IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 9, 2019, accepted June 29, 2019, date of publication July 10, 2019, date of current version July 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927958

Edge-Based IoT Medical Record System: Requirements, Recommendations and Conceptual Design

AHMAD F. SUBAHI
Department of Computer Science, University College of Al Jamoum, Umm Al Qura University, Mecca 21421, Saudi Arabia
e-mail: afsubahi@uqu.edu.sa

ABSTRACT This paper presents a conceptual architecture, design, and recommendation for the IoT Edge-based healthcare management system. The suggested architecture aims at distributing the workload of system performance (electronic healthcare services), including monitoring, diagnosis, prediction, as well as managing and archiving medical data of patients across different points of the system (at the edge and on the cloud). The proposed system design consists of two main subsystems (one for monitoring tasks and another one for medical record management activities). Both subsystems interact with multiple kinds of database systems (SQL and NoSQL). Transformational-based system for data migration is presented as a contribution of this paper. Two styles of transformation compositions are considered in the architectural design of transformation agents.

INDEX TERMS IoT healthcare application, edge-based, system architecture, composition of model transformation, database migration, database design.

I. INTRODUCTION

Nowadays, with the rapid evolution of network technologies and the growth in the number of users who can connect, use and get access to data from anywhere and anyplace, carrying such paper-based personal healthcare data and exchanging it manually between different parties inside the healthcare domain have become an outdated habit. Various computer-based systems have emerged in different domains to support this transformation. In order to cope with this advancement, the demand for producing and adopting electronic healthcare and medical records has been increasing. Many investigations in this domain have been conducted over the last decade in favor of producing such medical information systems that deal with electronic patient records, as well as machine learning AI-based healthcare applications for disease diagnosis and bioinformatic data analysis.

Recently, the rise of cloud computing and fog computing has brought many benefits that have encouraged the healthcare sector to get involved and adopt these types of technologies, along with their benefits, for providing modernized health services. Like cloud computing, fog computing has also emerged and spread widely to meet the need of various requirements, such as reliability, security, latency and performance efficiency. Fog computing also brings a new strategy of solving problems and handling application, data processing at small edge servers, located at the network edge or in a remote data center.

Healthcare information systems is one significant type of system recently introduced with the utilization of cloud and fog computing technologies in different circumstances. Most of these systems share the same focus, whereby the emphasis is on digitalizing the current healthcare services and practices within hospitals and ER units. From a centralized cloud data storage to a distributed fog one, different works have discussed the adaptation of the type of database system and argued whether the benefits of using a relational database management system outweighs the benefits of using the trendy document-based or graph-based dynamic database (NoSQL) systems.

The main contribution of this work is to produce a new architectural design of an Edge-based IoT system that is applicable for human-centric applications, which can process massive amounts of data using model transformations. Model transformation (MT) is a mechanism of model-driven software engineering discipline that can be used here to share data

The associate editor coordinating the review of this manuscript and approving it for publication was Jafar A. Alzubi.

between different database schemas. Requirements, recommendations and design of an edge-based healthcare system that works with dynamic national repositories of populace health data and its applications in enhancing national health services are also introduced and discussed in this paper.

II. HISTORICAL OVERVIEW ON DEVELOPMENT OF DATABASE TECHNOLOGY

The development of database technology has occurred during several generations, started around the late 60s. With the Codd relational model in 1970 and the emergence of the first version of Oracle database system in 1979, relational database systems gained popularity for database design, where a 'table' concept that has fixed-length records is introduced to represent an independent type of a real-world entity [1]. Years later, many relational database systems were introduced by different vendors, such as Microsoft, MySQL, Postgres and Object-Oriented Database Management System.

In the 2000s, types of structured document-oriented database systems have appeared as an alternative database system approach that aims at solving common problems of traditional relational database systems, which are scalability and availability. Several data models, such as graph-based, document-based, XML-based, Column-based and Key-value based data models are introduced, by different vendors, with their own query languages and APIs. For instance, Cassandra and MongoDB are examples of NoSQL database systems that have column-based and document-based data models respectively. In Cassandra, the Cassandra Query Language (CQL) is introduced as a primary language for data manipulation and query, whereas JavaScript language is the language used for supported MongoDB query and data manipulating processes [2]–[4].

From an IoT perspective, it is worth mentioning that NoSQL systems are highly horizontally scalable, schema-free and very fast in query processing as they store deformalized and unstructured data and avoid using join operations in contrast to the traditional databasesŕthe so called OldSQL ones. All these features make NoSQL capable of handling big data in IoT applications [3].

III. DATABASE SYSTEMS IN IoT

Any IoT system depends on edge technology to process and provide data, while cloud computing requires a database management system to store and use processed data. As a result, IoT and edge-based systems need a centralized database, which is available on the cloud. Cloud-based database refers to a database system which resides on the cloud infrastructure and provided as database-as-a-service (DBaaS) [5]. Relational databases are not fully designed for cloud-based applications, even though it is important to have a backend database that supports the distributed nature of IoT and cloud/fog computing.

It is worthwhile noting that the cloud supports both the relational and non-relational databases. There are several factors that are considered when determining the better option for cloud database. This includes availability, maintainability, scalability, fault tolerance, security, interoperability, data portability and simplification of the queries. The work presented in [6] indicates that the IoT devices produce a lot of data which requires processing by the cloud systems. However, an important consideration is that a database management system ensures the data is managed in accordance with the outlines properties of the NoSQL databases [6]. As a matter of fact, [6] proposes the use of new perspectives which will ensure availability and flexibility in the storage and retrieval of data which can be achieved through further research in NoSQL systems.

On the other hand, another research, such as [7], outlines the current database management systems that are offered as a cloud service. According to [7], traditional relational database systems (SQL) are able to be integrated onto the cloud to facilitate IoT systems but they need to clarify data migration onto the cloud, parallelization of workload on the cloud, data security issues, and architecture design issues [7]. Additionally, relational databases can be implemented as a cloud database using SCALEDB, which is a storage engine that is compatible with MySQL. The framework works on the principle that nodes are clustered and each node has the ability to access the whole database (Figure 1).

