Architectures and Patterns for Moving Towards the Use of High-Frequency, Low-Fidelity Data in Healthcare

By

Peng Zhang

Dissertation
Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
in
Computer Science

September 30, 2018

Nashville, Tennessee

Approved:

Jules White, Ph.D.

Ted Bapty, Ph.D.

Douglas C. Schmidt, Ph.D.

Shelagh A. Mulvaney, Ph.D.

Aniruddha Gokhale, Ph.D.

To my beloved parents and grandmother, who are my strength in everything I do.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincerest respect and gratitude to my advisor, Prof. Jules White, my mentor, Prof. Douglas C. Schmidt, and my co-adviser, Prof. Ted Bapty, for their continuous encouragement and guidance and generous financial support throughout my Ph.D. journey. Dr. White has helped me tremendously from providing research directions to giving me many opportunities to work with industry companies to practice my research. Dr. Schmidt has offered me so many valuable insights on research projects and papers and also future career paths. Dr. Bapty gave me the opportunity to pursue my degree at Vanderbilt University and trusted me with my research decisions, to which I am very thankful for.

I would like to thank Prof. Shelagh Mulvaney and Prof. Anirddh Gokhl for serving on my Ph.D. committee. They have guided my research in projects and provided insightful feedback on my work. I could not have asked for a more caring and supportive committee. I also wish to thank Dr. Zsolt Lattmann for his encouragement and support and Mr. Gunther Lenz for his inspirations and mentorship during my most fruitful summer internship at Varian Medical Systems.

I would like to express my special thanks to my role models, best friends, and parents, Mrs. Guizhi Ren and Mr. Liqun Zhang for their unconditional love, continuous emotional and moral support, and everything they have done for me that cannot be summarized in this short acknowledgement. I want to also thank my dearest family and friends, Michael and Rita Cochrane, Shunxing Bao, Philip and Kristine Scatena, Lei Chen, Ed and Janice Nichols, Kim Pham, Monica Hedda, Kate Brady, Hakan Tunc, Ajay Chhokra, Yi Li, Di Yao, Fangzhou Sun, and many others who have always cheered me on during both my ups and downs.

ABSTRACT

The U.S. healthcare data has undergone significant transformations as computerized technologies have evolved in recent decades. Data trends in healthcare have transitioned from less frequent, higher-fidelity provider-documented electronic health records (EHR) to more frequent, lower-fidelity patient and device-generated data. In particular, prevalent Internet of Things (IoT) devices and apps collect enormous amounts of information associated with individuals health statuses, physical activities, and environmental triggers to chronic conditions. As a result, health-related data generated from IoT devices today is now exceeding EHR data in terms of volume and frequency. It is important, however, to integrate these data into healthcare decisions since they reflect various aspects of citizens lifestyles and well-being in a comprehensive and continuous manner. Given these trends, a key problem facing heathcare researchers and practitioners is how to successfully migrate towards the use of the high-frequency, low-fidelity (HFQ) data in the healthcare domain.

Addressing this problem requires research that focuses on the following issues: (1) how to scalably extract insights from large volumes of health data, (2) how to integrate and share HFQ data along with learned insights from those data, and (3) after an integrated health system is created, what methods and techniques are needed to evolve it to adopt more efficient technology or perform upgrades and updates as needed. This dissertation presents software architectures and patterns targeting these issues. First, we propose a machine learning based filtering architecture for drawing insights from HFQ data at scale. Second, we describe a data sharing framework based on distributed ledger technologies (DLT) to address technical requirements defined by the Office of the National Coordinator for Health IT (ONC). Lastly, we document a design pattern sequence for effectively designing and maintaining a DLT-enabled healthcare data sharing system in a secure and evolvable manner.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Emerging Trends in Healthcare Data

The U.S. healthcare data has undergone significant transformations as computerized technologies advanced over the last several decades. Prior to the adoption of computerized medical records systems in the early 1990s [1], health data predominantly took shape of paper-based records and charts, which were often poorly legible and potentially contributing to medical errors [2]. Later with computers and the Internet introduced in hospitals and other healthcare facilities in the 1990s [3], medical records largely transitioned from paper-based to electronic health records (EHR). As native EHR systems in large medical centers matured, physicians were able to directly interact with them in early 2000s, which promoted widespread adoptions of EHRs among hospitals and physician offices. More health data became electronically documented, with many EHRs becoming commercialized to provide more advanced features and enhanced experience for providers [3]. However, due to the lack of common standards and nomenclature for representing data and identifying individuals, many EHR systems became too custom-crafted for different practices to interoperate and exchange clinical messages from one medical setting to another. In 2009, the The American Reinvestment & Recovery Act (ARRA) provisioned the Health Information Technology for Economic and Clinical Health (HITECH) Act [4] to incentivize the ”meaningful use” of EHRs, promoting the use of certified EHRs with more consistent clinical exchange standards. The goal was to improve care coordination by enforcing more efficient standards and thus to bridge the apparent interoperability gaps [5]. By 2014, 96.9% of non-federal acute care hospitals adopted a certified EHR system [6], incorporating a variety of data, such as lab results, billing information, allergies, and radiology images, etc.

Meanwhile, the Internet of Things (IoT) has permeated through daily functions and activities of both communities and individuals today, with embedded sensors, smartphones, and wearables being the most prevalent. These devices continuously collect data in real-time, and many contain pertinent information regarding individuals' lifestyle and wellbeing. For instance, given that 77% of Americans own a smartphone [7] and that more than 165,000 apps were available on Apple iTunes that target health and wellness alone [8], smartphone and wearable users are contributing to an enormous amount of patient-generated health data (PGHD). Some examples of PGHD include heart rate, weight, sleep and fitness data, carbohydrates intake, so on and so forth. Moreover, sensors that are used for monitoring the environment, like air quality and humidity, can also be used to understand the effect of environmental stressors on certain chronic conditions, such as asthma, which is a very common respiratory disease [9].

