IEEE Access and Transaction on Blockchain,Volume:10,Issue Date:09.May.2022

A Blockchain Based System for Healthcare Digital Twin

SADMAN SAKIB AKASH[1] **AND MD SADEK FERDOUS**[1,2]**, (Member, IEEE)**
[1]Department of Computer Science and Engineering, BRAC University, Dhaka 1212, Bangladesh
[2]Imperial College Business School, Imperial College London, London SW7 2AZ, U.K.

Corresponding author: Md Sadek Ferdous (sadek.ferdous@bracu.ac.bd)

ABSTRACT Digital Twin (DT) is an emerging technology that replicates any physical phenomenon from a physical space to a digital space in congruence with the physical state. However, devising a Healthcare DT model for patient care is seen as a challenging task as the lack of adequate data collection structure. There are also security and privacy concerns as healthcare data is very sensitive and can be used in malicious ways. Because of these current research gaps, the proper way of acquiring the structured data and managing them in a secure way is very important. In this article, we present a mathematical 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 our proposed Healthcare Digital Twin (*HDT*) system based on Blockchain. Accordingly, the proposed system is described with all the key components as well as with detailed protocol flows and an analysis of its different aspects. Finally, the feasibility of the proposed model with a critical comparison with other relevant research works have been provided.

INDEX TERMS Digital twin (DT), healthcare, mathematical model, blockchain.

I. INTRODUCTION

According to the latest statistics from the World Health Organization, about 930 million people worldwide are at risk of falling into poverty due to out-of-pocket health spending of 10% or more of their household budget [1]. Currently, there is a surge in improving the healthcare situation and a myriad of developments are ongoing in the healthcare sector with respect to Artificial Intelligence [2]–[4], Big data [5], [6], and in other spectrum. Though, it is not something that can be assuaged outright, whereas the real problem is not in the slow advancement of the technology, rather the mishaps in real life, e.g., adverse events, late diagnosis, etc [7]. DT can bring an immediate alteration in the healthcare sector from its root by incorporating analysis, predictive measurements, decision making paradigm, and data collection [8].

There are some notable developments in the healthcare sector incorporating DT. Martinez-Velazquez *et al.* [9] have developed a cardio twin based on the heart that can mitigate the risk of any Ischemic heart disease. Barbiero *et al.* [10] have proposed a general framework to provide a panoramic view over current and future physiological conditions. However, the recent developments in DT for the healthcare sector, have some drawbacks from the perspective of data sharing, storage, and access control [11]. Also, without any proper framework, collecting a large amount of data haphazardly will cause a disarray which will perpetuate when involving other data transformation techniques [12]. For these reasons, it is a prominent task to decide in which way DT will perceive which healthcare data from which dimensions [13]. To solve these problems, we propose a structured mathematical data model to collect the patients' data in a systematic and predefined way so that a cluster of acute information about a physical patient and its surrounding environments can be accumulated while they are at the hospital. With the proposed data model the patient can be individually identified as well as the patient portfolio can be concocted with the clinical data.

It is often reported that people show a lack of concern regarding the security of the health data which leads to integrity and confidentiality breaches [14]. Around 881 breach reports have been recorded within the last 24 months and are under investigation by the U.S. Department of Health & Human Services [15]. Therefore, to effectively solve this problem, a system is needed that can store

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Gaggero.

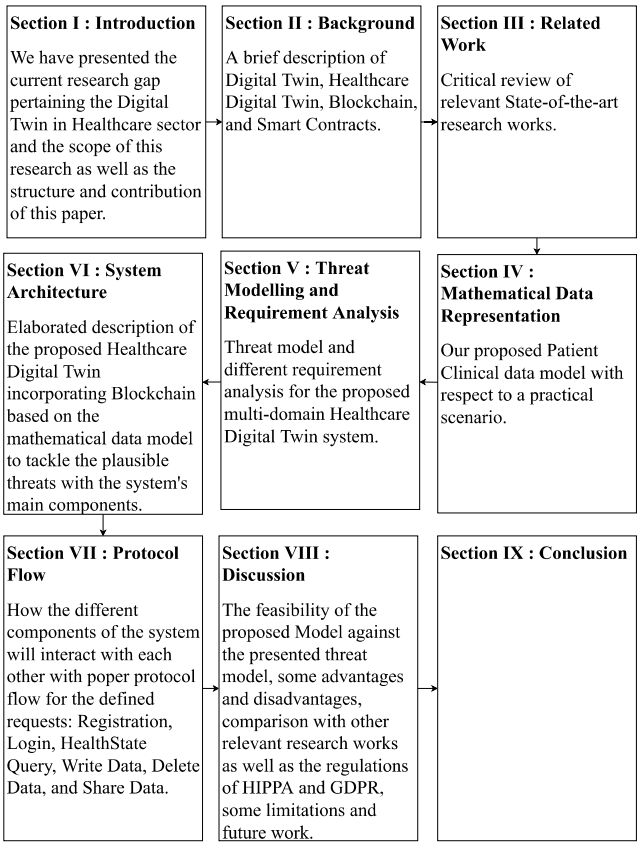

FIGURE 1. **Paper structure.**

and keep data securely with proper structure. Moreover, around 60% of the counties in the world have the capacity to review the progress and performance of the healthcare systems and around 59% can use data to drive policies and planning for the health sectors [16]. To cover these wide distributed nationwide healthcare sectors, having the mentioned potentials, a distributed network can be implemented by enforcing a distributed storage facility without any central governing authority [17]. For this reasons, the blockchain technology can be integrated with DT to accumulate this insurmountable health data in a structured and distributed way with adequate security properties. In a blockchain based DT system for healthcare, blockchain renders the services of collecting intricate and diverse data immutably with proper access and sharing mechanism, on the other hand, DT provides proper data analysis, aggregation, prognosis, and representation services which are conducive to build a proper healthcare DT. To mitigate these issues, in this article, we present a concrete mathematical model for patients' clinical data and then propose a blockchain based Healthcare Digital Twin system based on the presented data model.

A. CONTRIBUTION

The major contributions of this article is presented below:

1) A patient centric mathematical data model to represent the patient data in a defined and structured way.
2) The proper delineation of the clinical data with real life contexts which will be perceived by DT while the patient is on the treatment phase.
3) A blockchain integrated Healthcare Digital Twin System architecture based on the proposed data model with proper threat modeling and requirement analysis.
4) A number of protocol flows utilizing the blockchain based system which showcases how the system can be utilized in different scenarios.
5) A detailed analysis of the proposed system covering its feasibility, advantages/disadvantages, comparisons with Health Insurance Portability and Accountability Act (*HIPAA*) [18] and the General Data Protection Regulation (*GDPR*) [19] as well as with other existing research works.
6) Finally, the limitations and the future scopes of the presented system.

B. STRUCTURE

Section II presents a brief background on Digital Twin, Healthcare Digital Twin, and Blockchain. Next, we critically review a number of relevant researches in Section III. In Section IV, the mathematical data model is presented with some pragmatic examples. Then, we provide the detailed threat model and requirement analysis in Section V. Next, Section VI outlines the architecture of the blockchain Based System for Healthcare Digital Twin. After that, in Section VII, a number of protocol flows utilising the proposed system is illustrated with some use-cases. We discuss how the proposed system for Healthcare DT has helped to satisfy different requirements and explore its feasibility, advantages and disadvantages, a comparison with *HIPAA* and *GDPR* as well as with some recent research works, limitations, and future work of the presented model in Section VIII. Finally, Section IX. concludes our findings.

II. BACKGROUND

In this section we present a brief background on Digital Twin, Healthcare Digital Twin, and Blockchain and its different aspects.

A. DIGITAL TWIN (DT)

DT stands for the representation of the anatomy of a digital asset in a digital space which is the depiction of a physical phenomena from a physical space. It is a complex system which keeps the consistency between a digital and physical space and can develop cognitive knowledge about the physical environment [20]. The most prominent task of DT is the interaction between the physical and digital scenarios [21]. With the rapid growth of new generation of information technologies like Radio-Frequency Identification (RFID) and Internet of Things (IoT), data collection from each aspect of a physical phenomenon can be modulated conveniently [22]. DT can be divided into 2 types depending on for which purpose it will be used [23]:

1) DIGITAL TWIN FOR DEVELOPING A PRODUCT

This type represents a physical product which has not been developed yet, however, its representative DT already has all the information necessary to develop the physical product. In this respect, by using previous knowledge, the current state of the development, work distribution, product description, etc, DT can predict the workflow and the behavior of the product. For example: a DT can be implemented while at the manufacturing phase of a hospital. This falls under the Product Lifecycle Management (PLM [24]–[26]) in the healthcare sector.

2) DIGITAL TWIN FOR AN INDIVIDUAL INSTANCE

This type of DT has the awareness of a physical product or a non-spatial phenomenon and can constantly update the virtual state with the real time data from the physical space through IoT devices [27]. Let us assume that an automobile has been built with each important part integrated with sensors. So, the DT would be constantly receiving the data from this vehicle and could assess the current state of the instance through vehicle health management [28]. The maintenance time of a part, the longevity of the product, and other factors can be devised remotely which will lead to derive the knowledge of the instance.

B. HEALTHCARE DIGITAL TWIN

The primary requirement for being a DT is that both the virtual and physical properties need to be congruous at any moment of time for the corresponding purpose. From the scope of a patient centric Healthcare DT, it means a virtual patient in a digital space requiring a bulk of data, representing the patient from a physical space. But just by having only one facet of the patient data, it cannot have a holistic picture of a patient who might face uncertain health mishaps. It needs all the pertinent data of medication, hospital, and management of a patient [29]. We also need to consider that the human body is full of complexities depending on many external influences like environments, age, social activities, etc. In addition, the causal factors of a disease or co-morbidity is a completely separate case where different aspects will play differently for different individuals [30]. For this reason, from birth till death, if all the relevant data about the patient can be accumulated as much as possible, the DT can analyze and give accurate deduction on the current state or even can predict future threats for the patient by perceiving real time data [31]. This is the main motivation for a patient centric healthcare DT [23]. This is useful in many scenarios, e.g., someone needing to observe a patient remotely at a particular instance, and then the DT will provide all the necessary data for responding to the interrogation.

C. BLOCKCHAIN

Blockchain is a distributed ledger where a ledger containing all the blocks' data is distributed to all the peer nodes over the network. Because of this replication process over a distributed network, immutability and non-reputability can be easily achieved [32]. Also, by remedying the hassle of depending on a third party and central authority, blockchain has rendered an advent of technological evolution. With blockchain, members of disparate groups can carry out transactions in a distributed environment without the need of a centralized system [33]. The data stored in a blockchain is append-only as blocks can only be added at the end of the blockchain and all the blocks are linked with its previous block with the help of Cryptographic Hash Function [34]. Once a block is validated and added to the chain it cannot be erased or updated [35]. Nevertheless, it becomes really cumbersome to achieve the same state for all the nodes having the same ordered blocks of blockchain over a distributed environment. Thus, blockchain has adopted a mechanism called consensus which uses the prior agreement of the rules and follows the principle of majority dominance [36]. As a result, blockchain system primarily depends on consensus algorithms which ensure consistency over the distributed nodes [37].