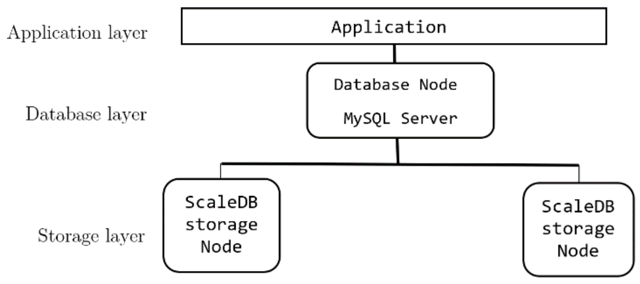

FIGURE 1. ScaleDB Architecture (cloud platform).

According to Figure 1, the storage layer is found on the cloud which has mirrored data across multiple sites. The database layer uses a MySQL in several nodes that all have access to the backend database and work in coordination allowing easy scalability. The application is the third layer, which is the place where the application server, web server and load balancer run. It is important to note that the main drawback of this architecture is that it does not solve the problem of need for distributed transactions as the transactions are still atomic [8].

Through supporting MySQL and other relational database management systems, it is now easier to use databases in IoT systems with the databases being hosted on the cloud. Relational databases affect major domains such as ecommerce, agriculture, smart cities, healthcare, and education. Taking the example of use in healthcare, wearable devices form the IoT platforms, which gather data from the patients such as glucose levels for diabetic patients.

However, a major problem is the high volumes of data gathered and the limited memory of the wearable devices [8]. In order to overcome this, DBaaS which is a cloud computing service, provides the required solution. According to [9], XML is used to record data from patients with wearable devices. The same data is then uploaded using simple relational database queries instead of complicated NoSQL databases. In this system, a load balancer, which is at the application layer of the ScaleDB architecture, improves performance through uploading the data to the cloud while saving the workload of the CPU, and storage, through load distribution. The system, therefore, can capture the physiological vitals of a patient and then the design of the system allows the data to be processed and stored efficiently on the cloud.

Regarding the findings, it can be said that when the data is uploaded, a controller is needed to upload the data into a database. In this case, it will be stored on the cloud [9]. The ScaleDB framework acts as the API, which links up IoT wearable devices to the processor, which is the cloud processing capabilities. Simple SQL queries can be run in a distributed environment thus ensuring that the data is always available and relevant to the patients who can check the relevant reports of their vitals.

IV. IRECOMMENDED SYSTEM DESIGN AND COMPONENTS

The recommended design of the system can be viewed as a collaboration of two main subsystems, namely, monitoring system and medical record management system. The monitoring system consists of a series of monitoring subsystem components that are collaborated to cover healthcare monitoring services based on IoT technology. Examples of these systems are addressed as:

- Body Temperature Monitoring System
- Blood Glucose Level Monitoring System
- Elderly Healthcare Monitoring system
- Heart Rate Monitoring System

Additionally, the medical record management subsystem is designed to work with a comprehensive cloud-based database storage that stores all health-related data about patients to support long-term decision making and medical data analysis activities. Examples of medical subsystems that feed the cloud storage directly can be listed as:

- Immunisation System
- Medication System
- Medical Imaging System

V. SYSTEM REQUIREMENTS

In this section, requirement specifications of each recommended subsystem are discussed in contrast to existing work.

A. REQUIREMENTS OF THE BODY TEMPRATURE MONITORING SUBSYSTEM

The body temperature monitoring system is part of the commonly used e-health systems that aim at monitoring health parameters associated to the body temperature of a patient. This kind of system can detect abnormal changes in temperature that require to be recorded and considered. In serious cases, the system notifies a doctor when a higher temperature than within the parameters of a certain range is detected.

Similar smart body temperature monitoring systems [10]–[13] have different kinds of temperature sensors that allow the measurement of the changes in human body temperature. LM35 in [11], TMP75 in [12] and SHT11 in [13] are examples of the types of sensors used for collecting body temperature data. These sensors normally get connected to a Raspberry Pi device and wireless network nodes.

In order to achieve a complete body temperature monitoring feature in the proposed system, some requirements must be considered as follows:

1- The system shall collect the body temperature data several times (e.g. every two hours) per day via an appropriate sensor.
2- The system shall detect any abnormal body temperature with respect to the normal body temperature and other parameters.
3- The system shall store recorded data into the healthcare monitoring database system in the edge.
4- The system shall transfer a summary of any interesting data into the medical record database on the cloud.
5- The system shall apply suitable artificial intelligence techniques to extract and predict useful knowledge.
6- In serious situations, the system shall notify the patient and the responsible third party about the case using prediction results, calculated in the edge.

B. REQUIREMENTS OF THE HEART RATE MONITORING SUBSYTEM

The main goal of the heart rate monitoring system is to collect heart rate information (usually raw data) based on pulses. The system serves the process of detecting any abnormal change in the heart rate and classify the patient's condition. When the rate falls between 60 and 110 bpm it can then be classified as a normal condition if the patient is in a rest position; however, if the heart rate goes beyond 110 bpm when the patient is resting, then his condition is classified as abnormal [14]. Many smart systems were designed to perform these jobs, such as [14]–[16]. In [15], a pulse rate sensor with a Raspberry Pi board and IR communication system is used to measure the heart rate in digital format using a 7414 invertor, whereas a more advanced system is presented in [16] that can transfer and store healthcare data on the cloud, as well as send SMS messages or notifications to a predefined individual or caregiver, associated to the patient, which might be a doctor, a nurse or even just a relative. The widely known Arduino Uno and Raspberry Pi 3 Model B are integrated together to support the single lead heart rate monitor (sensor).

In the proposed design, some requirements must be satisfied as follows:

1- The system shall measure the Beats Per Minute (BPM) rate of a patient.
2- The system shall present a digital result of heart rate using the finger of a patient to measure the BPM rate.
3- The system shall transfer the normal and abnormal recorded BPM rate to the healthcare monitoring database on the edge.
4- The system shall transfer a summarized BPM rate, of the normal and abnormal BPM rate, to the patient medical record database on the cloud.
5- In serious situations with abnormal BPM rate, the system shall notify the patient and/or the responsible third party about the case using AI prediction results, calculated at the edge.