Today, health data generated from IoT devices has far exceeded EHR data both in terms of volume and frequency. Nevertheless, the fidelity of these high-frequency data, in general, is much lower compared to EHR data that is documented by licensed healthcare professionals based on expert knowledge or obtained from certified medical equipment used in lab tests. As a general trend, health data has shifted from provider-documented low-frequency, high-fidelity (LFQ) data towards device/patient-generated high-frequency, low-fidelity data (HFQ) that is easily accessible today. Notably, HFQ data can contain very rich information regarding a community and/or an individual, and thus should be leveraged in healthcare decision making. According to a recent study, the use of PGHD data could potentially reduce the length of hospital stays by 64% and the associated costs by 72% [10]. Furthermore, because of the recognizable impact of PGHD, in 2015 the Office of the National Coordinator for Health Information Technology (ONC) began drafting a policy framework to integrate PGHD as part of the patient-centered outcomes research [11].

1.2 Key Research Problem

Given the emerging data trends in healthcare, a key problem facing heathcare researchers and practitioners today is thus **how to successfully migrate towards the use of the high-frequency, low-fidelity (HFQ) data in the healthcare domain** [12]. In this section, we describe the three dimensions to this key problem in detail. As alluded to previously in Section 1.1, HFQ data is becoming substantial for helping providers make better care decisions for patients. Patient-generated health data in particular (1) discloses important information about the wellbeing of patients in between medical visits, (2) gathers data on an ongoing basis rather than at each episodic provider visit, and (3) provides insightful information relevant to preventive care and long-term care management [13].

1.2.1 Challenge Overview

To better illustrate the scale of HFQ data, we provide the following example scenario. Suppose that a patient uses a smart watch to track her heart rate 10 times a day and her phone to track physical activities and sleep patterns throughout the 24-hour day. If each data entry requires 20 bytes of storage (which includes only a timestamp and some observed value) and fitness and sleep data are generated at every 10 seconds, then in a year 63MB ($= 6 records/min * 20B/record * 525600 min/year + 10 records/day * 20B/record * 365 days/year \approx 63 * 10^6 B = 63MB$) of data will be generated for that single patient. For a primary care provider who on average sees approximately 2500 patients in a year [14], the raw data available to a single provider could potentially be as large as 160GB per year. Understandably, it is impractical for a provider to examine and analyze such enormous amounts of raw HFQ device data in real-time during each patient's in-office visit, but the data may contain important information regarding that patient's continuous health status.

Furthermore, compared to medical and lab tests, many health conditions can be captured much more quickly and cheaply via IoT-based devices. For instance, one basic metabolic lab test for blood glucose at a care setting could cost between $35 to $7300 [15]. While drug store blood glucose meters and test strips that provide at-home self-tests have a one-time cost between $20 to $80 on average [16]. When treating chronic conditions, clinicians may also prescribe mobile apps to patients as a more accessible and cost-effective tool to monitor continuous health behavior [17, 18].

As shown in Figure 1.1, HFQ data (such as diet, sleep/activity, environmental data, and photograph diaries) should be integrated with LFQ EHR data (including but are not limited to provider notes, allergies, lab results, and prescriptions) when available to provide clinicians with more comprehensive knowledge about patient health. Unfortunately, the technical infrastructures in healthcare today lack the support for utilizing HFQ data. An important research question to answer is thus: **what are the architectures, techniques, and tools that can help providers move towards the use and integration of HFQ health data at scale?** In the following subsections, we investigate three dimensions to this key research question.

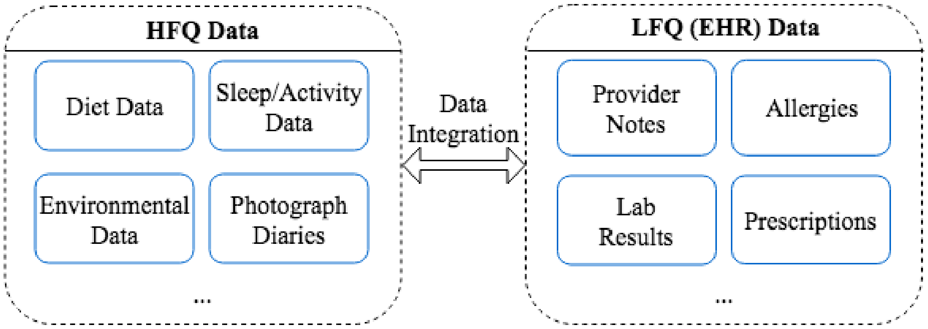


Figure 1.1: Example Types of HFQ and LFQ Data that Should be Integrated to Provide a Comprehensive View of Patient Health.

1.2.2 Challenge Dimension 1: How to scalably extract insights from large volumes of HFQ data

Electronic health record (EHR) systems present patient data to providers in a structured and readable manner that they have been trained to understand. On the contrary, the new generation of HFQ health data originating from IoT devices or mobile apps either exists in raw signal forms or in forms that are vendor-specific. Patients can often access these data through with months or years of history records. However, because each patient has limited face-time with a provider during each visit, it is impractical for providers to examine those heterogeneously formatted high-volume data generated by patients, much less to extract any clinical insights directly from those data in real-time. As such, a technical architecture is needed to convert insights obtained from the HFQ data into similarly structured and readable formats as the EHRs that providers are accustomed to. In particular, the architecture must also meet the following expectations: (1) it should leverage traditional LFQ data as ground truth to help improve the reliability of HFQ data, (2) it needs to allow clinicians or researchers to answer questions or validate inferences about HFQ data, upon which insights may be learned from the data, and (3) it reduces the dimensionality of HFQ data if/when necessary by intelligently filtering out impertinent data to a specific condition under observation.

1.2.3 Challenge Dimension 2: How to Create a Scalable HFQ Data Integration/Sharing System

In cases where HFQ data may be converted to readable records or succinct insights, all newly procured information should be integrated with existing EHR records to maintain a more complete and meaningful picture of patient health history. However, healthcare today faces interoperability challenges even across EHR systems provided by different vendors [19, 20], which makes the secure sharing/integration of HFQ data much harder to achieve. Several major hurdles are barring the systems from interoperating at their full potential. First, EHRs (the most common LFQ data) and filtered HFQ data alike remain siloed and fragmented within the security settings of their data owners. Second, it is hard to identify and authenticate healthcare participants (*e.g.*, clinicians, healthcare app, and patients) to initiate the data sharing process between them. Third, the lack of governance makes it challenging to establish necessary trust relationships particularly between health app providers and certified healthcare providers, which also creates a barrier to enabling direct clinical communications [21]. Fourth, inconsistencies in the adoptions of health data standards prevent shared data from being interpretable by entities that do not employ the same data representation. Finally, there is incompatibility in existing health management software systems across different vendors that impedes the direct exchange of clinical messages [20].