Blockchain can be generally categorized in 2 types:

- **Public Blockchain:** A public blockchain allows anyone to participate in creating and validating blocks as well as write data on the blockchain. A public blockchain has no restriction over nodes, for this reason it is also called a permissionless blockchain [38]. In a public blockchain, all the transactions are public and the participants are anonymous. It has been proven to be eminent in terms of security, whereas in terms of propagation it takes a considerable amount of time, consequently, transaction throughput is limited and the latency is comparatively high [39]. Some examples of public blockchain systems are Bitcoin [40] and Ethereum [41] and public blockchain systems for healthcare are: *MedRec* [42], *FHIRChain* [43], etc.
- **Private Blockchain:** A private blockchain is permissioned and only a predefined set of entities can take part in the validation process. Such a blockchain has no concern of energy consumption and renders enough security [44]. Because of the small of participants in the block validation process, the synchronization of the blocks over the distributed network is very prompt [45]. There are circumstances where the privacy and the obscurity of systems' data are mandatory, e.g., in financial statements, health data, etc. In those cases, a private blockchain is used and for the same reason, we also envision our Healthcare DT system to utilize a private blockchain platform. The prominent private blockchain platforms are Hyperledger Fabric [46], Hyperledger Sawtooth [47], and Corda [48]. There are some private blockchain based healthcare systems, e.g., *HealthChain* [49], *ModelChain* [17], *Ancile* [34], *MeDShare* [50], etc.

D. SMART CONTRACTS

Smart contracts are computer programs that are deployed within a blockchain platform and can be executed on

distributed networks among distrusted entities. It translates real life contractual conditions into executable computer codes, integrates them with digital assets, and can run autonomously without any trusted authority [51]. As a smart contract runs inside an immutable blockchain environment, its code is immutable and therefore the required contract needs to be properly validated before deploying it on the blockchain [52].

III. RELATED WORK

From the advent of DT, because of its numerous adaptability and usefulness, it has piqued the attention of the researchers. Here we provide an overview of some of the recent notable research. Peng *et al.* [53], in their article have presented a construction case on hospital DT in China, which had already been built. The authors have delineated how the hospital twin has been developed based on Continuous Lifecycle Integration method. A lot of sensors, for acquiring real time data of the hospital, have been planted during construction and the whole system can be controlled from a single point through DT. However, there is nothing mentioned about access control and encryption mechanisms for the collected data.

On the other hand, Liu *et al.* [54] have proposed a cloud based framework with healthcare DT. The reason behind the project is that there are elder people who hardly take medical services because of their indifference toward diseases. The authors have developed the system comprising of 4 parts: Physical object, Virtual object, Cloud healthcare service platform, and healthcare data. Although, some important aspects have been described, however, no algorithm has been mentioned for the predictive measures.

With the help of edge computing, in this article [9], the authors have developed a healthcare Twin to alleviate heart diseases. The real time data will be collected through IoT devices with the means of smartphones and the resultant data will be stored in a central data storage after going through data fusion transformation. They have developed the twin in 3 structures: data source, AI-Inference Engine, and Multi-modal Interaction and Smart Service. Their main incentive is to train a Convolutional Neural Network (CNN), though the issues of data storage and security concerns have not been addressed.

A similar type of work has been presented by Shamanna *et al.* [55], introducing Precision Nutrition to DT. The paper is about Twin Precision Nutrition (TPN) which monitors a group of 64 years old type 2 diabetic patients to reduce HbA1c in blood. The platform collects data from body sensors and a mobile app (TPN) to track and analyze the body health signals in order to personalize the patients' treatment. 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.* [56], in their article have proposed an architecture combining the qualities of a generative model with a graph-based representation of pathophysiological 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 structured way. Moreover, they have utilized graph neural networks for deep learning and produced predictions about the evolution of the physiological state of the patient.

In [57], 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. One of the primary components of this platform is functionality for collecting data for the DT. Another component is advanced analytical application which provides services for data aggregation, enrichment, analysis, and visualization, and can be used to produce new knowledge and support decisions. The authors have mentioned that the patients' data will be collected from Electronic Health Records (EHRs), open clinical datasets, data from social networks, and other external applications. However, they have not provided any necessary steps to stop data integrity and confidentiality breaches.

A risk diagnosis Digital Twin system has been proposed in [58] to enhance the decision makings for liver diseases with explainable artificial intelligence. The authors have used the Random Forest (RDF) model they have developed and a state-of-the-art explainable artificial intelligence library called Local Interpretable Model-Agnostic Explanations (LIME). As healthcare is a delicate matter for this reason they have proposed to use explainable AI. 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 [59]. In this work, the authors have proposed and implemented a framework which is beneficial to digital healthcare and healthcare operations. An intelligent context-aware healthcare system is described and it is mainly focused on diagnosing heart problems and detecting heart disease by classifying ECG heart rhythms with DT. DT combines Artificial Intelligence (AI), Data Analytics, IoT, Virtual and Augmented Reality paired with digital and physical objects. This integration allows real time data analysis, status monitoring to tackle problems before they even occur. It also provides insights regarding risk management, cost reduction, and future opportunities prediction. Like other works discussed here, they have not taken any precaution to safeguard the stored data.

In article [60], the authors have provided a vision about how multi-agent systems can be integrated with DT in the healthcare domain. Here, multi-agent means a software agent which can give a response before taking any action. From the perspective of DT, agents provide a blueprint for engineering intelligent systems embedding AI and Distributed AI techniques, featuring some level of autonomy on top of DTs, so that DT features can be exploited. But, the authors have not provided any empirical process or analysis for the proposal.

From these developments, it is understandable that a large degree of advancement is happening in the healthcare sector for DT. Unfortunately, the most grievous issue is, how this learning and analysis will proceed if elaborated and sophisticated data cannot be gathered from the physical environment. Also, how this bulk data and insights can be stored with integrity and confidentiality is the main concern for DT in the Healthcare sector. With these in mind, we will try to assuage the collection of ambiguous data by providing a conceptual data model mathematically in Section IV and will solve the insecure storing problem by representing a full system architecture for blockchain based Healthcare Digital Twin in Section VI.

IV. MATHEMATICAL DATA REPRESENTATION

Let us denote the Healthcare DT system as HDT and moreover H and P denote the set of all hospitals and patients respectively under HDT. The set of peripheral hospitals can be denoted as HP. To have a comprehensive and cogent understanding, let us consider a scenario where Brad is a patient who was using services from some hospitals, $hp \in HP$, outside of HDT, as a result no data about Brad is available in HDT and now he is inclined to take the services of our proposed HDT. So, he will go through the registration process in a hospital, $h \in H$ under HDT. His patient identity will be stored with a unique userName which can be worked as an identifier and at the same time all his patient centric previous data will be collected from hp with the means of a system, $s \in S$ under HDT. S renders the chance to extract data from outside the jurisdiction of HDT with the help of some special APIs. The elaborated analysis of System S has been given in Section VI and Section VII and $HDT = \{H, S\}$.

A human body works in a complex way. For this reason which factors affect the body in which way is very difficult to estimate. For this reason, from social status to health condition, all data regarding Brad will be accumulated to have a better understanding of any disease and the surroundings. Social data contains Brad's age, home attributes, workplace, etc [61]. There is a possibility that Brad's condition is already severe and he has been taking care-giving services so Brad's caregivers' information will be collected from hp. There will be a myriad of similar types of data like Brad's previous tests, medication, surgery data which will be accumulated succinctly. There are some special types of data like food intake, regular exercise, breathing counts, etc, which will be collected with the consent of Brad as self reported data assuming Brad will provide the correct data in a timely manner. Here, we assume that Brad will be honest providing such data accurately for the betterment of his own health. 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's regular life and 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 2.

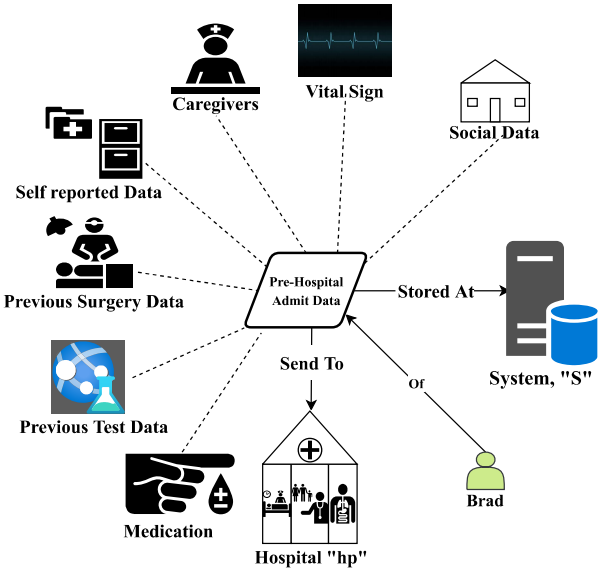

FIGURE 2. Pre-Hospital Admit Data (PHA): The necessary data required before introducing a patient to the system.

TABLE 1. Notation & semantics for pre-hospital admit data.

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

The necessary semantics and notations for Pre-Hospital Admit Data are described in Table 1.

As our Healthcare DT system has an explicit objective to assuage the ambiguity for the care of patients, defining the encompassing data of patients will be the most prominent and preemptive task. So, while introducing a patient, $p \in P$, like Brad who is new to the HDT, it is ubiquitous that there will be various Pre-Hospital Admit Data that needs to be collected through $s \in S$ from other systems meticulously with proper 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. 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 (SD_s^p) of Pre-Hospital Admit Data for p under s with respect to time is defined in Equation (1), as shown at the bottom of the next page.

For convenience let us consider all the data was accumulated at the same time (*Tstamp*). As a patient, Brad's Social Data can be considered as presented in Equation (2), as shown at the bottom of the page, where SD_s^{Brad} represents a set of Social Data for Brad under system $s \in S$. 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 defined in Equation (3), as shown at the bottom of the page.

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 (from where Brad had been taking services) has Caregivers' data represented according to Equation (4), as shown at the bottom of the page. So, $patientToCaregiver\ (Brad,\ s) = CG_s^{Brad}$ can be accumulated using Definition (1). An example is provided in Equation (5), as shown at the bottom of the page. All the accessible previously stored data regarding Brad's caregivers from other systems can also be amassed with the help of the function defined in 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 depicted as Equation (6), as shown at the bottom of the page.

Now, Brad has become a patient of the hospital $h \in H$ under HDT. Before starting the treatment, Brad's disease needs to be determined. So, to determine Brad's disease, a specialist physician will be assigned [62]. The physician will provide a Check-Up Prescription containing a myriad of tests and health conditions. After getting the Check-Up Prescription, Brad will go through the tests. The tests data will be recorded as Diagnose Test Data [63]. 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 3. The necessary semantics and notations for Patient Disease Diagnose Data are described in Table 2.

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 $ph \in PH_h$. Equation (7), as shown at the bottom of the page, can define the ph provided Check-Up Prescription (CUP_h^p).

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 (DTD_h^p) set for patient $p \in P$ within the system h can be represented according to Equation (8), as shown at the bottom of the page.

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 (DTD_h^{Brad}) for Brad can be portrayed as in Equation (9), as shown at the bottom of the page.