C. REQUIREMENTS OF THE BLOOD GLUCOSE LEVEL MONITORING SUBSYSTEM

This kind of systems aims at helping, especially diabetic patients, to better self-monitor and manage their chronic condition. It is used to optimize treatment strategies by analyzing the effect of different external factors on diabetic patients such as diet, medication and daily exercise [17]. Monitoring glucose levels can help to understand the relation between blood glucose level, insulin, food and exercise [18]. Besides, the usage of glucose monitoring systems can be expanded to help non-diabetic patients in detecting abnormal changes of the sugar level in their blood when performing exertion or activity, and then reporting results.

Different smart systems were designed to perform similar jobs [19] and [20]. In [19] for example, a conceptual framework for disease diagnosis using IoT technologies is presented. Trajectory information of measurements, including glucose level, is observed over a time to introduce disease diagnosis schemes.

D. REQUIREMENTS OF THE ELDERLY HEALTHCARE MONITORING SUBSYSTEM

The elderly healthcare monitoring system aims at providing emergency medical assistance within a few hours of occurrence. Nowadays, it is widely recognized that elderly health monitoring and an emergency alert system is one of the main application areas of IoT and cloud computing and biomedical applications [21]. In other moderate cases, the system can offer computerized decision making support for community doctors, nurses and clinicians by monitoring and analyzing all activities of the elderly using wearable and comfortable all-in-one monitoring device(s). Data collected from these devices is also used to notify the hospital, nurse or even just an elderly's relative about abnormal situations. In the long run, the data can be used to construct a personalized model that helps to forecast the elderly wellness condition using suitable AI techniques [21]–[23].

Monitoring and tracking motion, blood oxygen content (SPO2), and human brain activity are examples of critical parameters that need to be considered in addition to the previously mentioned parameters covered in this research.

VI. OVERALL IoT ARCHITECTURE FOR HEALTHCARE SYSTEMS

This section discusses the required end-to-end Fog computing architecture as a suitable platform for IoT solutions. The suggested architecture consists of four layers: Healthcare sensing, Fog, Communication and Cloud layer as demonstrated in the following figure (Figure 2), whereas the detailed architecture is illustrated by Figure 4 at the end of this section.

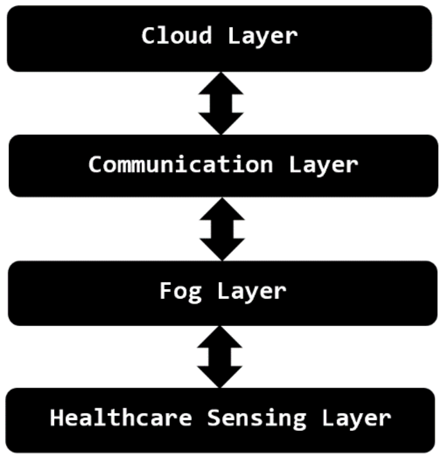

FIGURE 2. **The main layers of the suggested IoT edge-based system.**

A. HEALTHCARE SENSING LAYER

This layer consists of some networked smart devices and sensors that relay complete health-related information; capturing, processing and transferring to the next layer. Some of these devices and sensors regularly monitor various health parameters, including body temperature, heart rate, blood pressure, and any other parameters that come from specific medical devices such as, medical devices for diabetes or even x-ray scan. Other sensors are responsible for transferring data, notifications and reports to the related medical division or to other relevant parties to perform further actions.

B. FOG LAYER

The fog layer consists of a number of connected fog nodes. These nodes, called Edges, are responsible for receiving data from smart sensors of IoT devices, process this data locally, analyze it, then determine unusual health patterns and report to the most related party in the network to take an action. Consequently, each edge must have a communication, data storage and processing, and controlling units to achieve accelerated and low latency decision making. Figure 3 illustrates the architecture of edge nodes adopted in the system.

The communication unit acts as an interface between fog nodes and other components at different layers. Controllers, on the other hand, transfer any collected health information

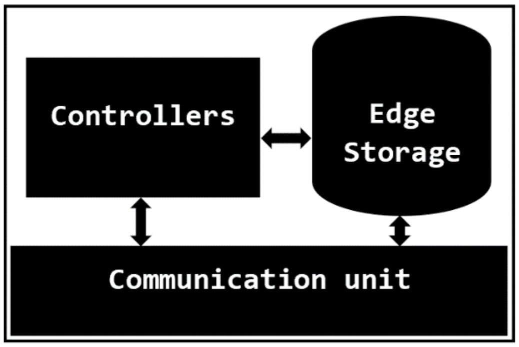

FIGURE 3. The Architecture of a node in the proposed edge-based system.

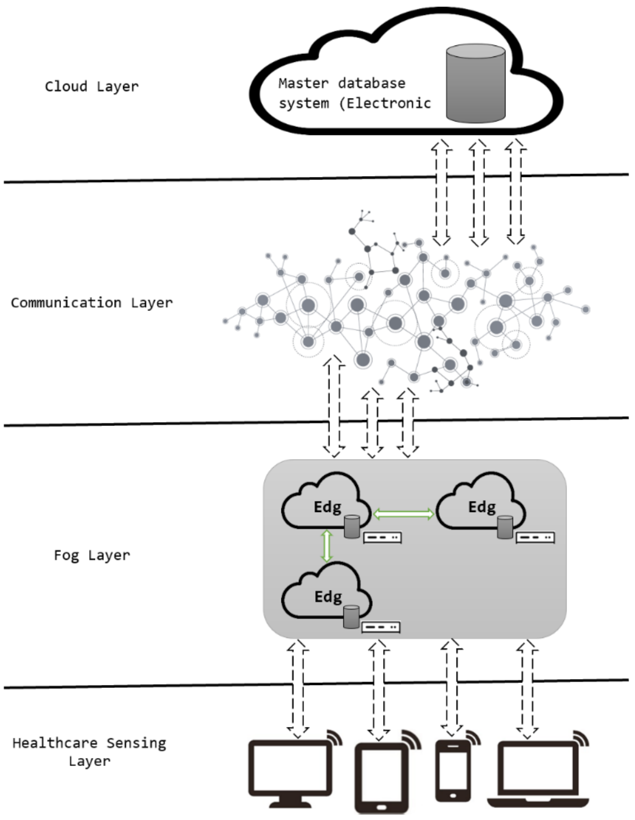

FIGURE 4. The detailed architecture of the suggested IoT Edge-based System.

from IoT sensors to other devices in the fog, to pass it to the higher layer of the proposed network architecture. While the data storage unit is a high-speed database system that can process massive amounts of data in a short period of time.

