peer file sharing mechanism,” in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, IEEE, 2019, pp. 1–4.
- [142] N. Al-Zaben, M. M. H. Onik, J. Yang, N.-Y. Lee, and C.-S. Kim, “General data protection regulation complied blockchain architecture for personally identifiable information management,” in *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, IEEE, 2018, pp. 77–82.