Let us denote the set of all sensors SN_h under system h. The sensor data (PS_h^p) for p, perceived from various sensors can be considered as presented in Equation (10), as shown at the bottom of the page. Now, the Patient Disease Diagnose

$$SD_s^p = \left\{ (AA_{SD}^p \times AV_{SD}^p) \times t \mid AA_{SD}^p \,\&\, AV_{SD}^p \text{ is defined } \wedge\ t \in TS \right\} \tag{1}$$

$$SD_s^{Brad} = \{(Age,\ 38),\ (Sex, male),(Address,\ Dhaka - BD),\ (Type, maintenance),\ (Env., outdoor),\ (Tstamp, 1625240415)\} \tag{2}$$

$$CGD_S = \left\{ (AA_{CGD}^{cg} \times AV_{CGD}^{cg}) \mid AA_{CGD}^{cg} \,\&\, AV_{CGD}^{cg} \text{ is defined } \wedge\ cg \in CG_S \right\} \tag{3}$$

$$CG_S \times CGD_S = \{\ \{\ (Caregiver - Name, Bob), (Patient, Brad), (Duration, 213\ days),\ (Tstamp, 1625240415)\ \},\ \ldots\ \} \tag{4}$$

$$patientToCaregiver\ (Brad,\ s) = \{\ \{\ (Caregiver - Name, Bob), (Duration, 213\ days), (Tstamp, 1625240415)\ \},\ \ldots\ \} \tag{5}$$

$$PHA_s^p = \left\{ SD_s^p \cup VS_s^p \cup MD_s^p \cup SR_s^p \cup PSG_s^p \cup PTD_s^p \cup CG_s^p \right\} \tag{6}$$

$$CUP_h^p = \left\{ (AA_{CUP}^{ph} \times AV_{CUP}^{ph}) \times t \mid AA_{CUP}^{ph} \,\&\, AV_{CUP}^{ph} \text{ is defined } \wedge\ ph \in PH_h \wedge\ t \in TS \right\} \tag{7}$$

$$DTD_h^p = \left\{ (AA_{DTD}^e \times AV_{DTD}^e) \times ph \times t \mid AA_{DTD}^e \,\&\, AV_{DTD}^e \text{ is defined } \wedge\ t \in TS \wedge\ e \in E_h \right\} \tag{8}$$

$$DTD_h^{Brad} = \{\ \{(PhysicianName, Selim), (Type, diagnostic), (TestData, tr), (Tstamp, 1625240415)\},\ \{(PhysicianName, Mark), (Type, Laboratory), (TestData, tr), (Timestamp, 1625240415)\}\ \} \tag{9}$$

$$PS_h^p = \left\{ (AA_{PS}^{sn} \times AV_{PS}^{sn}) \times t \mid AA_{PS}^{sn} \,\&\, AV_{PS}^{sn} \text{ is defined } \wedge\ t \in TS \wedge\ sn \in SN_h \right\} \tag{10}$$

$$PDD_h^p = \left\{ CUP_h^p \cup DTD_h^p \cup PS_h^p \right\} \tag{11}$$

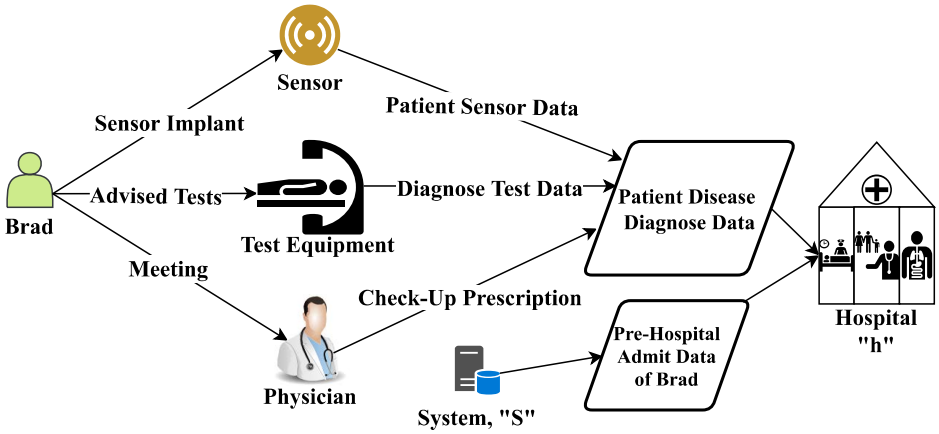

FIGURE 3. Patient Disease Diagnose Data (PDD): The data accumulated while patient goes through disease assessment phase.

Data (PDD_h^p) aggregation set can be depicted according to Equation (11), as shown at the bottom of the previous page.

At this point, as a patient, Brad can be defined by the set of Pre-Hospital Admit (PHA_s^{Brad} from Equation 6) and Patient Disease Diagnose Data (PDD_h^{Brad} from Equation 11). But, there may be need of a 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 (PDD_h^{Brad}) as well as newly found 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 [64]. 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, this surgery sequence data needs 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 [65]. 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 [66]. The necessary Notation and semantics for Surgical Operative Procedure are described in Table 3 and illustrated in Figure 4.

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 is $e \in E_h$, then Pre-Operative Assessment (POA_h^p) can be defined according to Equation (12), as shown at the bottom of the next page.

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 as in Equation (13), as shown at the bottom of the next page.

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 defined according to Equation (14), as shown at the bottom of the next page.

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 the surgery type and the relevant data regarding different sequences will be recorded by a member of ST_h^p, usually surgeon's assistant [65]. The Sequence of Surgery (SOS_h^p) can be defined as in Equation (15), as shown at the bottom of the next page.

Some special notable or exceptional attributes of the surgery will be recorded by the surgeon and will be stored as Surgeon Specific Factor (SSF_h^p) and can be defined as in Equation (16), as shown at the bottom of the next page. In the same manner, a compendium knowledge about Operating Room Factor (ORF_h^p) regarding surgery will be recorded. Equation (17), as shown at the bottom of the next page, can be used to represent ORF_h^p.

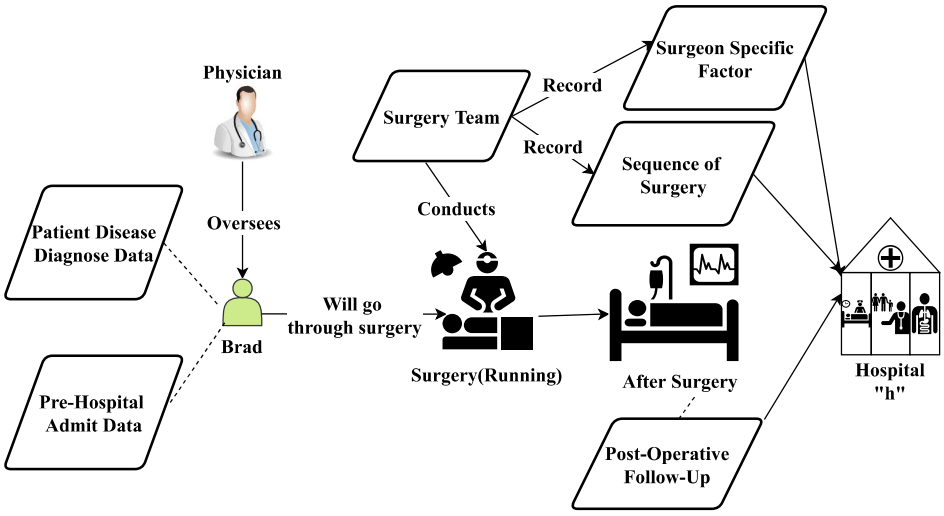

FIGURE 4. Surgical Operative Procedure (SOP): All the surgery pertinent data.

After the surgery, patient p will be under complete observation in hospital h. The Post-Operative Follow-up data (POF_h^p) for p can be defined as in Equation (18), as shown at the bottom of the page.

Based on Patient Disease Diagnose Data and Pre-Operative Assessment data ($PDD_h^p \wedge POA_h^p$) from Equation (11) and Equation (13) respectively, 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 (SOP_h^p) and this action can be represented as in Equation (19), as shown at the bottom of the page.

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.

TABLE 2. Notation & semantics for patient disease diagnose data.

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

In the same manner, for a patient $p \in P$ under the Healthcare DT system $hdt \in HDT$, HDT is comprised of system H and S, so $hdt = \{h, s\}$ where, $h \in H$ and $s \in S$, a compendium representation of Patient Data (PD_{hdt}^p) can be defined by using Equation (20), as shown at the bottom of the page, accumulating Equation (6), (11), and (19).

$$POA_h^p = \left\{ (AA_{POA}^e \times AV_{POA}^e) \times t \mid AA_{POA}^e \,\&\, AV_{POA}^e \textit{ is defined } \wedge\ t \in TS \wedge\ e \in E_h \right\} \tag{12}$$

$$POA_h^{Brad} = \left\{ \{ (\textit{AdvisedCheckUp } cp) (\textit{PreSurgeryTestData } tr) (\textit{SurgeryNumber } 1) (\textit{Tstamp } 1625240415) \} \right\} \tag{13}$$

$$STD_h = \left\{ (AA_{STD}^{stt} \times AV_{STD}^{stt}) \times SG_h \times t \mid AA_{STD}^{stt} \,\&\, AV_{STD}^{stt} \textit{ is defined } \wedge stt \in ST_h \wedge\ t \in TS \right\} \tag{14}$$

$$SOS_h^p = \left\{ (AA_{SOS}^{st} \times AV_{SOS}^{st}) \times sg \times t \mid AA_{SOS}^{st} \,\&\, AV_{SOS}^{st} \textit{ is defined } \wedge\ st \in ST_h^p \wedge\ sg \in SG_h \wedge\ t \in TS \right\} \tag{15}$$

$$SSF_h^p = \left\{ (AA_{SSF}^{st} \times AV_{SSF}^{st}) \times sg \times t \mid AA_{SSF}^{st} \,\&\, AV_{SSF}^{st} \textit{ is defined } \wedge\ st \in ST_h^p \wedge\ sg \in SG_h \wedge\ t \in TS \right\} \tag{16}$$

$$ORF_h^p = \left\{ (AA_{ORF}^{st} \times AV_{ORF}^{st}) \times sg \times t \mid AA_{ORF}^{st} \,\&\, AV_{ORF}^{st} \textit{ is defined } \wedge\ st \in ST_h^p \wedge\ sg \in SG_h \wedge\ t \in TS \right\} \tag{17}$$

$$\begin{aligned} POF_h^p = \Big\{ \big((AA_{POF}^{ph} \times AV_{POF}^{ph}) \cup (AA_{POF}^{sn} \times AV_{POF}^{sn}) \cup (AA_{POF}^{e} \times AV_{POF}^{e})\big) \times t \mid AA_{POF}^{ph},\ AV_{POF}^{ph},\ AA_{POF}^{sn}, \\ AV_{POF}^{sn},\ AA_{POF}^{e} \,\&\, AV_{POF}^{e} \textit{ is defined } \wedge\ sn \in SN_h \wedge\ e \in E_h \wedge\ ph \in PH_h \wedge\ t \in TS \Big\} \end{aligned} \tag{18}$$

$$SOP_h^p = ST_h^p \xrightarrow[(SOS_h^p \wedge SSF_h^p) \models POF_h^p]{(PDD_h^p \wedge POA_h^p) \prec ORF_h^p} \tag{19}$$

$$PD_{hdt}^p = \left\{ PHA_s^p \cup PDD_h^p \cup SOP_h^p \right\} \tag{20}$$

V. THREAT MODELLING AND REQUIREMENT ANALYSIS

A. THREAT MODELLING

According to [23], a complex system has 4 emergent behaviors: predictable desirable, predictable undesirable, unpredictable desirable, and unpredictable undesirable. In terms of Digital Twin, where data will be perceived from physical space in real time and, concurrently the analyzing, processing, and decision making, will be done in virtual space [67], there will be a myriad of hindrances for different reasons. In this manner, threat modeling facilitates understanding various conundrums and threats before deploying the system for practical use. To model threats, we have chosen a well established threat model called STRIDE [68], developed by Microsoft, which encapsulates different security threats as presented 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.
- **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).
- **T3-Repudiation:** An attacker can repudiate after altering data.
- **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).
- **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.