In order to obtain a scalable system, it is essential to support various types of end points. This requires supporting different network protocols for both wired and wireless connections within the network.

C. COMMUNICATION LAYER

This a typical networking layer that is responsible for managing the traffic of data and security, and other critical issues related to this type of networked system, between the fog network and the backend cloud. Some types of security threat that might be considered at this level are Man-in-the-middle, Spoofing, Confidentiality compromise and Replay attack.

D. CLOUD LAYER

A cloud data center is implemented at this layer, which contains the promising dynamic national repository of health data of the population. The comprehensive health record database system is constructed with three main features, namely, supporting data integration from multiple sources at the edges, long-term data analysis and processing for different points of care, and decision making support for various medical practitioners, patients and other parties. The data in the central cloud database contains data passed from fog for historical analysis of chronic diseases or other problems for early detection or treatment.

To achieve this need, a database system such as relational database or data lake might be adopted at this level of the system. There are some issues that can be left as open research questions for further investigation, such as:

- Which type of database system is the most suitable for this system (e.g. federated database or NoSQL cloud database)?
- Would it be feasible to have different database types at the fog level and cloud level, then a schema translating or transforming system designed and implemented?

VII. COMPONENTS DESIGN OF THE SYSTEM

In this section, the architectural design of the recommended IoT Edge-based system is discussed. There are two main services provided by the system, namely, healthcare monitoring and medical record management service. From that, the recommended system can be organized and designed into two main subsystems: healthcare monitoring subsystem and medical record management subsystem. The widely adopted UML Deployment diagram is used to illustrate the hardware components where all software components are deployed with respect to the overall architectural design proposed in this paper, especially for the sensing, fog and cloud storage layer.

A. ARCHITECTURAL DESIGN OF MONITORING SUBSYSTEMS

Figure 5 demonstrates a generic architectural design used in each monitoring subsystem. IoT devices collect health data and transfer it to the edge via a physical sensor and a controller component, whereas the collected data is stored at the edge side in an edge database and processed locally by data analysis and prediction software. The health summary and reports are displayed for the patient or doctor via mobile, tablet screen or traditional PC.

B. ARCHITECTURAL DESIGN OF MEDICAL RECORD MANAGEMENT SUBSYSTEM

Figure 6 illustrates a generic architectural design used in a medical record management subsystem. Each Healthcare unit, which is responsible for providing some specific health

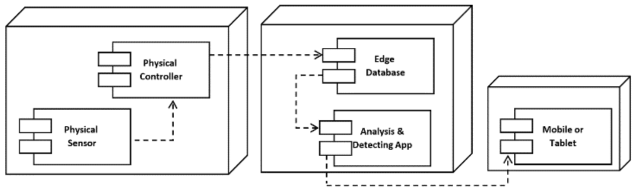

FIGURE 5. Architectural design of monitoring systems.

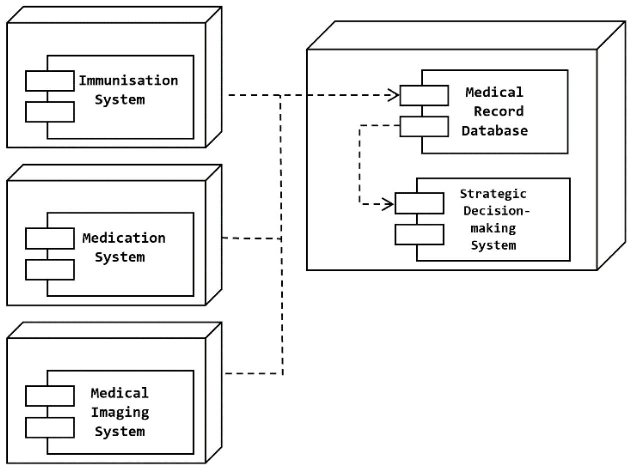

FIGURE 6. A centralized cloud-based storage with terminal systems.

and medical services to patients, is connected to a centralized back-end cloud data storage to store and retrieve patient health and medical information (medical record) in the longer term. The system also has a centralized decision making component that is responsible for analyzing and processing the archived patient health data and mining the data store to provide valuable information.

VIII. DATA MANAGEMENT IN IoT

In order to construct a complete IoT Edge-based system, various structures of data that is produced, consumed or even flows within different parts of the system must be considered; as this kind of system consists of different types of IoT devices, sensors that are connected and distributed across many geographical regions. These devices process and store massive amounts of real-time data to enable the system to deliver services at its terminal parts. Additionally, the majority of IoT Edge-based systems have centralized databases on the cloud that store long-term data and perform series of analysis and predicting tasks that support decision making. From that, in the proposed design, it is recommended to consider more than one database system that are described as follows:

A. TERMINAL DATABASES AT THE DEVICES AND EDGE

As discussed earlier in this report, the promising healthcare system is not a simple application that is used in a specific location, whereby it consists of a set of distributed subsystems that are connected and highly integrated. Each subsystem belongs to a distinguished healthcare unit that deals with a specific type of health data without the need to get access to all the central information of the patient's medical record.