These barriers to healthcare interoperability have spurred a lot of interest in the use of blockchain technology, which is an emerging infrastructure with properties of decentralization, trustless exchange, and replicated storage. These unique properties of blockchain technology make it a potential solution to mitigate many of the aforementioned problems [22]. Although, the most popular (and successful) type of blockchain use cases is the trustless exchange of cryptocurrency tokens [23, 24], which is much less complex than the data sharing problem facing healthcare. Existing blockchain solutions cannot be employed out of the box to support clinical data sharing needs. Important design decisions must be made in the context of technical requirements defined by healthcare specialists. Therefore, a blockchain-based architecture that accounts for healthcare-focused data sharing is therefore needed to integrate health data, including the traditional LFQ EHR data as well as the increasing HFQ data (and insights learned from it).

1.2.4 Challenge Dimension 3: How to Effectively Design Blockchain-Based Healthcare Data Integration/Sharing System in a Secure and Evolvable Manner

Unlike conventionally centralized systems, a blockchain-based data integration and sharing system requires much more precise decisions to be made during the design phase. Because of the decentralized nature of blockchain, data and operations stored and encoded on it (i.e., via smart contracts, which are programmable and executable code) are available to all participants of the blockchain network and are much harder to modify if at all possible [23, 24]. This is by design, as it provides an important property of blockchains–immutability–that ensures integrity and non-repudiation of the data along with transactions of data on-chain.

These properties make blockchains feasible for performing data exchange operations without a centralized intermediary but can also introduce additional domain-specific challenges for healthcare. First, immutability may render a blockchain-based system vulnerable to attacks without an extra careful design. Previous infamous attacks on several blockchain apps revealed that even the most subtle bug or mistake concealed in a smart contract code could lead to the most severe security breach (e.g., health data compromise with serious financial and legal consequences). One example is the bug exploited from a crowdfunding app (*i.e.*, DAO) built in the Ethereum blockchain that caused then worth $50 million dollars of the Ethereum cryptocurrency to be siphoned in 2016 [25].

Second, due to healthcare user or regulation requirements, system updates may be mandated to meet user expectation or reach compliance. In consequence, historic versions of any system components implemented or legacy data stored on-chain may be completely unusable, if logic for referencing updated components or methods for retrieving previous data were not carefully crafted in the first place. This results in a strong tension between components implemented on-chain (via smart contracts) being permanent and the fact that the system design has to accommodate changes.

Just like good design practice helps maximize modularity and separation of concerns in a centralized system design, it is especially critical in designing a blockchain-based system to protect sensitive health information by minimizing security vulnerabilities while providing capabilities to support future upgrades to the system. Repeatable design patterns and anti-patterns must both be identified and well-documented to serve as guiding principles for designing blockchain-based health data sharing systems.

1.3 Overview of the Research Goals

In this section, we present an overview of our research goals to facilitate the use and integration of high-frequency, low-fidelity data in healthcare by solving the challenges discussed in Section 1.2. Specifically, we describe the construction of key architectures related to HFQ data and data sharing and then provide blockchain-based design recommendations based on our research and design experience. A summary of identified research goals and corresponding research approaches appears in Table 1.1, with references to their respective sections.

Table 1.1: Summary of research goals and respective approaches with references to the respective sections

Research Goal	Research Approach	Section
Extracting scalable insights from HFQ data	Learned filtering architecture in the self-management behavior of adolescents with type 1 diabetes case study	Section 2.3
	Learned filtering architecture in the hand hygiene compliance monitoring case study	Section 3.8
Secure and scalable data sharing and integration	FHIRChain using a remote tumor board case study	Section 4.5
Evolving the decentralized data sharing system	A pattern language for implementing blockchain-based healthcare systems	Section 5.5

Figure 1.2 presents an integrated view of this research that provides three main contributions: (1) a learned filtering architecture that uses data science methods to scalably extract clinical insights from HFQ data, (2) a blockchain-based decentralized architecture called ”FHIRChain” that enables the secure sharing and integration of healthcare data

(HFQ and/or LFQ) using FHIR, which is an existing clinical data exchange standard, and (3) a pattern language describing well-documented design practice for software engineers to implement blockchain-based health systems that meet technical requirements defined by healthcare experts while avoiding fatal mistakes in the design.

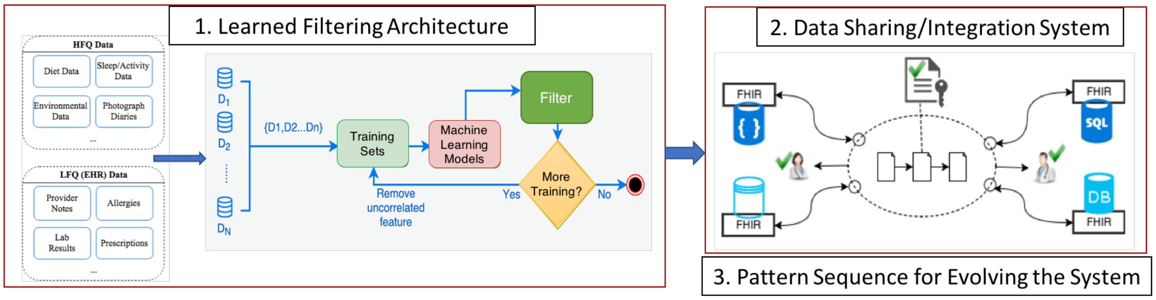


Figure 1.2: The integrated view of the research approach

Contribution 1: The Design of a Learned Filtering Architecture for Drawing Scalable Insights from HFQ Data

To facilitate the process of drawing scalable insights from HFQ data, we designed a learned filtering architecture (LFA) using data science approaches. The goals for LFA are: (1) to leverage existing LFQ data as ground truth (*i.e.*, labeled target variable to learn) to help gauge the overall accuracy of collected HFQ data via one or more machine learning models, (2) to help understand the connection between LFQ and HFQ data by testing and validating relevant hypotheses on various HFQ data subsets, and (3) to reduce the dimensionality of HFQ data for future data collection if/when necessary.