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:** Unnecessary and overuse of system's calculation power. e.g., naive or with bad intentions, users may create multiple query requests for different trivial purposes which will exert the system's calculating power.

As the system deals with a raft of entities, various types of data will be exchanged from entities to entities or system to outside. In this regard, the lack of any privacy control by any user creates different privacy threats to the system. Based on this assumption, the identified privacy threats are as follows.

- **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.
- **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.

TABLE 3. Notation & semantics for surgical operative procedure.

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

B. REQUIREMENT ANALYSIS

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., if a physician wants to create a digital twin of a particular patient's heart, then the system needs to provide all the forthcoming data of that patient's sensor, diagnose test, surgery, check up prescription, and other pertinent data of the heart of that patient till that time.

F2. Users can share their private data with entities inside the jurisdiction of the system, which should be corroborated.

F3. The system must check that the storing of dynamic data from IoT devices is consistent.

F4. The system should be integrated with a private blockchain infrastructure for the implemention 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.

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.

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.

VI. SYSTEM 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. 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. 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. Next, the components will be explained elaborately.

A. 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*.

B. 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, 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.

C. 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. As the data is already

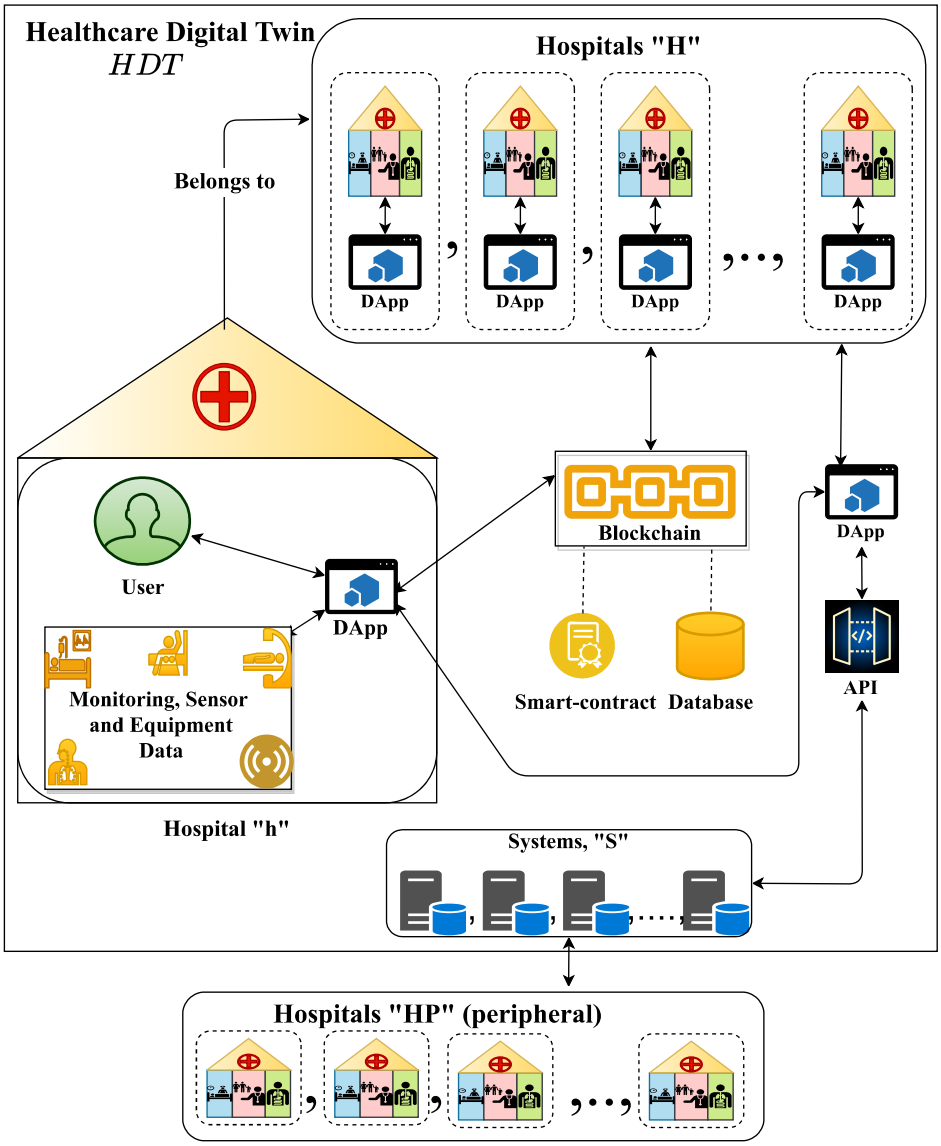

FIGURE 5. High-level architecture.

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 IV, most of the Pre-Hospital Admit Data (PHA) from Equation (6) can be collected in a codified way via a system in S.

D. BLOCKCHAIN PLATFORM

Blockchain platform is one of the core components in our proposed Healthcare DT system (*HDT*). Hospital user and equipment are clients according to the blockchain network and are linked to the 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 [69]. 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 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.

VII. 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 4 and data model in Table 5.

A. DATA MODEL

All the activities of the users in the system can be aggregated as request which is denoted as *req* in Table 5. *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 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 ($response_{req}$) and other additional data (*returnedData*) against the *req*. There are few types of *req* for which there will be no *returnedData* except $response_{req}$ for *resp*, and for these types of *resp*, *returnedData* will contain null ($\emptyset$) value. *TYPE*, *DATA*, and *resp* are defined in Table 5.

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 5. *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 nurse 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, $response_{req}$ 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

TABLE 4. Cryptographic and other notations.

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
$[]_{https}$	Communication over *HTTPS* 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 HDT
$ZZ_{s}^{p.t}$	The resultant data set queried from outside HDT
WDT	New unassigned generated data
$\emptyset$	null

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 5. 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 IV, to do read, write, and write operation. In the definition of *permissions* in Table 5, $q \in PD_{hdt}^{p}$ from Equation (20) 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 corresponding 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 IV. By providing *userName*, conditions (c_n, here $n = 1, 2, \ldots, k$ *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 5.

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

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 (11) and (19)) 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 IV. 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 5. In the same manner for *delete* request, by providing *userName*, q, and $\{H(q)\}_{K_u^{-1}}$ as *deleteData*, the data owner can delete data from off-chain database consequently the action will be updated in the blockchain. For *delete* type *req* there will be only $response_{req}$. The *deleteData* is defined in Table 5. 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 5.

B. 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).

C. PROTOCOL FLOW

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

1) REGISTRATION PROTOCOL

To participate, a user must register following the protocol presented in Table 6 and illustrated in Figure 6. 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 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 (6)) 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, 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.
  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). Now, *regisFunc*

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

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

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 (Line 18).

Step 4: Later, BCH creates a transaction and returns *TRUE* for a successful completion of registration ($M4$ in Table 6). Then, SC provides an access control set entitled as *permissions* in Line 19 and the *returndedData* contains it (Line 21). A general response according to the *req* which is ''Successfully Registered'' is being provided as $response_{req}$ (Line 20).

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

Input: *req* : *Request from the user*
Ouput: *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* = $data.c_n$;
17: *utime* = *data.t*;
18: *ujson* = *getState*(*uName*); ▷ Retrieve the metadata for the user from blockchain
19: *Index* = Finding out the locations for the queried data from *ujson*;
20: **return** *Index*;
21: **end function**
22: **function** writeFunc(*data*)
23: *uName* = *data.userName*;
24: *uDataProperty* = *data.q*;
25: *uIndex* = *data.Index*;
26: *boolean x* = Store new data with a transaction and return a Boolean;
27: **if** (*x* == *TRUE*) **then**
28: $response_{req}$ = Transaction Successful;
29: **else**
30: $response_{req}$ = 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*); ▷ Retrieve the metadata for the user from blockchain
40: *Index* = Finding out the locations for the requested data from *ujson*;
41: *shareRecordjson* = {*Sender* : *uapprover*, *Receiver* : *urequester*, *Data* : *umetadata*, *Sign* : *usignature*};
42: Store shareRecordjson into blockchain with a transaction and returns a Boolean;
43: **return** *Index*;
44: **end function**

TABLE 6. Registration protocol.

$M1$	$u \rightarrow D:$	$[N_1, \{req\}_{K_D}]_{https}$
$M2$	$D \rightarrow$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:	$[N_1, resp, \{H(resp)\}_{K_D^{-1}}]_{https}$

Step 5: Then a final response *resp* containing $response_{req}$ and *returndedData* is being returned to D in Line 23 and it is defined as message $M5$ in Table 6. 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. 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).
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). The user stores its public and private keys in device for any future correspondence.

2) 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*. 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.

3) HealthState QUERY PROTOCOL

The health state query follows the protocol represented in Table 7 and Table 8. 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) 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*.

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 4. 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 (20). The resultant data, $ZZ_{hdt}^{p.t} = \{ ZZ_h^{p.t} \cup ZZ_s^{p.t} \}$

TABLE 7. HealthState query protocol under system H.

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

TABLE 8. HealthState query protocol under system S.

$M1$	$u \rightarrow$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:	$[N_1, resp, \{H(resp)\}_{K_D^{-1}}]_{https}$

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 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 7 and Table 8 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 Table 7 and Table 8, u sends the *req* encrypted with the public key (K_D) of DApp (D) to the DApp (D) under hospital h.
- **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 7) or DApp (D_s) connected to system, S ($M2$ in Table 8) 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 (BCH) in Line 18 ($M3$ in Table 7) which returns the *ujson* data set according to $M4$ in Table 7.
- **Step 4 for h:** After finding out the *Index* locations from the *ujson* for the queried data in Line 19, SC returns the *Index* number to D in congruous with $M5$ in Table 7 (Line 20).

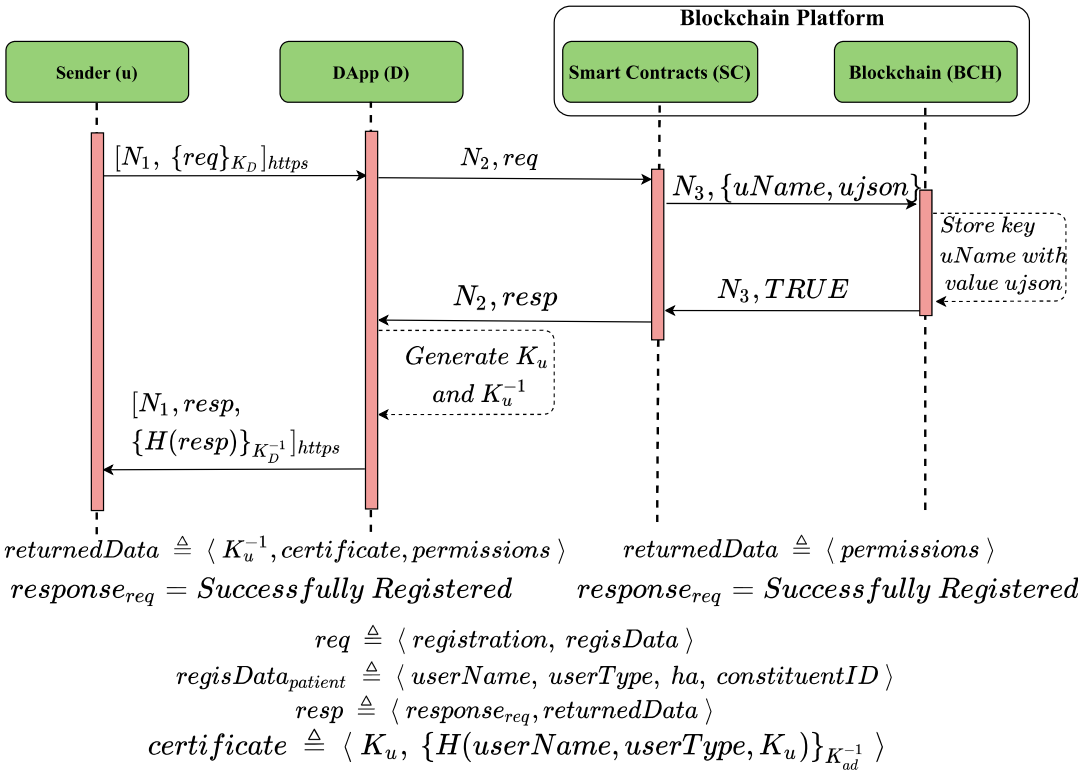

FIGURE 6. Registration flow.