This side of the database system is aiming at collecting and analyzing (processing) data in the most lightweight form with highest efficiency. Consequently, it is required to adopt a flexible data model in contrast to the traditional RDBMS. The nature of required schema of the data storage here must be dynamic. The dynamic data model is needed when dealing with data in machine-generated systems, which is offered in NoSQL database. There are special characteristics that must be supported by the database system implemented at device or edge side, such as the performance and memory efficiency.

Data (health monitoring data) will be collected frequently in different types from monitoring subsystems, and partial data from the medical record of a patient might be required by the healthcare units to perform data processing and decision making. These data processing tasks can be achieved locally or at the edge side without the need to transfer massive amounts of data over the network to the centralized cloud storage and retrieve it again later before making the decision.

1) RECOMMENDED DATABASE SYSTEMS

- NoSQL Document-based database such as Cassandra or MongoDB.
- NoSQL Graph-based database such as Neo4JDB.

B. CENTRALISED CLOUD-BASED STORAGE (ELECTRONIC MEDICAL RECORD)

There are various benefits of the relational data model in RDBMS, such as robustness and the ability to store massively structured data. Laboratory medication and scan results, which are not done frequently, are aimed to be stored and archived in this part of the database forming comprehensive information about patients' health (electronic medical record). The medical record of a patient is a type of historical archive that contains a summary of various historical attributes related to the patient's health, such as initial diagnoses of incurable diseases or infections and more.

The healthcare system stores the comprehensive patient health/medical information for long-term use, which is normally collected from different sources, such as medical units or device sensors, including personal details, address, demographics, progress notes, immunization table, laboratory results, radiology analysis results, past medical history, family history of chronic disease, vital signs, tobacco history, alcohol history, physiotherapy examination reports, blood pressure, diabetes level, heart rate data, complete blood count (CBC) results and more.

From that, it can be noticed that different information comes from different medical units in multiple formats and structures. Thus, it might be beneficial to distribute partial portions of the electronic medical record as dynamic schemas over the different medical units/locations and let the subsystem at each unit perform the data processing locally or at the edge side. Then, these data must be migrated, occasionally, into the centralized patients' data storage on the cloud.

The centralized database system needs to have the capability to integrate and interchange data between the different dynamic data schemas at the subsystems and the edge. This helps to keep the medical records updated with the most recent health information, adequate to support any further decision-making task that may require a combination of data from different sources over the network.

1) RECOMMENDED DATABASE SYSTEMS

- Oracle Cloud Database System
- Oracle MySQL Cloud Database System
- Microsoft Azure SQL Database

In balance, from the previous discussion, the database systems in the recommended IoT design must be constructed with three main features, namely, data integration from multiple sources, data capturing at the point of care, and decision-making support for medical practitioners.

IX. DESIGN OF THE ELECTRONIC MEDICAL RECORDS DATABASE

Principles and strategies, or data modeling, are considered during the construction process of a comprehensive medical record database. Traditional hospitals consist of several wards (departments). Each type of ward provides different kinds of healthcare services for patients. On one hand, some of the wards provide general health services and are called acute care wards, such as emergency ward/room (ER), general surgery unit, or physiotherapy center. On the other hand, there are other kinds of wards that provide specialized services to patients, such as cardiology units, elderly services unit, gastroenterology unit, tumors center and chronic treatment unit. Besides these, there are some units that commonly exist in every hospital to serve various kinds of wards, such as radiology unit, pharmacy and clinical laboratory.

From that, the overall database can be designed as a number of clusters based on the main operations and required data that is used in each ward.

A. PRLIMINARY DEFINITIONS

Before writing the recommended textual database schema, a list of mathematical definitions is presented in this subsection based on relational calculus and First-Order Predicate Logic (FOPL):

- A database DB is a set of all database schemas, where $DB := \{S_1, S_2, \ldots, S_n\}$. A database may contain at least one schema and each schema S is corresponded to a hospital ward or department that provides healthcare/medical services to patients.
- A relational database schema S is a set of relations (tables), where $S := \{R_1, R_2, \ldots, R_n\}$. Each relation R in S represents a unique entity in the universe of discourse (UoD).
- A relation (Table) is a set of n-tuples T, where $T := \{t_1, t_2, \ldots, t_n\}$. Each element in T represents a unique record in a relation R.
- A tuple t is a function on a finite set of attributes A from names to atomic values v of each element, where $A := \{a_1, a_2, \ldots, a_n\}$ and represents all columns in a table (relation). The function t is $t(A) = \{(a, v) \cdot a \in A \wedge (a, v) \in t\}$.
- The n-tuple of a relation R is denoted by $t(R) :=< v_1, v_2, \ldots, v_n >$ as each value vv corresponds to a specific attribute a.
- Each attribute a has a name that represents a specific role played in the UoD, and it is expressed as $Dom(a)$. The number of attributes n in each relation R is called the degree of R.

B. COMPREHENSIVE MEDICAL RECORD (CMR) RELATIONAL DATABASE SCHEMA

After applying the set of definition rules, presented in section A, to construct and normalize the relational data model of the system, the database schema is formulated and a snapshot of it is expressed in the following subsections due to the limitation of the size of this paper.

1) BASIC INFORMATION SCHEMA

The basic information schema consists of five relations (tables) that are defined as follows:

$$\begin{aligned}
&PATIENTMEDFILE(PID, FName, LName, DoB,\\
&Gender, MaritalStatus, BloodGroup, PersonalIDPic)\\
&ADDRESS(PID, Line1, Line2, PostCode, City)\\
&CONTACT(PID, MobileNo, HomePhone, WorkPhone,\\
&Email, Fax)\\
&NEXTOFKEN(PID, NOKID, NOKName,\\
&NOKMobileNo, NOKRelationType)\\
&DEMOGRAPHICDETAIL(PID, Religion, Race,\\
&Education, IsTwin, Nationality)
\end{aligned}$$