LFA eases the transition from using LFQ to HFQ data with iterative training methods from machine learning to obtain insights from HFQ data as closely resembling LFQ data as possible. LFA requires an initial observation period to obtain both the LFQ data (e.g. human-observed outcomes or provider-documented clinical outcomes) and HFQ data (e.g. device- or patient-generated data that could be used to produce the same observed outcomes). Using LFQ data as ground truth labels and preprocessed HFQ data as training features, the combined dataset is then fed into several machine learning models to learn to predict ground truth-like outcomes using any subset of HFQ-extracted features. The preliminary results from the learning models serve two purposes: (1) to demonstrate the overall accuracy of HFQ data against the ground truth and (2) to filter out one or more best-fit learning models using which HFQ data can learn to produce similar insights as ground truth LFQ data.

This architecture also supports posing and verifying hypotheses regarding correlations between the training features and ground truth. Features that can be classified into the same category (such as time of day and day of week, both of which are a class of ”time variables”) are, by default, segmented as a subset to train various machine learning models. Similarly, automatically or manually selected feature subsets are also fed into the learning model component. The aggregated results from all machine learning models involved are then combined to create a learned filter with a set of commonly used performance metrics in evaluating predictive models, such as accuracy, recall, and precision [26]. The filter component then temporarily stores learned results during the iterative training process of evaluating different hypotheses. When the evaluation process completes, the filter's threshold tolerance values can be configured to produce feature subsets meeting the thresholds. This is a key feature used to reduce the dimensionality of the HFQ data input by eliminating portions of the original data that did not satisfy the thresholds.

We will demonstrate the use of the learned filtering architecture with two specific case studies: self-management behavior in adolescents with type 1 diabetes and hand hygiene compliance monitoring. The details of these two case studies are presented in Chapter 2 and 3.

Contribution 2: The Design of FHIRChain–a Blockchain-Based Decentralized Architecture for the Secure and Scalable Health Data Integration/Sharing

As discussed previously in Section 1.2.3, secure and scalable integration/sharing of HFQ data is essential for presenting patient health condition in a comprehensive manner and also for collaborative clinical decision making. Conventional clinical data efforts are often siloed, creating barriers to efficient information exchange and thus impeding effective care. To ensure that health-related data (both LFQ data and learned insights or the entirety of HFQ data) can appropriately flow across various healthcare entities, we created a blockchain-based decentralized architecture, called FHIRChain [21], to enable the secure and scalable data integration and sharing. In particular, we studied the feasibility of applying blockchain technology to clinical data sharing in the context of technical requirements defined in the ”Shared Nationwide Interoperability Roadmap” by the *Office of the National Coordinator for Health Information Technology* (ONC) [27].

To deliver a reusable conceptual framework, we employed a multi-phase approach. First, we provided an in-depth analyses of five key ONC requirements and their implications for blockchain-based systems (Section 4.4), including: (1) verifying identity and authenticating all participants, (2) storing and exchanging data securely, (3) ensuring permissioned access to data sources, (4) applying consistent data formats, and (5) maintaining modularity. Second, we created FHIRChain as a blockchain-based architecture designed to meet each of the five ONC requirements via: (1) the use of public key infrastructure (PKI) based digital health identities for authenticating healthcare participants (*e.g.*, certified providers and mobile health app providers), (2) hybrid on-chain/off-chain data store and exchange via reference pointers, (3) a token-based permission model, (4) enforcing HL7’s *Fast Healthcare Interoperability Resources* (FHIR) [28] standard for exchanging clinical data, and (5) applying the model-view-controller (MVC) pattern in the design. Third, we demonstrated a FHIRChain-based decentralized app (DApp) by walking through a remote tumor board case study. Fourth, we applied the principles learned in designing FHIRChain to another case study in the context of opioid prescription tracking [29].

Contribution 3: A Key Pattern Sequence for Effectively Designing Blockchain-Based Health Data Sharing Systems in a Secure and Evolvable Manner

To address the healthcare-specific requirements for a data sharing system that is secure and upgradeable, we propose and provide detailed documentations of a key pattern sequence for effectively designing a blockchain-based system for healthcare. Widely known in the software engineering discipline, a design pattern is a general, repeatable solution to a problem that commonly occurs in software designs [30]. Unlike certain software design techniques that developers may understand how to apply to specific problems, design patterns can apply to a broader range of problems. They are typically documented in a format that allows software engineers to communicate design choices using well-known, well-understood names without requiring specifics tied to a particular problem [31].

For designing an end-to-end system targeting the specific data sharing problem in healthcare, a pattern sequence that details the order and combination of essential patterns can provide more guidance to developers than standalone design patterns. The application of the pattern sequence would help create a basic template or reference architecture that take into account domain-specific requirements, *i.e.*, a blockchain-based data sharing system that is secure and upgradeable. The patterns discussed in our sequence were (1) implemented based on our understanding of the healthcare technique requirements and our experience in designing various blockchain-based healthcare systems, (2) mined using commonality and variability analysis [32] by examining code from $\tilde{7}$,000 Ethereum smart contracts written in Solidity, available on *Etherscan.io* [33], and analyzing the applicability of repeatable patterns in the healthcare space, or (3) selected from traditional software patterns which we deemed relevant to the healthcare domain. Overall, we identified and integrated a sequence of eight patterns for creating a system design that would meet the ONC requirements. Despite the pattern sequence being presented using the Ethereum [24] blockchain (the most popular blockchain environment for writing smart contract today) as an example infrastructure for a data sharing system, the core concepts are generally applicable to other blockchains that support smart contract development.

Dissertation Outline. The remainder of this dissertation is organized as follows: Chapter 2 presents the learned filtering architecture and its application in a case study of self-management behavior in adolescents with type 1 diabetes; Chapter 3 describes another case study of the learned filtering architecture applied in the context of hand hygiene compliance monitoring; Chapter 4 describes the FHIRChain architecture that is a general solution to data sharing and a FHIRChain-based decentralized app applied to a case study of a remote tumor board in telemedicine; Chapter 5 proposes a pattern sequence that summarizes recommended design practice for software engineers to develop evolvable and scalable health IT systems based on blockchain technology; Chapter 6 presents concluding remarks and future directions of this research.