- **Step 5 for h:** Without changing anything, D also relays the *Index* number toward database (ODB) according to $M6$ in Table 7.
- **Step 6 for h:** Then, ODB retrieves the *Index*s' encrypted data ($\{ZZ_h^{p.t}\}_{K_u}$) as depicted in Figure 7 and returns it to D ($M7$ in Table 7).
- **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 8).
- **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 8.
- **Step 5 for s:** Without any alteration, D_s relays back the data to D according to message $M5$ in Table 8.

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 $respnse_{req}$ is created and after that D returns the encapsulated *resp* with its signature ($\{H(resp)\}_{K_D^{-1}}$) to u ($M8$ in Table 7 and $M6$ in Table 8 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, $ZZ_{hdt}^{p.t} = \{ ZZ_h^{p.t} \cup ZZ_s^{p.t} \}$.

4) WRITE DATA PROTOCOL

Now we present the protocol flow for writing data on blockchain. The process follows the protocol flow of Table 9.

- **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 9 over an *HTTPS* channel.
- **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 $\{H(\{wdt\}_{K_u})\}_{K_u^{-1}}$. 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 (ODB) according to $M2$ in Table 9.
- **Step 3:** ODB stores the encrypted data and returns the Index number of the stored data to D ($M3$ in Table 9).
- **Step 4:** Now D replaces the encrypted data property of *writeData* with Index number and forwards the *req* to smart contracts (SC) ($M4$ in Table 9).
- **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 9 (Line 26).
- **Step 6:** BCH creates a transaction for writing the healthcare data on ODB and for a successful transaction operation a Boolean response is being returned to SC ($M6$ in Table 9).

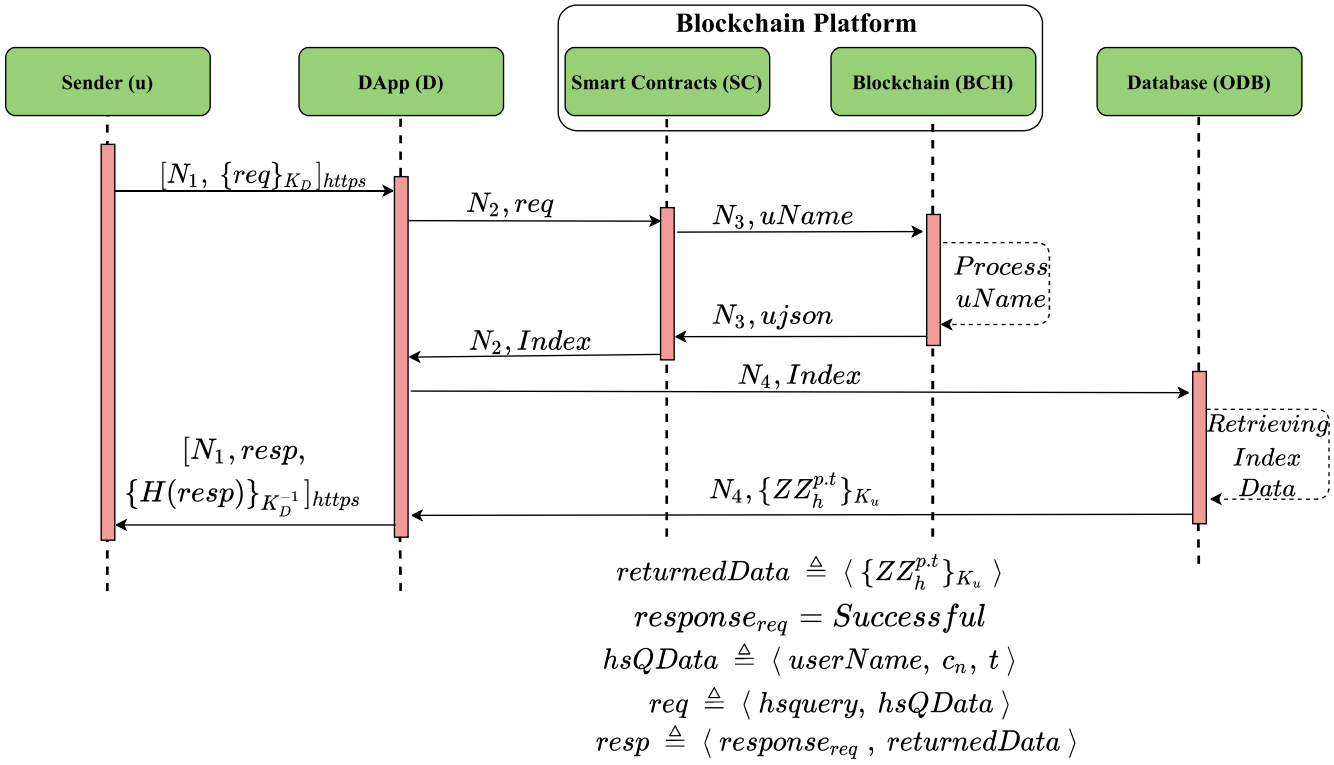

FIGURE 7. HealthState query flow under system H.

- **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 *retunrnedData* 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 9.
- **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 9 over an *HTTPS* channel.

5) DELETE DATA PROTOCOL

The protocol flow for deleting data follows the protocol flow of Table 10 and is illustrated in Figure 8.

- **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 10 over an *HTTPS* channel.
- **Step 2:** After decrypting the *req*, D forwards it to SC according to $M2$ in Table 10.
- **Step 3:** Now, SC sends the userName and Data Property to blockchain (BCH) in accordance with $M3$ which is illustrated in Figure 8.
- **Step 4:** Then, SC retrieves the Index information of the data from BCH ($M4$ in Table 10).
- **Step 5:** SC returns the *Index* to D ($M5$ in Table 10).
- **Step 6:** Without changing anything, D sends the *Index* number to off-chain database (ODB) in congruous with $M6$ in Table 10.
- **Step 7:** ODB deletes the specified Index data and returns *TRUE* to D in accordance with $M7$ which is depicted in Figure 8.
- **Step 8:** Now D confirms the deletion request to SC ($M8$ in Table 10).
- **Step 9:** For auditing and to stop any repudiation action, SC sends the deletion record and $\{H(q)\}_{K_u^{-1}}$ to BCH to create a transaction according to $M9$ in Figure 8.
- **Step 10:** After the compilation of transaction, BCH returns a Boolean result ($M10$ in Table 11) and SC sends *resp* containing $response_{req}$ to D in accordance with $M11$ as illustrated in Figure 8.
- **Step 11:** Finally, after signing the hash of *resp*, D returns $\{N_1, resp, \{H(resp)\}_{K_D^{-1}}\}$ to u in congruous with $M12$ in Table 10 over an *HTTPS* channel.

6) SHARE DATA PROTOCOL

The protocol flow for sharing data follows the protocol flow of Table 11 and is illustrated in Figure 9. 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

TABLE 9. Write data protocol.

$M1$	$u \rightarrow$D:	$[N_1, \{req\}_{K_D}]_{https}$
$M2$	$D \rightarrow$ODB:	$N_2, \{wdt\}_{K_u}$
$M3$	$ODB \rightarrow$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:	$[N_1, resp, \{H(resp)\}_{K_D^{-1}}]_{https}$

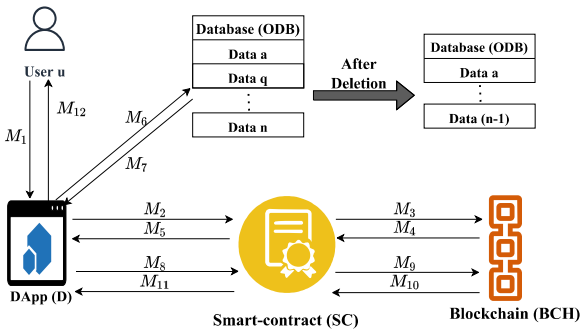

FIGURE 8. Delete data protocol.

DApp (D) in congruous with $M1$ in Table 11 over an *HTTPS* channel.

- **Step 2:** After decrypting the *req*, D forwards it to the data owner or approver (A) according to $M2$ in Table 11.
- **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 11 and it is also illustrated in Figure 9.
- **Step 4:** After that, D relays the *req* to smart contracts (SC) ($M4$ in Table 11), 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 11.
- **Step 6:** Then, SC retrieves the relevant data for A from BCH which is *ujson* in Line 39 ($M6$ in Table 11).
- **Step 7:** The *Index* locations of 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 to create a transaction according to $M7$ in Table 11.
- **Step 8:** After the compilation of transaction, BCH returns a Boolean result in Line 42 ($M8$ in Table 11).
- **Step 9:** Then SC returns the *Index* to D in Line 43 ($M9$ in Table 11).
- **Step 10:** Without changing anything, D sends the *Index* number to off-chain database (ODB) in congruous with ($M10$ in Table 11).

TABLE 10. Delete data protocol.

$M1$	$u \rightarrow$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:	$N_3, index$
$M5$	$SC \rightarrow$D:	$N_2, Index$
$M6$	$D \rightarrow$ODB:	$N_4, Index$
$M7$	$ODB \rightarrow$D:	$N_4, TRUE$
$M8$	$D \rightarrow$SC:	$N_5, req, TRUE$
$M9$	$SC \rightarrow$BCH:	$N_6, \{H(q)\}_{K_u^{-1}}$
$M10$	$BCH \rightarrow$SC:	$N_6, TRUE$
$M11$	$SC \rightarrow$D:	$N_5, resp$
$M12$	$D \rightarrow$u:	$[N_1, resp, \{H(resp)\}_{K_D^{-1}}]_{https}$

TABLE 11. Share data protocol.

$M1$	$R \rightarrow$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:	$N_4, Index$
$M10$	$D \rightarrow$ODB:	$N_7, Index$
$M11$	$ODB \rightarrow$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:	$[N_1, resp, \{H(resp)\}_{K_D^{-1}}]_{https}$

- **Step 11:** ODB retrieves the specified data $\{ZZ_h^{p.t}\}_{K_A}$ which is encrypted by A's public key (depicted in Figure 9) and returns it to D in accordance with $M11$ in Table 11.
- **Step 12:** Now D stores the acquired data as $returnedData'$ and provides a "Transaction Successful" response as $response'_{req}$ (illustrated in Figure 9). After that, D sends the encapsulated $resp'$ to the data owner signing the hash of the $resp'$ over an *HTTPS* channel ($M12$ in Table 11).
- **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 11.
- **Step 14:** Lastly, the requested data is being sent to requester R by D over an *HTTPS* channel ($M14$ in Table 11). Now, R can decrypt the data with its private key (K_R^{-1}) and can access it.