The detailed definition of the attributes of the *PATIENTMEDFILE* relation is described as follows:

$$\begin{aligned}
Dom\,(PID) &:= Patient\ ID;\\
Dom\,(FName) &:= First\ name\ of\ the\ patient;\\
Dom\,(LName) &:= Last\ name\ of\ the\ patient;\\
Dom\,(Gender) &:= Gender\ of\ the\ patient;\\
Dom\,(MatirtalStatus) &:= the\ marital\ status\ of\ the\ patient;\\
Dom\,(BloodGroup) &:= the\ group\ type\ of\ patient\ blood;\\
Dom\,(PersonalIDPic) &:= a\ personal\ photo\ of\ the\ patient;
\end{aligned}$$

2) DRUG HISTORY SCHEMA

The drug history schema consists of two relations (tables) that are defined as follows:

$$\begin{aligned}
&ALCOHOLHISTORY(PID, DrinkType, StartDate,\\
&EndDate, SideEffect)\\
&TOBACOOHISTORY(PID, PacksNo, WakeUp,\\
&SomkingStartDate, SmokingEndDate, SideEffict)
\end{aligned}$$

The detailed definition of the attributes of the *ALCOHOLHISTORY* relation is described as follows:

$$\begin{aligned} Dom\,(PID) &:= Patient\ Id;\\ Dom\,(DrinkType) &:= Type\ of\ the\ alcoholic\ drink;\\ Dom\,(StartDate) &:= date\ of\ start\ drinking\ alcohol;\\ Dom\,(EndDate) &:= date\ of\ end\ drinking\ alcohol;\\ Dom\,(SideEffect) &:= the\ side\ effect\ of\ drinking\ alcohol; \end{aligned}$$

3) FAMILY HISTORY SCHEMA

The family history schema consists of three relations (tables) that are defined as follows:

$$\begin{aligned} &FAMILYHISTORY\,(PID, RID, RelativeTYPE, RDoB,\\ &IllinessID, InfectionYear)\\ &PASTMEDICALHISTORY\,(IllinessID, IllinessName)\\ &PASTMEDHISTORYTEST\,(PID, IllinessID,\\ &IllinessTestDate) \end{aligned}$$

4) OPERATIONS SCHEMA

The operations schema consists of three relations (tables) that are defined as follows:

$$\begin{aligned} &OPERATION\,(PID, OPID, OPType, Date, Dignsis, Notes)\\ &OPERATIONTYPE\,(OPID, OPName)\\ &OPERATIONTYPEDETAIL\,(OPID, OPSubTypeID,\\ &OPSubTypeName) \end{aligned}$$

5) ALLARGIES SCHEMA

The allergies schema consists of two relations (tables) that are defined as follows:

$$\begin{aligned} &ALLARGIES\,(PID, AID, ADate, Notes)\\ &ALLARGIESTYPE\,(AID, AName) \end{aligned}$$

6) IMMUNSATIONS SCHEMA

The immunsations schema consists of two relations (tables) that are defined as follows:

$$\begin{aligned} &IMMUNZATION\,(IMMID, IMMName)\\ &IMMUNZATIONTESTS\,(PID, IMMID, TestDate, PAge) \end{aligned}$$

7) MEDICATIONS SCHEMA

The medications schema consists of two relations (tables) that are defined as follows:

$$\begin{aligned} &MEDICATIONS\,(PID, MEDID, MEDDATE, Dose, Taken,\\ &Reasons, StartTakenDate, EndTakenDate)\\ &MEDICATIONTYPE\,(MEDID, MEDName,\\ &SeriousSideEffect) \end{aligned}$$

X. TRANSFORMATION STRATEGY OF DATA MIGRATION

Model transformation is a key concept in Model-Driven Engineering (MDE) that aims at automating the construction and modification of system artifacts (models) or generating executable code. Many transformation strategies in the domain of database schema generation have been introduced in various works, such as MySQL schema generation in [24], and NoSQL code generation and data migration in [25], [26], [28] and [29]. Additionally, another key concept of MDE, which is Domain-Specific Language (DSL), is also adopted in some data migration approaches, such as the Data Transformation Language used in [27] for migrating legacy data.

The novelty of the recommended architecture and design of the proposed IoT healthcare system is centered on the adaptation of appropriate bidirectional model translations as a complete data migrating solution to support the transferring of data between the subsystems and database schemas mentioned earlier in this paper and get the benefit of both kinds of database systems in the proposed architecture. Two styles of transformation compositions are discussed as a part of the contribution of this work.

The following bullet points address the main recommendations that must be considered when designing the transformation system and rules:

- Transformation code must be concise and short
- Transformation rules must be distributed in modularized transformation components (agents).
- Adopt a rich transformation language that is powerful enough to represent the semantics of the transformation rules involved.
- Introduce a suitable query language that is able to execute SQL query on NoSQL data storage.

XI. RECOMMENDED DESIGN OF THE MAPPING AGENTS

Before diving too deep into the internal design of agents, basic terminologies must be clearly defined to be distinguishable to the reader. The notation of the widely known First-Order Logic (FOL) is used to formalize the definitions as:

A. TRANSLATION AGENT

The translation agent is a component of the transformation system that is responsible for translating a source schema/model that belongs to a specific metamodel into another kind of target schema/model that belongs to another metamodel.

$$\begin{aligned} &\forall x : SOURCE \in M : METAMODEL\\ &\forall y : TARGET \in N : METAMODEL\\ &\forall T : TRANSLATOR \cdot T : M \rightarrow N\\ &\qquad \Rightarrow T(x) := y \end{aligned}$$

B. TRANSFORMATION AGENT

The transformation agent is a component of the transformation system that is responsible for transforming a source code/schema/model into an optimized form at a different level of abstraction, where both source and target belong to the same metamodel.

$$\forall x : SOURCE \in M, M' : METAMODEL$$
$$\forall T : TRANSFORMAR \cdot T : M \rightarrow M'$$
$$\Rightarrow T(x) := x' \in M$$

XII. COMPOSITIONAL STRATEGY OF TRANSFORMATION AGENTS

The structure of the transformational agents is designed based on two compositional strategies, namely, linear and hierarchical.