Chapter 2

LEARNED FILTERING ARCHITECTURE AND ITS APPLICATION IN UNDERSTANDING THE SELF-MANAGEMENT BEHAVIOR OF ADOLESCENTS WITH TYPES 1 DIABETES

Despite the availability of high-frequency, low-fidelity (HFQ) data capturing health related activities, it has been underutilized in clinical settings. In this chapter, we describe a learned filtering architecture (LFA) using a case study focused on understanding the self-management behavior of adolescents with type 1 diabetes (T1D) to demonstrate how to scalably leverage the use of HFQ data for improving quality of care.

T1D a prevalent pediatric chronic disorder with worldwide economic and social impact. It requires patients to perform many daily self-management tasks for survival, which is particularly challenging for young people with T1D due to developmental, psychosocial, and contextual barriers. Technologically-assisted ecological momentary assessment methods can be used to assist patients with identifying potential barriers that interfere with appropriate T1D self-management where and when they occur. Key contribution of this chapter is a generalized learned filtering architecture based on advanced machine learning methods to better utilize HFQ data. We also provide the following two contributions specifically to the research on self-management of T1D in adolescents: (1) we demonstrate how EMA data are used in the LFA to construct machine learning classifiers that predict two T1D self-management behaviors: insulin administration and self-monitoring of blood glucose (SMBG) and (2) we investigate and discover significant impact of novel data capturing contextual, psychosocial and time-varying factors on patient self-management behavior.

2.1 Problem Overview

Type 1 diabetes (T1D) is a prevalent chronic illness with increasing incidence rates reported worldwide [34, 35]. It is an autoimmune disorder where the body produces little or no insulin and requires patients to perform critical self-management tasks multiple times per day [36]. Self-management in T1D involves frequent monitoring of blood glucose, estimating carbohydrate intake, and administering insulin amongst other regular tasks related to maintenance of devices, supplies, and attention to factors that influence blood glucose variability and patterns.

Inadequate self-management and poor glycemic control is related to serious short- and long-term consequences, including retinopathy, neuropathy, and mortality [37, 38, 39]. Adolescents and young adults have the worst glycemic control of any age groups [37]. For young people with diabetes, living successfully with T1D is particularly hard due to developmental, psychosocial, and contextual barriers to self-management [40, 41, 42].

A common approach used to improve self-management of diabetes involves promoting and supporting problem solving skills [43]. To identify problems related to self-management, patients, caregivers, and clinicians must rely on the review of blood glucose and insulin data from devices along with a patient-generated recall of potentially relevant behavioral, emotional, and/or situational events. This method of utilizing retrospective memory or recall, however, has been identified as generally unreliable and potentially biased in nature [44].

To address the limitations of recall in health behavior research, ecological momentary assessment (EMA) methods have been developed and successfully utilized in a range of health conditions. EMA methods provide a more proximal (and often more accurate) technology-mediated method to monitor and assess the contexts, subjective experiences, and processes that surround health decisions in daily life [45, 46]. Furthermore, with more relevant, proximal, and frequent observations per patient, EMA methods generate rich data from which to more accurately relate previously identified correlates of health behavior and identify novel correlates for interventional targets [47].

The data generated from EMA systems is particularly suited to analytic techniques that identify patterns. In particular, machine learning methods have been employed to detect type 2 diabetes and identify targets for improvement in diabetes management and outcomes [48, 49, 50]. These advanced methods have been used less frequently, however, to examine patient-generated data, behavioral patterns, and self-management in diabetes. We believe that machine learning methods will ultimately become more effective at identifying meaningful sub-groups of self-management styles and self-management phenotypes upon which to base personalized behavioral treatments [51].

In this chapter, we present research on leveraging machine learning methods to help investigate how novel data focused on contextual, psychosocial, and time-varying factors relate to patient self-management. In particular, we provide the following contributions to this study:

- We describe a generalized learned filtering architecture (LFA) that is used to extract clinical insights from data generated by patients
- We demonstrate the application of LFA with a Random Forest [52] classifier in this case study to extract groups of similar features that are predictive of two self-management behaviors in adolescents with T1D: insulin administration (IA) and self-monitoring of blood glucose (SMBG)
- Using results from the predictive models, we investigate whether novel EMA data focused on contextual, psychosocial and time-varying factors relate to patient self-management behavior

2.2 Background and Related Work

This section summarizes the background of our work and related research, focusing on the barriers to maintaining treatment adherence, the importance of problem solving skills for adolescents with T1D, and use of ecological momentary assessment (EMA) methods and machine learning applications in other diabetes studies. We then present gaps in existing research that we target in this chapter.

2.2.1 Overview of Type 1 Diabetes

T1D is an autoimmune disease where the body produces little or no insulin, necessitating multiple daily injections of insulin or insulin pump therapy for survival. A key issue for individuals with T1D is glycemic control, where T1D patients monitor their blood glucose (BG) levels multiple times per day using BG meters and (less frequently) with the addition of continuous glucose monitoring devices. A 2-3-month average of glycosylated hemoglobin is assessed in clinics via the HbA1c test, which is indicative of overall BG control. In-target glycemic control is critical in delaying or avoiding complications, both short-term (*e.g.*, hypo- or hyperglycemia, diabetic ketoacidosis) and long-term (*e.g.*, retinopathy, kidney disease, neuropathy, cardiovascular disease) [53].

In addition to monitoring BG, other related tasks performed daily by individuals with T1D include counting carbohydrates and insulin self-dosing and administration. Support of self-management behaviors that increase in-target BG values is especially important in adolescents with T1D. These behaviors are important not only because of the long-term health impacts of inadequate glycemic control, but also because this population is at high risk of struggling with adherence to their diabetes treatment regimen [54].

Barriers to adherence. Diabetes adherence is hard due to the frequency and complexity of self-management, *e.g.*, tasks must be performed around meals, snacks, and exercise. Psychosocial and environmental factors, such as location, emotional state, social context, and other activities, can thwart diabetes treatment adherence. Moreover, disrupted self-management may be associated with daily living patterns, such as time pressures during certain times of day, social context, or specific activities like sports practice [55]. Adolescents with T1D are also susceptible to negative emotions and difficulties in dealing with society and interacting with others, which could also result in poorly controlled symptoms [56].

Importance of problem solving skills. Problem solving interventions have shown success in helping adolescents with T1D improve their self-management practices and health outcomes through reducing barriers to adherence [55, 57]. Successful problem solving is predicated upon accurately identifying those barriers and patterns of behavior. Based on previous research [57, 58, 59, 60], improved recognition of how self-management is related to situational, contextual, and psychosocial factors should provide a data-based means to address the first step in problem solving, known as problem orientation, problem identification, and/or problem awareness.