VIII. DISCUSSION

In this Section, we explore how the *HDT* system has satisfied its different requirements and the credibility of it, discuss its advantages and disadvantages, provide a comparison between our proposal with other existing works and how it is partially in-align with *HIPAA* and *GDPR* regulations and some plausible future works with some limitations.

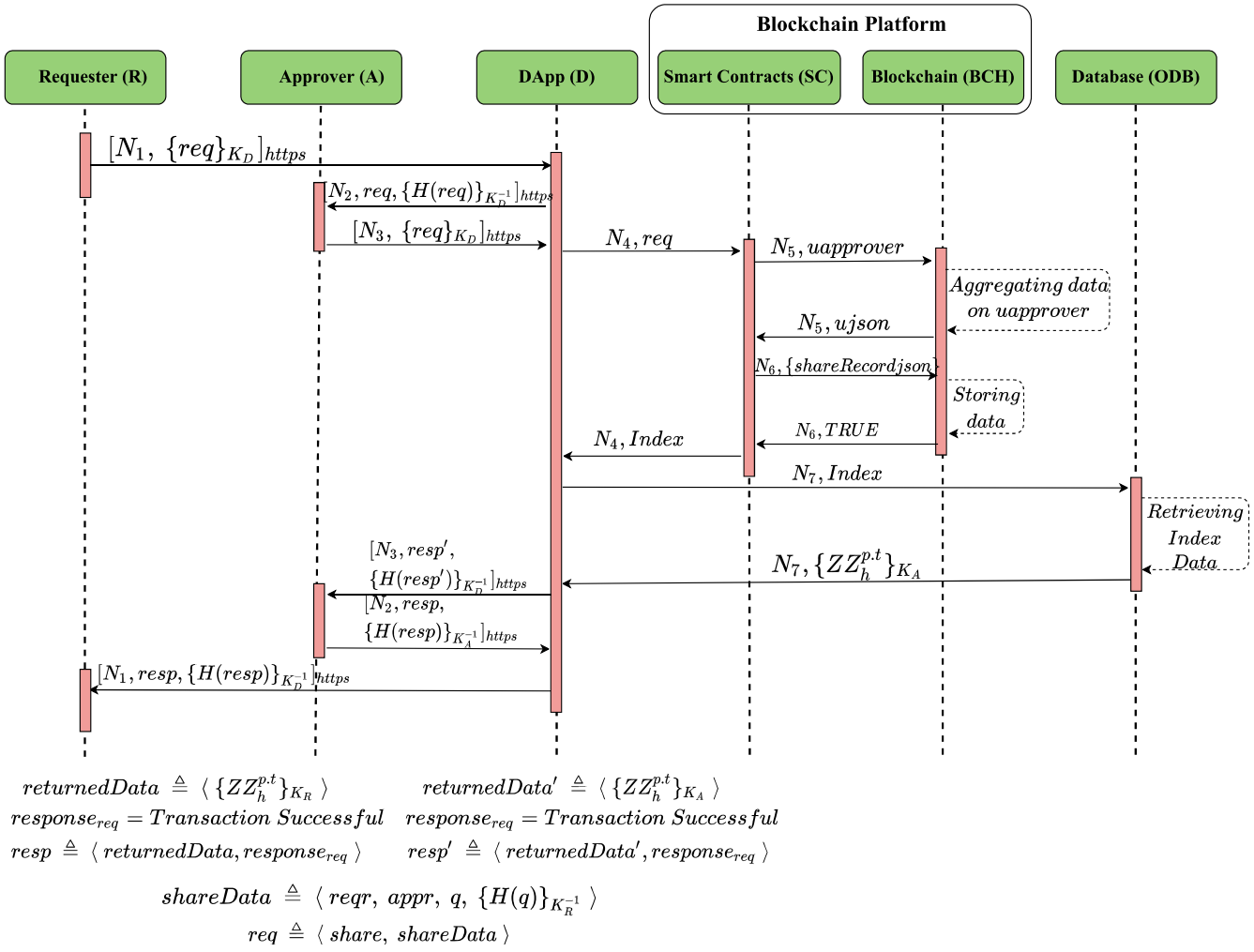

FIGURE 9. Share data flow.

A. ANALYSING REQUIREMENTS

At first we analyze how *HDT* satisfies the defined requirements of Section V.

1) FUNCTIONAL REQUIREMENTS

By correctly making a *hsquery* request (Section VII-C3), *HDT* facilitates a user to gather all the necessary data to create a digital twin, consequently satisfying F1. The *share* request (Section VII-C6) enables the sharing of personal data among entities and hence satisfies F2. By keeping the hash of each off-chain data as a transaction in the blockchain, 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 VII-C1), a *userName* attribute will be accepted by both the user and the system as a unique ID for that specific user. This satisfies F5.

2) SECURITY REQUIREMENTS

According to the defined protocol flow of Section VII, 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 [70]. 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.

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 VII-C6), 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.

B. FEASIBILITY OF HDT

At first glance, storing data in an off-chain database and updating the blockchain with the corresponding metadata concurrently seems implausible against a large number of requests. Therefore, it is important to analyze the feasibility of the proposed system.

However, the feasibility will also depend on which private blockchain platform is being used during the implementation phase. For this particular example, we are using Hyperledger Fabric, a state-of-the-art private blockchain platform [71]. The organizational aspect of Hyperledger Fabric [46] meets our multi-domain proposal scenario. As by design, *HDT* stores the data off-chain and stores the metadata, index, and other necessary information in blockchain, this consequently reduces the weights (memory size) of the transactions to store the blocks. Several researches have reported that the performance in terms of transaction per second (tps) of the Hyperledger Fabric is quite impressive. For example, authors in [72] reported a tps of 20000 by introducing a novel consensus algorithm for Hyperledger Fabric. To add, Hyperledger Fabric supports a pluggable consensus algorithm feature, meaning it can accommodate different consensus algorithms for different applications. Another research by Purbo *et al.* [73] reported a favourable performance of Hyperledger Fabric in comparison to MySQL database. Also, a comparatively low fault tolerance requirement of a digital twin system can be effectively satisfied using a blockchain system for providing the core functionalities and using a distributed database for storing off-chain data.

The improvement in performance and scalability of blockchain systems is an active research area. We strongly believe that many future blockchain systems (both private and public) will be comparable with many traditional databases in different performance matrices.

C. ADVANTAGES AND DISADVANTAGES

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

- *HDT* extracts only the external data that needs to be collected from outside systems in a predefined way with the consent of the data owners and requester. In this way, the system is more efficient and can collect data without any data conversion techniques.
- The proposed *HDT* system is the first system of its kind which is based on a concrete mathematical model. The rigorous mathematical model ensures that data from multitude sources are collected and taken into consideration while creating a health digital twin for a user.
- With the feature of aggregating data from external *HDT*, the system is not only confined with the data from an internal system, which renders an effective measure for healthcare sectors.
- The data can be shared among hospitals or other entities with proper synchronization with the help of defined requests.
- Residing all the raw data in an off-chain database, blockchain does not have to be burdened with insurmountable data, which makes the system faster and more convenient.
- By utilizing a private blockchain, all the features of blockchain, such as decentralization, transparency, immutable transaction data, and other benefits can be achieved.

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

- At this point, there is only read access to the external patient data that is stored outside the system. By building a system with a proper access control mechanism and a set of APIs, this issue can be resolved.
- As per our proposal, data is stored in an off-chain database. So, there is a high chance of a threat to availability. To ensure that there is no single point of failure, distributed databases such as BigchainDB [74] or IPFS [75] can be utilized. However, proper security measures must be ensured to facilitate this in such cases.
- Certain patient health data need to be pre-shared with other entities of the *HDT* with the consent of the corresponding patient which is not introduced in the proposed model. Some other security threats can arise from this amenity if it is incorporated without due considerations.

D. COMPARISON

Now we present a comparison between our proposed *HDT* with other recent research works based on some evaluation criteria. The selected criteria and the comparison is presented in Table 12. We have used notation "$\circ$" to indicate a research work is out of scope for corresponding criterion while the notation "$*$" implies the corresponding criterion is not needed for the research work. The notation "$\circledcirc$" represents there is nothing mentioned regarding the criterion for that specific research work. Moreover, the notation "$\blacktriangleright$" means the criterion is needed but not considered for the corresponding research work. We have used the notation "$\CIRCLE$" 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 work has fully considered the corresponding criterion and has been implemented.

A few observations can be made from Table 12:

- From the perspective of user identity and authentication, it can be perceived from Table 12 that most of the research works lack any authentication or identity mapping mechanism, whereas our proposed system has proper authentication and secure identity storage facility with the help of defined protocol flow in Section VII.
- Collecting data through different sources without any ambiguity is the most prominent and fundamental task for developing a solid structure of digital twin. Having a predefined data model can facilitate collecting and storing of this insurmountable amount of data. Among the mentioned research works, [54] and [56] have developed a few data models to acquire data in defined ways, though, there was no delineation of the used data models. On the contrary, from the perspective of a patient, we have presented a concrete mathematical data model with a practical use case in Section IV.
- Depending on the collection of data, there may need of some data conversion before using the data for analyzing. Some of the selected works have used data pre-processing techniques for refining acquired data. For the scope of our proposed system, we have not discussed anything regarding data pre-processing technique as most of the data will be collected in accordance with the predefined data model.
- In terms of data sharing, most of the research do not provide any definite information. Data share is an important criterion for a system to share data among its entities. With the presented share data protocol in Section VII-C6 for our proposed system, data can be shared among entities with proper integrity and confidentiality.
- Additionally, privacy is a prominent factor for handling data inside or outside the system environment. From the stated comparison in Table 12, 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.
- For different purposes, previous data of the targeted domain can be used in many ways, e.g., patient data, generating knowledge, isolating important features,etc. Therefore, collecting such data is a fundamental task with respect to the research perspective. According to Table 12, most of the works have accumulated previous data, but the methods of collecting data are not dynamic and mostly inefficient. On the contrary, with our proposed system, previous data of patients can be acquired by means of the presented system *S* in Section VI-C and the HealthState Query protocol in Section VII-C3.

From the Table 12, we can conclude that our proposed system provides better security and privacy in comparison to other works. Being able to collect data in a structured way, facilitates faster data handling. Furthermore, the previous data collection with respect to necessities and healthcare data sharing with the help of blockchain are some novel features from the perspective of the digital twin in the healthcare sector which let our proposed system stand out among other research works.

E. SYNERGY WITH HIPAA AND GDPR

There are a number of territorial and international regulations for ensuring the privacy and safety of sensitive patient health data [76]. Among these regulations, Health Insurance Portability and Accountability Act (*HIPAA*) [18] and the General Data Protection Regulation (*GDPR*) [19] are the prominent ones. Now, we present a short analysis how *HDT* can align with *HIPAA* and *GDPR* regulations from a patient's perspective. There are a myriad of sub-sections within these regulations, however, we have selectively chosen a few which are appropriate for the scope of *HDT*.