A. LINEAR COMPOSITION

It is recommended to adopt a multiple-level (linear) transformational approach in order to achieve a solution that can map between a wider range of schemas. Thus, it can cover types of translations between different database vendors, SQL vendors like MySQL and Oracle, and NoSQL vendors like MongoDB and Cassandra, and even the graph data model such as Neo4J database.

From this point of view, the demand for introducing the more common and less technical intermediate representation (IR) model using a textual Domain-Specific Modeling Language (DSML) has emerged. This common IR model can support the productivity of transformation agents, reducing their complexity and optimizing their performance.

The following UML Package diagram (Figure 7) illustrates the architecture of the (two-step) linear translating framework that maps NoSQL document-based data model into the SQL relational model.

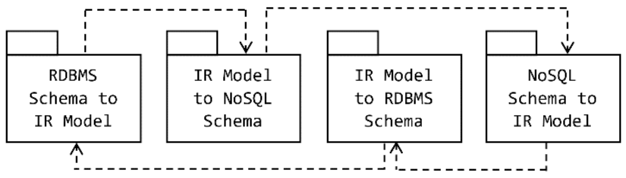

FIGURE 7. A linear composition of transformational agents.

B. HIERARACAL COMPOSITION

The hierarchical composition is applied to the internal design of each translation system. Where the translation rules are distributed across different translation agents, based on the structure of the target schema (model). It is worth mentioning that it is still an open question whether the benefit of considering the structure of the target schema when designing the translation system outweighs the structure of the source one.

To illustrate the hierarchical composition in the internal design of the transformation agent, the case of mapping NoSQL graph DB structure into RDBMS schema is considered via GraphToRDBMS translation system. This system consists of several translation agents as shown in the following figure (Figure 8).

It is worth mentioning that defining the mapping between NoSQL schemas, such as graph- or document-based and RDBMS ones is not a straightforward task. Based on the transformation language and framework used (declarative/imperative), the mapping process normally involves a series of transformation iterations to form the final structure of entities, its properties and data integrity and constraints, including relationships. The detailed design and implementation of the suggested transformation framework is out of the scope of this paper. Further works, including an intensive survey on possible model transformation languages and frameworks for database generation, are required to decided which DSML might be used with which model transformation approach in the proposed system.

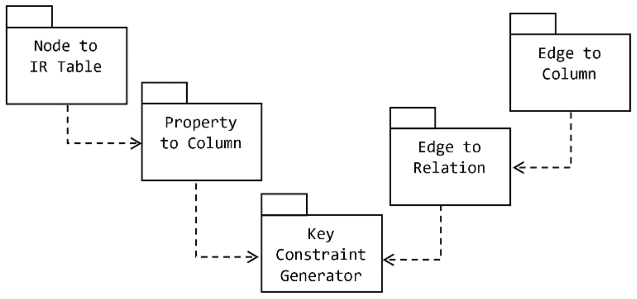

FIGURE 8. A hierarchal composition of transformational agents.

XIII. CONCLUSION

Requirements, recommendation and architectural design of an edge-based healthcare system that works with dynamic healthcare crowd data repositories are introduced and discussed in this paper. An IoT Edge-based healthcare system is introduced with two subsystems, namely monitoring subsystem and medical record management subsystem. Additionally, two kinds of data models are considered in the work (relational and NoSQL data model). An architectural design of a recommended transformational system is also introduced in this work with two styles of compositions (linear and hierarchical).

Based on the work presented in this paper, further research directions might be formulated. Investigation and implementation of the model-driven engineering side, including the internal design of mapping rules, adopted model transformation language, designing the language of the intermediate common data model are interesting dimensions of research. Additionally, the investigation and implementation of the cloud technology and networking is another interesting dimension that includes internal design, protocols, security and performance.

REFERENCES

[1] E. F. Codd, "Extending the database relational model to capture more meaning," *ACM Trans. Database Syst.*, vol. 1, no. 2, pp. 397–434, Dec. 1979.

[2] V. Abramova and J. Bernardino, "NoSQL databases: MongoDB vs cassandra," in *Proc. Int. C* Conf. Comput. Sci. Softw. Eng.*, Jul. 2013, pp. 14–22.

[3] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of 'big data' on cloud computing: Review and open research issues," vol. 47, pp. 98–115, Jan. 2015.

[4] A. Nayak, A. Poriya, and D. Poojary, ''Type of NOSQL databases and its comparison with relational databases,'' *Int. J. Appl. Inf. Syst.*, vol. 5, no. 4, pp. 16–19, Mar. 2013.

[5] T. A. Phan, J. K. Nurminen, and M. Di Francesco, ''Cloud databases for Internet-of-Things Data,'' in *Proc. IEEE Int. Conf. Internet of Things (iThings), IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom)*, Sep. 2014, pp. 117–124.

[6] Kalay, M. U. ''Database system suggestions for the Internet of Things (IoT) systems,'' *Mugla J. Sci. Technol.*, vol. 4, no. 1, pp. 46–52, 2018. doi: 10.22531/muglajsci.418488.

[7] Y. Januzaj, J. Ajdari, and B. Selimi, ''DBMS as a cloud service: Advantages and disadvantages,'' *Procedia Social Behavioural Sci.*, vol. 195, pp. 1851–1859, Jul. 2015. doi: 10.1016/j.sbspro.2015.06.412.

[8] W. Shehri, ''Cloud database database as a service,'' *Int. J. Database Manage. Syst.*, vol. 5, no. 2, p. 1, 2013.

[9] H.-T. Chang and T.-H. Lin, ''A database as a service for the healthcare system to store physiological signal data,'' *PLoS ONE*, vol. 11, no. 12, 2016, Art. no. e0168935. doi: 10.1371/journal.pone.0168935.

[10] M. S. Jassas, A. A. Qasem, and Q. H. Mahmoud, ''A smart system connecting e-health sensors and the cloud,'' in *Proc. IEEE 28th Can. Conf. Elect. Comput. Eng. (CCECE)*, May 2015, pp. 712–716.