By guiding pattern recognition and problem awareness, MyDay was designed to improve diabetes self-management skills. In particular, it provides IoT-enabled personalized real-time feedback and behavioral problem solving support. Behavioral pattern recognition and problem awareness are cognitively hard for adolescents due to their normative developmental stage of higher-order executive functions, the multifactorial nature of causation, and the repetitiveness of self-management.

Ecological Momentary Assessment (EMA). EMA is a method for providing more accurate problem solving data by systematically studying an individual in (or near) real-time to assess and relate the individual's experiences and environment to health behaviors and outcomes [61]. EMA helps identify novel behavior patterns through data collection at either random or specified critical points over time [61, 62, 63, 64]. By collecting assessments close in time or at the time of events of interest, EMA helps minimize response bias that may otherwise occur using retrospective methods [61].

Given the pervasiveness of smartphone adoption in adolescents and emerging adults, momentary assessment can be feasibly implemented via mobile and wireless technologies and then streamed to researchers. Adolescents with T1D perform virtually all their self-management practices outside of a medical setting (*e.g.*, they are expected to check their

BG, count carbohydrates, and dose insulin while at home, school, or out with friends). To discern and address factors interfering with appropriate diabetes self-management, potential barriers must be identified where and when they occur. EMA is an ideal tool for studying the interaction between person variables and the natural environment of health behaviors [65] and has been successfully used to study diseases like asthma, cancer, eating disorders, and diabetes [66, 67, 68, 69].

2.2.2 Related Work

Prior research [42] has focused on identifying psychosocial correlates and predictors of self-management in chronic illness in general and specifically in diabetes. Our study focuses on factors that were previously associated with self-management, but were also amenable to EMA methods. Factors most appropriately assessed through these methods are those that are

- thought to vary more frequently and/or occur relatively more frequently and
- hard to identify in daily experience to associate them to medical events, health decision-making, and/or symptoms.

Our EMA pilot study [70] assessed a broad sampling of factors that influence diabetes self-management. These factors included stress [71], fatigue [72], mood [73, 74], location [75], and social context [41]. We also collected other factors, including contextual barriers, such as rushing, lack of diabetes supplies (such as blood glucose test strips), and stigma [42, 76]. As a continuation of the pilot study, our research described in this chapter leverages the EMA data to determine if psychosocial factors impact self-management behavior. If so, we aim to identify the type(s) of features which have relatively greater impact. Understanding the potential connections between psychosocial phenotypes and self-management behavior can help focus behavioral interventions tailored to individual patients.

Machine learning (ML) methods have been applied in various studies focusing on the improvement of diabetes management and control. The following subsections present recent research using ML in clinical intervention of diabetes and also self-management of diabetes.

2.2.2.1 Machine Learning Approaches in Clinical Interventions.

Many efforts have been associated with clinical interventions that study the effect of therapy and overall patient's lifestyle on glucose metabolism. Philip et al. [77] surveyed various types of sensors used in real-time continuous glucose monitoring (RT-CGM) in youth with T1D across different clinical studies. They observed that RT-CGM can potentially help patients improve in metabolic control of T1D, provided that there is adequate education and support on sensor therapy and the devices used. Studies in [78, 79, 80] constructed and fine-tuned different ML models to predict future blood glucose levels based on historical physiological data, such as readings from continuous glucose monitoring (CGM) systems. Bondia et al. [81] used Support Vector Machines to detect incorrect blood glucose measurements in CGM systems.

Artificial neural networks were applied in [82] to create a controller for potentially managing insulin dosage. Biester et al. [83] applied ML methods to predict low blood glucose levels for triggering an automatic stop of insulin delivery in a sensor-augmented insulin pump. Their study documented reduced risk for hypoglycemia in pediatric T1D patients without increasing HbA1c. Prototype portable artificial pancreas (AP) [84, 85] have been developed using glucose sensors, insulin pumps, and radio-bluetooth connections. More advanced AP systems, such as presented by Kovatchev et al. [86], also integrated smart phones with a wireless network for data transmission and remote monitoring. Short-term clinical studies of these new systems conducted safety of use in young people with T1D, but longer-term studies are needed to monitor their full functionality.

2.2.2.2 Machine Learning Approaches to Improve Self-Management.

Another category of related T1D research focuses on health monitoring systems to provide patients with effective means for tracking, displaying, or predicting important T1D self-management variables, such as BG, food intake, and physical activity, as seen in [87, 88, 89]. Sudharsan et al. [90] trained and compared various prediction models to identify hypoglycemia for patients with type 2 diabetes using self-monitored blood glucose (SMBG) readings. More recent work has involved more personalized approaches, such as individually-tailored notifications and educational support. Li et. al. [91] proposed a predictive model by capturing patient similarities of pooled population data to personalize blood glucose prediction for an individual. Using a mobile-based approach, they collected pertinent daily events including insulin, meals, exercise, and sleep, and implemented the proposed prediction model as a prototype mobile application to create personalized notifications.

Machine learning has also been applied to provide lifestyle support, such as the smartphone-based food recognition system described in [92] and the prediction of energy expenditure and type of physical activity using accelerometers [93]. Boulos et al. [94] presented a class of digital intervention in diabetes that gamifies disease management using the Internet and affordable mobile and tablet devices. Digital games utilize social cognitive theory to increase healthy behaviors and psychological outcomes, promoting better self-care.

2.2.3 Gaps in Existing Research

Existing machine learning approaches in clinical interventions reply heavily on sensor therapy, which may not be easily accessible to many adolescents with T1D. Furthermore, patients must be educated against extra carbohydrate intake in response to an alarm associated with low BG prediction to avoid rebound hyperglycemia. Existing ML approaches targeting the improvement of self-management have not seen the integration of important aspects of behavior data into the study. One common thread of existing research is the primary focus on predictability, *i.e.*, how accurately a model can predict a specific outcome such as glucose values and hypothermia. What is missing in research is the understanding of what phenotypes, conditions, or group(s) of those factors truly influence the outcome vectors of interest–what type(s) of data that a clinician or patient should pay special attention to in order to make a better care decision or perform more effective self-care.