At first, let us discuss about how *HDT* can be aligned with *HIPAA*'s privacy rules [77]

1) SIMILARITIES

- *HIPAA* states all patients need to be identifiable individually [78]. In our case, each user needs to go through a *registration* request (Section VII-C1) and with the assertion of an administrator, *userType*, and a system provided *certificate*, not only patients but also all other type of entities can be individually identifiable.
- According to *HIPAA* clause 45 CFR 164.502 (a), for a system like *HDT* has to provide amenity to physicians so that they can request access to the patient data from a patient. With the help of *share* request (Section VII-C6), a patient can share requested data with physicians.
- *HIPAA* administrative rule 45 CFR 164.304 states that the confidentiality, integrity, and availability of any patient data, which is concocted by a physician or other healthcare personnel, needs to be ensured [79]. *HDT* stores all the hash of the stored data in blockchain which ensures the integrity of patient data. Similarly, all data are encrypted while being stored in the database which ensures the confidentiality. Furthermore, no one can read/share the data without the endorsement or consent of a patient which in a way ensures privacy.

2) DISSIMILARITIES

- *HIPAA* states that on certain circumstances (e.g., investigation, government ordinance, etc), physicians are obligated to share patient data (45 CFR 164.502 (a) (2)). However, *HDT* does not comply with this rule as to compile a transaction, a user's request needs to be valid and endorsed.

Now, we present our analysis between *HDT* and *GDPR*.

TABLE 12. Comparison among some state-of-the-art digital twin research works.

Research Work	Criteria to compare							
	Identity	Authen-tication	Data Share	Storage System	Data model	Data Preprocessing	Privacy (Access Control, Encryption, Security)	Previous Data collection
Hospital Buildings [53]	⊛	⊛	○	Data warehouse	BIM Technology	Booklet of interior data standards	○, ○, ●	Genotype-Tissue Expression Dataset
CloudDTH [54]	Cloud ser-vices	Cloud ser-vices	⊛	Cloud	5 types	Data cleaning, fusion	●, ⊛, ●	Patient Health Record
Cardio Twin [9]	○	○	⊛	⊛	⊛	Data fusion	○, ○, ○	ECG for recent patient update
Reducing HbA1c [55]	Not anony-mous	►	⊛	◐	⊛	Data fusion	►, ►, ●	Vitals, electrocardiogram, and biothesiometry
Graph Representation [56]	○	○	○	⚹	4 types	⚹	○, ○, ○	●
Diagnostics and Rehabilitation [57]	Anony-mous	►	►	PostgreSQL Relational Database	Patinet Information Model	Cleaning, consistency checking, linking	●, ⊛, ●	Patient and clinician personal record
Digital Twin Clinical DSS [58]	⊛	⊛	○	⊛	⊛	⚹	○, ○, ○	Indian Liver Patient Dataset
Intelligent Healthcare Systems [59]	Not secure	►	⊛	Cloud	Model-Building	Cleansed, preprocessed, transformed, filtered	►, ►, ►	MIT-BIH Arrhythmia Database
Agents and digital twins [60]	⊛	⊛	⊛	Cloud infrastructure	⊛	⊛	►, ►, ●	Digital twin data from its lifecycle
Our proposed *HDT*	●	●	●	Private Blockchain and Off-chain database	Concrete mathemati-cal model	○	●, ●, ●	Separate system, S

Symbol Legends: Out of scope: ○, No need: ⚹, Not mentioned: ⊛, Needed but not available: ►, Mentioned but not stated: ◐, Applied: ●

3) SIMILARITIES

- Article 20 (1 (a, b), 2) of *GDPR* dictates that patients have the full authority to get a copy or portability of personal health data. *HDT* is compliant to this rule as the patients are the owner of their data and *share* request (Section VII-C6) facilitates a patient to share a portion or full health data with other entities.
- *GDPR* Article 16 and Article 17 state that patients have the right to update and delete patient data [80]. In the same manner, *HDT* provides the facility to delete and update patient data with the means of *delete* (Section VII-C5) and *hdatawrite* (Section VII-C4) requests.
- According to *GDPR* Article 7, patients have the full right to provide or restrict the permission to process its healthcare data. *HDT* renders the patients full ownership and access control authority over the respective patient data.
- *GDPR* Article 15 (1(b, c, g), 2, 3) collectively states that the patients will be notified if their data have been updated, shared or disclosed to any 3rd person [81]. *HDT* works in harmony with this regulation as no data can be read, written, and updated without the consent of the patient according to the Protocol Flow described in Section VII.

4) DISSIMILARITIES

- *GDPR* Article 21 states that patient has the full right to stop processing of its data. In terms of *HDT*, we have not developed any request to restrict the process of patient data on-process.
- *GDPR* Article 24, Article 26, and Article 28 collectively state that an entity controller will handle all the requests, policies, data processing, etc, for patients [82]. However, *HDT* system does not comply with this regulation as it has no such entity that will govern the processes except for blockchain consensus. However, the smart contract handles all the requests according to the predefined ways and hence, such a smart contract can act as a controller.

One of the crucial design goals of *HDT* was that no healthcare data would be stored on-chain, rather a traditional 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'). 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 [83]–[85].

Since we propose the usage of a traditional database, the availability of data stored in such blockchain needs to be ensured with appropriate technologies such as distributed database. Another important privacy issue that must be considered is that transactions record in the blockchain do not become a source of inference so that an authorized entity can attain unforeseen and unwanted inference (e.g., recognising to visit a hospital or a specialised treatment facility can be used to infer certain diseases and other problems). A careful consideration must be carried out while implementing the *HDT* to ensure it does not happen.

F. LIMITATIONS

There are some limitations which are presented as bellow:

- There is no facility for data analysis concern, e.g., forecasting, health state estimation, etc. With due consideration, after implementing the proposed model, it will be taken into account.
- The proposed model does not provide any mechanism to regulate the data generated inside the system after deployment. Without proper services and data analysis methods, defining the data in use of the *HDT* is unfathomable, so during the implementation phase, we will work on it.
- On special circumstances, e.g., death, emergency, etc, how the data disclosure or ownership will be handled have not been taken into account.

G. FUTURE WORK

Now, we present some future works.

- The proposed system model can bring a new spectrum in the domain and in future we will develop and implement the proposed system and examine its pragmatic influence and performance.
- As *HDT* can be used for simulation, forecasting and estimation of future health conditions, it would be useful to develop a predictive system on top of the proposed architecture. In future, once the system has been developed, there will be scopes to explore how these aspects can be accommodated for the proposed *HDT*.
- In future, we will also investigate how to add additional features which could remove dissimilarities between *HDT* and *HIPAA* as well as between *HDT* and *GDPR* as mentioned in Section VIII-E.

IX. CONCLUSION

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.

REFERENCES

[1] *Primary Health Care, World Health Organization*. Accessed: Apr. 8, 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/primary-health-care

[2] 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, 2021.

[3] 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. 74044–74067, 2021.

[4] 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 *Proc. 5th Int. Conf. Commun. Electron. Syst. (ICCES)*, Jun. 2020, pp. 870–875.

[5] M. A. Bazel, F. Mohammed, and M. Ahmed, "Blockchain technology in healthcare big data management: Benefits, applications and challenges," in *Proc. 1st Int. Conf. Emerg. Smart Technol. Appl. (eSmarTA)*, Aug. 2021, pp. 1–8.

[6] J. Stauffer and Q. Zhang, "S2Cloud: A novel cloud system for mobile health big data management," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput.; Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData) IEEE Congr. Cybermatics (Cybermatics)*, Dec. 2021, pp. 380–383.

[7] T. L. Rodziewicz, B. Houseman, and J. E. Hipskind, *Medical Error Reduction and Prevention*. Tampa, FL, USA: StatPearls Publishing, 2021.

[8] Z. Liu, N. Meyendorf, and N. Mrad, "The role of data fusion in predictive maintenance using digital twin," *AIP Conf. Proc.*, vol. 1949, no. 1, 2018, Art. no. 020023.

[9] R. Martinez-Velazquez, R. Gamez, and A. El Saddik, "Cardio twin: A digital twin of the human heart running on the edge," in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Jun. 2019, pp. 1–6.

[10] P. Barbiero, R. V. Torné, and P. Lió, "Graph representation forecasting of patient's medical conditions: Towards a digital twin," 2020, *arXiv:2009.08299*.

[11] S. Huang, G. Wang, Y. Yan, and X. Fang, "Blockchain-based data management for digital twin of product," *J. Manuf. Syst.*, vol. 54, pp. 361–371, Jan. 2020.

[12] S. Sadjina, S. Skjong, A. Pobitzer, L. T. Kyllingstad, R.-J. Fiskerstrand, S. Torben, and J. D. Granholt, "Seismic RTDT: Real-time digital twin for boosting performance of seismic operations," in *Proc. Int. Conf. Offshore Mech. Arctic Eng.*, vol. 58844. New York, NY, USA: American Society of Mechanical Engineers, 2019, Art. no. V07AT06A040.

[13] Y. Zheng, R. Lu, Y. Guan, S. Zhang, and J. Shao, "Towards private similarity query based healthcare monitoring over digital twin cloud platform," in *Proc. IEEE/ACM 29th Int. Symp. Quality Service (IWQOS)*, Jun. 2021, pp. 1–10.

[14] L. Ismail and H. Materwala, "Blockchain paradigm for healthcare: Performance evaluation," *Symmetry*, vol. 12, no. 8, p. 1200, Jul. 2020.

[15] United States Department of Health and Human Services. *Notice to the Secretary of HHS Breach of Unsecured Protected Health Information*. Accessed: Apr. 8, 2022. [Online]. Available: https://ocrportal.hhs.gov/ocr/breach/breach_report.jsf

[16] World Health Organization. (2020). *Score Global Report on Health Data Systems and Capacity*. Accessed: Apr. 8, 2022. [Online]. Available: https://www.who.int/data/stories/score-global-report-2020—a-visual-summary

[17] T.-T. Kuo and L. Ohno-Machado, "ModelChain: Decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks," 2018, *arXiv:1802.01746*.

[18] *Health Information Privacy*. Accessed: Mar. 22, 2022. [Online]. Available: https://www.hhs.gov/hipaa/index.html

[19] *General Data Protection Regulation (GDPR)*, Accessed: Mar. 22, 2022. [Online]. Available: https://gdpr-info.eu/

[20] A. Bécue, E. Maia, L. Feeken, P. Borchers, and I. Praça, "A new concept of digital twin supporting optimization and resilience of factories of the future," *Appl. Sci.*, vol. 10, no. 13, p. 4482, Jun. 2020.

[21] M. Schluse, M. Priggemeyer, L. Atorf, and J. Rossmann, "Experimentable digital twins—Streamlining simulation-based systems engineering for industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1722–1731, Apr. 2018.

[22] K. Ding and P. Jiang, "RFID-based production data analysis in an IoT-enabled smart job-shop," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 1, pp. 128–138, Jan. 2018.

[23] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*. Cham, Switzerland: Springer, 2017, pp. 85–113.

[24] J. Li, F. Tao, Y. Cheng, and L. Zhao, "Big data in product lifecycle management," *Int. J. Adv. Manuf. Technol.*, vol. 81, nos. 1–4, pp. 667–684, 2015.

[25] D. Kiritsis, A. Bufardi, and P. Xirouchakis, "Research issues on product lifecycle management and information tracking using smart embedded systems," *Adv. Eng. Informat.*, vol. 17, nos. 3–4, pp. 189–202, Jul. 2003.

[26] A. Matsokis and D. Kiritsis, "An ontology-based approach for product lifecycle management," *Comput. Ind.*, vol. 61, no. 8, pp. 787–797, Oct. 2010.