[11] N. D. Agham, V. R. Thool, and R. C. Thool, ''Mobile and Web based monitoring of patient's physiological parameters using LabVIEW,'' in *Proc. Annu. IEEE India Conf. (INDICON)*, Dec. 2014, pp. 1–6.

[12] N. Jagtap, J. Wadgaonkar, and K. Bhole, ''Smart wrist watch,'' in *Proc. IEEE Students' Conf. Elect., Electron. Comput. Sci. (SCEECS)*, Mar. 2016, pp. 1–6.

[13] A. Javadpour, H. Memarzadeh-Tehran, and F. Saghafi, ''A temperature monitoring system incorporating an array of precision wireless thermometers,'' in *Proc. Int. Conf. Smart Sensors Appl. (ICSSA)*, May 2015, pp. 155–160.

[14] H. Kemis, N. Bruce, W. Ping, T. Antonio, L. B. Gook, and H. J. Lee, ''Healthcare monitoring application in ubiquitous sensor network: Design and implementation based on pulse sensor with arduino,'' in *Proc. 6th Int. Conf. New Trends Inf. Sci., Service Sci. Data Mining (ISSDM)*, Oct. 2012, pp. 34–38.

[15] R. Kumar and M. P. Rajasekaran, ''An IoT based patient monitoring system using raspberry Pi,'' in *Proc. Int. Conf. Comput. Technol. Intell. Data Eng. (ICCTIDE)*, Jan. 2016, pp. 1–4.

[16] A. Rahman, T. Rahman, N. H. Ghani, S. Hossain, and J. Uddin, ''IoT based patient monitoring system using ECG sensor,'' in *Proc. Int. Conf. Robot., Elect. Signal Process. Techn. (ICREST)*, Jan. 2019, pp. 378–382.

[17] G. Alfian, M. Syafrudin, M. Ijaz, M. Syaekhoni, N. Fitriyani, and J. Rhee, ''A personalized healthcare monitoring system for diabetic patients by utilizing BLE-based sensors and real-time data processing,'' *Sensors*, vol. 18, no. 7, p. 2183, Jul. 2018.

[18] *Blood Glucose Monitoring. Diabetes Australia*. Accessed : Apr. 19, 2019. [Online]. Available: https://www.diabetesaustralia.com.au/blood-glucose-monitoring

[19] H.J. La, ''A conceptual framework for trajectory-based medical analytics with IoT contexts,'' *J. Comput. Syst. Sci.*, vol. 82, no. 4, pp. 610–626, Jun. 2016.

[20] T.N. Gia, M. Ali, I. B. Dhaou, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, ''IoT-based continuous glucose monitoring system: A feasibility study,'' *Procedia Comput. Sci.*, vol. 109, pp. 327–334, Jan. 2017.

[21] S.J. Park, M. Subramaniyam, S. E. Kim, S. Hong, J. H. Lee, C. M. Jo, and Y. Seo, ''Development of the elderly healthcare monitoring system with IoT,'' in *Advances in Human Factors and Ergonomics in Healthcare*. Cham, Switzerland: Springer, 2017, pp. 309–315.

[22] L. Yu, W. M. Chan, Y. Zhao, and K.-L. Tsui, ''Personalized health monitoring system of elderly wellness at the community level in Hong Kong,'' *IEEE Access*, vol. 6, pp. 35558–35567, 2018.

[23] I. Almarashdeh, K. M. Alsmadi, T. Farag, S. A. Albahussain, U. A. Badawi, N. Altuwaijri, H. Almaimoni, F. Asiry, S. Alowaid, M. Alshabanah, D. Alrajhi, A. A. Fraihet, and G. Jaradat, ''Real-time elderly healthcare monitoring expert system using wireless sensor network,'' *Int. J. Appl. Eng. Res.*, vol. 13, no. 6, pp. 3517–3523, Feb. 2018.

[24] A. F. Subahi and A. J. Simons, ''A multi-level transformation from conceptual data models to database scripts using Java agents,'' in *Proc. Int. Workshop Composition Evol. Model Transf.* London, U.K.: Kings Collage, 2011, pp. 1–7.

[25] I. Zečević and P. Bjeljac, ''Model driven development of hybrid databases,'' in *Proc. 7th Int. Conf. Inf. Soc. Technol. (ICIST)*, 2017, pp. 154–158.

[26] A. H. Chillón, D. S. Ruiz, J. G. Molina, and S. F. Morales, ''A model-driven approach to generate schemas for object-document mappers,'' *IEEE Access*, vol. 7, pp. 59126–59142, 2019.

[27] P. Carreira and H. Galhardas, ''Efficient development of data migration transformations,'' in *Proc. SIGMOD Conf.*, Mar. 2004, pp. 915–916.

[28] A. A. Mahmood, ''Automated algorithm for data migration from relational to NoSQL databases,'' *ALNAHRAIN J. Eng. Sci.*, vol. 21, no. 1, pp. 60–65, 2018.

[29] M. Hanine, A. Bendarag, and O. Boutkhoum, ''Data migration methodology from relational to NoSQL databases,'' *World Acad. Sci., Eng. Technol., Int. J. Comput., Elect., Automat., Control Inf. Eng.*, vol. 1, no. 2, pp. 2369–2373, Feb. 2016.

AHMAD F. SUBAHI received the B.Sc. degree in computer science from King Abdulaziz University (KAU), Jeddah, Saudi Arabia, in 2002, the first M.Sc. degree in information technology from the Queensland University of Technology, Brisbane, Australia, in 2008, the second M.Sc. degree in advanced computer science and the Ph.D. degree in computer science from The University of Sheffield, U.K., in 2010 and 2015, respectively. Since 2015, he has been appointed as an Assistant Professor in computer science with the Computer Science Department, University College of Al Jamoum (JUC), Umm Al-Qura University (UQU), Mecca, Saudi Arabia. He is the former Head of the department (2017–2018). He is currently the Vice Dean of development and entrepreneurship at JUC. His research is focused on software engineering and computer science field. His specific research areas include model-driven engineering, domain-specific modeling languages, code generation and programming languages design, software systems architecture and design, database systems, robotics, and the IoT systems engineering.