2.3 Our Research Design and Methods

This section describes the design of LFA and methods we employed to apply the LFA in this case study. We analyzed data from subjects enrolled in a feasibility trial of the mobile EMA and feedback MyDay app using a 30-day assessment period [95]. Subjects were randomized on a 2:1 ratio to the Myday app group + Bluetooth meter (n=31) and a control group (n=15) who provided blood glucose (BG) data only using Bluetooth BG meters.

Figure 2.1 presents the workflow of our learned filtering architecture (LFA) for processing, analyzing, and extracting insights from the data collection.

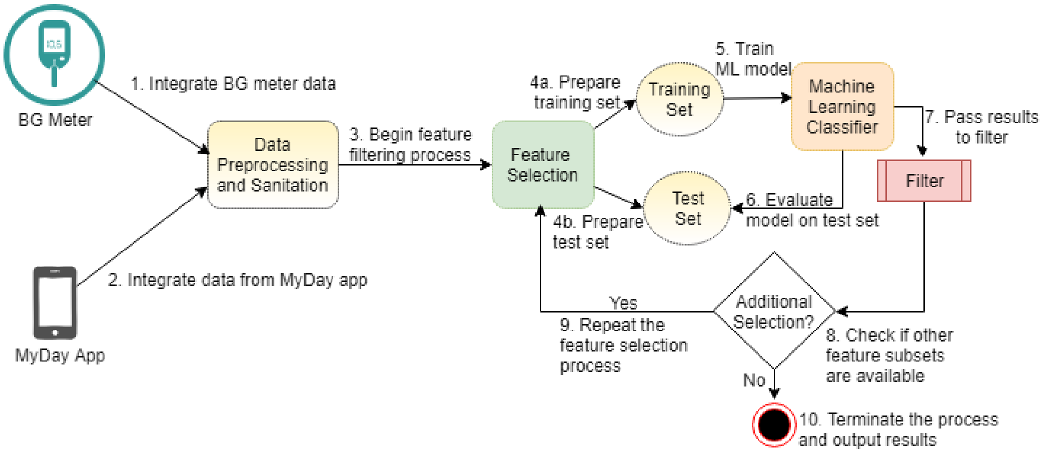

BG Meter
1. Integrate BG meter data
MyDay App
2. Integrate data from MyDay app
Data Preprocessing and Sanitation
3. Begin feature filtering process
Feature Selection
4a. Prepare training set
Training Set
5. Train ML model
Machine Learning Classifier
4b. Prepare test set
Test Set
6. Evaluate model on test set
7. Pass results to filter
Filter
Additional Selection?
8. Check if other feature subsets are available
Yes
9. Repeat the feature selection process
No
10. Terminate the process and output results

Figure 2.1: Iterative Process of Our Learned Filtering Architecture (LFA).

As shown in the workflow diagram, we first integrate BG meter data and the EMA data collected from the MyDay app as a complete dataset fed into the LFA (steps 1 and 2).

Next, the LFA performs necessary pre-processing and data sanitation, such as normalizing numeric values and removing empty entries (step 3). After this step, we begin the data filtering process where subsets of features are extracted from the cleaned data (either based on configurable human input or automatic selection, and in this particular study, the features are grouped together based on similar types) to create multiple data subsets that are then split for training and testing (steps 4a and 4b).

The training set is used to train a machine learning classifier *i.e.*, Random Forest in our study (step 5), and the test set is used to evaluate the trained model (step 6). The classification results obtained from the current feature subset are then sent to the *Filter* component to be later compared with other feature subsets (step 7). The filter component has a configurable tolerance value, which is used to select feature subset(s) that have relatively good classification results compared to the most performant model(s) or other benchmark(s).

Next, the LFA checks whether other feature subsets are available for processing (step 8). If so, the *Feature Selection* process is repeated to create the next subset (step 9). Otherwise, the filtering process terminates and ouputs the filtered results, *i.e.*, feature subsets with relatively strong predictive power of the target outcomes (step 10).

After feature selection, a large portion (*e.g.*, 75%) of the input data forms a structured training set. This training set is used to construct a machine learning classifier. The remaining data becomes a hold-out test set, which is used to evaluate and enhance the classifier.

The classification results then go through a filter component that extracts the most impactful predictor group(s) of the target class variable. For example, if the performance metrics exceed their threshold values, the predictor group is added to the final output queue. When all feature subsets have been evaluated, LFA returns the final insights learned from the input data.

2.3.1 Subjects

A total of 49 patients were recruited from an academic pediatric diabetes center. Youth who were patients in the clinic were invited to participate if they were between the age of 13 and 19, had been diagnosed of type 1 diabetes for at least 6 months, owned a smart phone, understood and spoke English, and were willing to use a Bluetooth meter during the study. Three subjects dropped out of the study noting competing demands, leaving 46 for our analyses (n=31 in the app + meter group; n=15 in the meter-only group).

2.3.2 Momentary Assessments and Glucose Meter Data

The goal of our study was to examine associations between self-management (SMBG, self-monitored blood glucose and IA, insulin administration) and other relevant collected data, including participant demographics and momentary assessment variables. All blood glucose data for both groups was unobtrusively obtained using iHealth [96] Bluetooth meters. The app group was instructed to use the MyDay mobile app at each mealtime and bedtime to answer questions focused on factors likely to impact self-management of diabetes, including stress, fatigue, mood, social context, location, and contextual barriers to self-care [95]. Mealtime insulin administration was also self-reported into the app.

Blood glucose monitoring was objectively assessed via data transfer from the Bluetooth meters. The MyDay app provided notifications personalized to meal-times identified by participants each day as a reminder to complete EMA. Timestamps were associated with all data entries. Bedtime EMA was not included in analyses since self-management tasks could not be expected at that specific time point as they are with mealtimes. A subset of only mealtime EMA were used in analyses for the app group. At the initial recruitment session, parents of minors and adult participants provided consent, assent, demographic information.

2.3.3 Statistical Analyses

We were interested in studying the factors associated with the following

- All daily SMBG frequency in terms of the following two observations for all study participants: (1) if a subject monitored more than 4 times a day (4 being the clinically recommended minimum number of daily BG measurements [97]) and (2) if a subject monitored fewer than 4 times a day,
- Whether SMBG was missed or not at mealtimes, and
- Whether insulin was administered or not at mealtimes.