[27] M. J. M. Chowdhury, M. S. Ferdous, K. Biswas, N. Chowdhury, and V. Muthukkumarasamy, "A survey on blockchain-based platforms for IoT use-cases," *Knowl. Eng. Rev.*, vol. 35, pp. 1–24, May 2020.

[28] C. M. Ezhilarasu, Z. Skaf, and I. K. Jennions, "Understanding the role of a digital twin in integrated vehicle health management (IVHM)," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 1484–1491.

[29] G. Ahmadi-Assalemi, H. Al-Khateeb, C. Maple, G. Epiphaniou, Z. A. Alhaboby, S. Alkaabi, and D. Alhaboby, "Digital twins for precision healthcare," *Cyber Defence in the Age of AI, Smart Societies and Augmented Humanity*. Cham, Switzerland: Springer, 2020, pp. 133–158.

[30] E. D'Auria, M. Abrahams, G. Zuccotti, and C. Venter, "Personalized nutrition approach in food allergy: Is it prime time yet?" *Nutrients*, vol. 11, no. 2, p. 359, Feb. 2019.

[31] P. Jouan and P. Hallot, "Digital twin: Research framework to support preventive conservation policies," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 4, p. 228, Apr. 2020.

[32] M. J. M. Chowdhury, M. S. Ferdous, K. Biswas, N. Chowdhury, A. S. M. Kayes, M. Alazab, and P. Watters, "A comparative analysis of distributed ledger technology platforms," *IEEE Access*, vol. 7, pp. 167930–167943, 2019.

[33] S. Jiang, J. Cao, H. Wu, Y. Yang, M. Ma, and J. He, "BlocHIE: A blockchain-based platform for healthcare information exchange," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, Jun. 2018, pp. 49–56.

[34] 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," *Sustain. Cities Soc.*, vol. 39, pp. 283–297, May 2018.

[35] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond bitcoin," *Appl. Innov.*, vol. 2, nos. 6–10, p. 71, 2016.

[36] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 2567–2572.

[37] M. S. Ferdous, M. J. M. Chowdhury, M. A. Hoque, and A. Colman, "Blockchain consensus algorithms: A survey," 2020, *arXiv:2001.07091*.

[38] Z. Alhadhrami, S. Alghfeli, M. Alghfeli, J. A. Abedlla, and K. Shuaib, "Introducing blockchains for healthcare," in *Proc. Int. Conf. Electr. Comput. Technol. Appl. (ICECTA)*, Nov. 2017, pp. 1–4.

[39] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, Jun. 2017, pp. 557–564.

[40] S. Nakamot, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Bus. Rev.*, p. 21260, Oct. 2008.

[41] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," Ethereum Foundation, Zug, Switzerland, White Paper, 2014, vol. 3, no. 37.

[42] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in *Proc. 2nd Int. Conf. Open Big Data (OBD)*, Aug. 2016, pp. 25–30.

[43] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "FHIRChain: Applying blockchain to securely and scalably share clinical data," *Comput. Struct. Biotechnol. J.*, vol. 16, pp. 267–278, Jul. 2018.

[44] I. Alom, R. M. Eshita, A. I. Harun, M. S. Ferdous, M. K. B. Shuhan, M. J. M. Chowdhury, and M. S. Rahman, "Dynamic management of identity federations using blockchain," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2021, pp. 1–9.

[45] C. Mohan, "State of public and private blockchains: Myths and reality," in *Proc. Int. Conf. Manage. Data*, Jun. 2019, pp. 404–411.

[46] E. Androulaki *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, Apr. 2018, pp. 1–15.

[47] *Sawtooth*. Accessed: Apr. 9, 2022. [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2018/01/Hyperledger_Sawtooth_WhitePaper.pdf

[48] *Corda*. Accessed: Apr. 9, 2022. [Online]. Available: https://www.r3.com/reports/corda-technical-whitepaper/

[49] J. Xu, K. Xue, S. Li, H. Tian, J. Hong, P. Hong, and N. Yu, "Healthchain: A blockchain-based privacy preserving scheme for large-scale health data," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8770–8781, Oct. 2019.

[50] Q. I. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14757–14767, 2017.

[51] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, Sep. 1997.

[52] D. Siegel. *Understanding the DAO attack*. Accessed: Apr. 8, 2022. [Online]. Available: https://www.coindesk.com/learn/2016/06/25/understanding-the-dao-attack/

[53] 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, Nov. 2020, Art. no. 8846667.

[54] Y. Liu, L. Zhang, Y. Yang, L. Zhou, L. Ren, F. Wang, R. Liu, Z. Pang, and M. J. Deen, "A novel cloud-based framework for the elderly healthcare services using digital twin," *IEEE Access*, vol. 7, pp. 49088–49101, 2019.

[55] P. Shamanna, B. Saboo, S. Damodharan, J. Mohammed, M. Mohamed, T. Poon, N. Kleinman, and M. Thajudeen, "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, Nov. 2020.

[56] P. Barbiero, R. V. Torné, and P. Lió, "Graph representation forecasting of patient's medical conditions: Toward a digital twin," *Frontiers Genet.*, vol. 12, Sep. 2021, Art. no. 652907.

[57] D. Petrova-Antonova, I. Spasov, I. Krasteva, I. Manova, and S. Ilieva, "A digital twin platform for diagnostics and rehabilitation of multiple sclerosis," in *Proc. Int. Conf. Comput. Sci. Appl.*, Cham, Switzerland: Springer, 2020, pp. 503–518.

[58] D. J. Rao and S. Mane, "Digital twin approach to clinical DSS with explainable AI," 2019, *arXiv:1910.13520*.

[59] H. Elayan, M. Aloqaily, and M. Guizani, "Digital twin for intelligent context-aware IoT healthcare systems," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 16749–16757, Dec. 2021.

[60] A. Croatti, M. Gabellini, S. Montagna, and A. Ricci, "On the integration of agents and digital twins in healthcare," *J. Med. Syst.*, vol. 44, no. 9, pp. 1–8, Sep. 2020.

[61] S. Hasavari and Y. T. Song, "A secure and scalable data source for emergency medical care using blockchain technology," in *Proc. IEEE 17th Int. Conf. Softw. Eng. Res., Manage. Appl. (SERA)*, May 2019, pp. 71–75.

[62] 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, Feb. 2018.

[63] F. Guo, Y. Mai, X. Zhao, X. Duan, Z. Fan, B. Zou, and B. Xie, "Yanbao: A mobile app using the measurement of clinical parameters for glaucoma screening," *IEEE Access*, vol. 6, pp. 77414–77428, 2018.

[64] M. A. Cassera, B. Zheng, D. V. Martinec, C. M. Dunst, and L. L. Swanström, "Surgical time independently affected by surgical team size," *Amer. J. Surg.*, vol. 198, no. 2, pp. 216–222, Aug. 2009.

[65] 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," 2017, *arXiv:1712.07809*.

[66] A. Wheelock, A. Suliman, R. Wharton, E. D. Babu, L. Hull, C. Vincent, N. Sevdalis, and S. Arora, "The impact of operating room distractions on stress, workload, and teamwork," *Ann. Surg.*, vol. 261, no. 6, pp. 1079–1084, 2015.

[67] C. Zhuang, J. Liu, and H. Xiong, "Digital twin-based smart production management and control framework for the complex product assembly shop-floor," *Int. J. Adv. Manuf. Technol.*, vol. 96, nos. 1–4, pp. 1149–1163, Apr. 2018.

[68] A. Shostack, *Threat Modeling: Designing for Security*. Hoboken, NJ, USA: Wiley, 2014.

[69] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, "GDPR-compliant personal data management: A blockchain-based solution," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1746–1761, 2019.

[70] Z. A. 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. 98893–98907, 2019.

[71] *Hyerledger Fabric*. Accessed: Apr. 1, 2022. [Online]. Available: https://www.hyperledger.org/use/fabric

[72] J. Sousa, A. Bessani, and M. Vukolic, "A Byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform," in *Proc. 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2018, pp. 51–58.

[73] O. W. Purbo, R. A. Aziz, and R. Herwanto, "Benchmark and comparison between hyperledger and MySQL," *Telkomnika*, vol. 18, no. 2, pp. 705–715, 2020.

[74] T. McConaghy, R. Marques, A. Müller, D. D. Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Granzotto. *BigchainDB 2.0 The Blockchain Database*. Accessed: May 10, 2022. [Online]. Available: https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf

[75] J. Benet, "IPFS–content addressed, versioned, P2P file system," 2014, *arXiv:1407.3561*.

[76] A. B. Haque, A. K. M. N. Islam, S. Hyrynsalmi, B. Naqvi, and K. Smolander, "GDPR compliant blockchains—A systematic literature review," *IEEE Access*, vol. 9, pp. 50593–50606, 2021.

[77] *Summary of the HIPAA Privacy Rule, Health Information Privacy*. Accessed: Mar. 22, 2022. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/index.html

[78] D. M. D. Simone, "When is accessing medical records a HIPAA breach?" *J. Nursing Regulation*, vol. 10, no. 3, pp. 34–36, 2019.

[79] S. U. Gardazi and A. A. Shahid, "Compliance-driven architecture for healthcare industry," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 5, pp. 568–577, 2017.

[80] R. N. Zaeem and K. S. Barber, "The effect of the GDPR on privacy policies: Recent progress and future promise," *ACM Trans. Manage. Inf. Syst.*, vol. 12, no. 1, pp. 1–20, Mar. 2021.

[81] L. Bufalieri, M. L. Morgia, A. Mei, and J. Stefa, "GDPR: When the right to access personal data becomes a threat," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Oct. 2020, pp. 75–83.

[82] A. Mahindrakar and K. P. Joshi, "Automating GDPR compliance using policy integrated blockchain," in *Proc. IEEE IEEE 6th Int. Conf. Big Data Secur. Cloud (BigDataSecurity) Int. Conf. High Perform. Smart Comput., (HPSC) IEEE Int. Conf. Intell. Data Secur. (IDS)*, May 2020, pp. 86–93.

[83] R. Herian, "Blockchain, GDPR, and fantasies of data sovereignty," *Law, Innov. Technol.*, vol. 12, no. 1, pp. 156–174, Jan. 2020.

[84] W.-C. Huang, L.-Y. Yeh, and J.-L. Huang, "A monitorable peer-to-peer file sharing mechanism," in *Proc. 20th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Sep. 2019, pp. 1–4.

[85] 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 *Proc. Int. Conf. Comput., Electron. Commun. Eng. (iCCECE)*, Aug. 2018, pp. 77–82.

SADMAN SAKIB AKASH received the B.Sc. degree in computer science from BRAC University, Bangladesh, where he is currently pursuing the M.Sc. degree with the Department of Computer Science and Engineering. His research interests include blockchain, cryptography, digital twin, and security.

MD SADEK FERDOUS (Member, IEEE) received the double master's degree in security and mobile computing from the Norwegian University of Science and Technology, Norway, and the University of Tartu, Estonia, and the Ph.D. degree in identity management from the University of Glasgow. He is currently working as an Associate Professor with the Department of Computer Science and Engineering, BRAC University, Dhaka, Bangladesh. He is also a Research Associate with the Centre for Financial Technology, Imperial College Business School. He has several years of experience of working as a Postdoctoral Researcher in different universities in different European and U.K.-funded research projects. He has published numerous research papers and book chapters in these domains in different books, journals, conferences, workshops, and symposiums. His current research interests include blockchain, identity management, trust management and security, and privacy issues in cloud computing, and social networks.