2.3.4 Feature Categories

Based on our hypothesis that different feature types may have varying impact on the self-management outcomes, we configured the LFA to produce data subsets of the following categories (wherever there were data present): demographics, time variables (time of day, weekday/weekend), context (social context and location associated with each mealtime app entry), stress/fatigue/mood values, and situational barriers such as without supplies (dichotomous behavioral questions). By grouping features into categories, we potentially eliminate variables that are less relevant to the outcomes. In turn, we significantly reduce the amount of information requested from MyDay app users in future studies.

Although the number of observations per participant was substantial, the overall number of participants was relatively small. Naturally, because performing self- management tasks is critical for patients with T1D, adolescents are expected to adhere to the daily regimen. As a result, the collected data encountered some imbalance in the distribution of the outcomes, with failure to perform these tasks (particularly missed mealtime insulin) being the minority instances.

It is well-known in the machine learning community that classification models constructed using imbalanced datasets may result in the minority class being neglected [98]. To avoid this problem, we applied an imbalanced learning algorithm that combined the Synthetic Minority Oversampling Technique (SMOTE) [99] and Tomek link [100]. Both SMOTE and Tomek link have been used effectively for training imbalanced data, especially for small datasets [101, 102, 103]. Our combined algorithm oversampled the minority class and cleaned noisy data, but only in the training set.

We employed SMOTE to enrich the minority class by creating artificial examples in the minority class, rather than replicating the existing examples to avoid the problem of overfitting. Specifically, SMOTE creates new samples from linear combinations of two or more similar samples selected from the minority class using a distance measure. Each instance is created by perturbing the original sample's attributes one at a time by a random amount within the difference to the neighboring instances.

We employed Tomek link to remove noisy data from the majority class that may have been introduced from oversampling. Noisy data is detected by comparing the distances between any two samples from different classes and the distances between an arbitrary sample and one of the two samples [100]. If the distance between the former pair is smaller, then either one of the samples in that pair is a noise or both are border-line instances [104].

2.4 Results

This section analyzes the results obtained from the LFA we constructed using the method described in Section 2.3.

2.4.1 Descriptive Statistics of the Sample

Table 2.1 shows the demographic and clinical characteristics of the study sample.

Table 2.1: Characteristics of the Sample (n=46)

Variable	Mean (SD) or %
Age	13.33 (1.67)
Female	53.33%
Race/ethnicity	
White	84.44%
African American	10.20%
Asian	2.22%
Hispanic	2.22%
Other	0.00%
Father education	
Less than high school	2.22%
High school/GED	28.89%
2-year college	15.56%
4-year college	33.33%
Master's degree	11.11%
Doctoral degree	0.00%
N/A	8.89%
Mother education	
Less than high school	0.00%
High school/GED	22.22%
2-year college	26.67%
4-year college	37.78%
Master's degree	4.44%
Doctoral degree	0.00%
N/A	26.67%
Income	
Less than $25,000	4.44%
25,001−35,000	6.67%
35,001−75,000	15.56%
75,001−100,000	31.11%
100,001−100,000	26.67%
More than $70,000	6.67%
N/A	8.89%
Duration of diabetes (years)	5.47 (3.59)
HbA1c	9.03 (1.91)
Use insulin pump (yes)	57.46%

2.4.1.1 Descriptive Statistics

From all 46 participants, we obtained a total of 6,524 blood glucose measurements from their Bluetooth glucose meters. After aggregating each individual's SMBG counts by day and combining their demographic data, we produced a new dataset with 1,779 daily SMBG entries with the following schema:

1. Feature Category: *Demographics*, including gender, age, father's education, mother's education, family income, and race

2. Feature Category: *Time Variables*, including weekday, weekend, and time of day.

After analyzing the target outcome variables, we observed the distribution as follows: *Below 4* class contains 794 True (count < 4) outcomes and 839 False (count ≥ 4) outcomes, which is a fairly evenly distributed set. For *Above 4*, however, the True (count > 4) outcome had 475 entries, while False (count ≤ 4) had 1158 entries, a fairly imbalanced class.

To minimize the potential imbalance in the training set and maximize learning performance, we first split the dataset into 75% for training and 25% for testing and then applied an automatic imbalanced learning algorithm to the *training set* for a more even distribution for *Above 4*. As discussed in Section 2.3, our imbalanced learning algorithm combines the SMOTE and Tomek Link methods.

2.4.1.2 Classification of Daily SMBG Occurrences

We trained the dataset using a Random Forest classifier with a 10-fold cross validation and obtained the classification results against the test data. The results are shown in Table 2.2 for SMBG below 4 and Table 2.3 for SMBG above 4.

As a benchmark for the learned filter component, we used all features for predicting the target variables and recorded the results. The filter then compared the benchmark value with the classification results obtained from each data subset. We configured a tolerance value of 15% for the filter to select subsets of significant predictive power.

Table 2.2: SMBG Below 4 Classification Performance Metrics

Feature Group	Accuracy	Precision	Recall	F1 Score
Demographics	72.6%	0.73	0.73	0.73
Time variables	49.1%	0.51	0.49	0.47
All	71.2%	0.71	0.71	0.71

Table 2.3: SMBG Above 4 Classification Results

Feature Group	Accuracy	Precision	Recall	F1 Score
Demographics	76.5%	0.78	0.77	0.77
Time Variables	55.6%	0.54	0.56	0.55
All	76.5%	0.77	0.77	0.77

2.4.2 Missing Mealtime SMBG and Insulin Administration

2.4.2.1 Descriptive Statistics of the Sample

From the app group with 31 subjects (n=31), we collected a total of 2,535 entries. From this data we extracted 1,855 valid entries that are associated with breakfast, lunch, and dinner records to analyze factor(s) that could impact SMBG and IA at mealtimes.

The target class *Insulin Administration* had a distribution of 1:6 for *True* (insulin missed) vs *False* (insulin administered) outcomes; whereas target class *Missing SMBG* had a class distribution of 1:5 for *True* (SMBG missed) vs *False* (SMBG taken). The dataset used to analyze both target classes was divided into the following subsets of features based on our hypothesis regarding features' relativeness:

1. Feature Category: *Demographics*, including gender, age, father's education, mother's education, family income, and race

2. Feature Category: *Time Variables*, including weekday, weekend, and time point (breakfast, lunch, dinner)

3. Feature Category: *Social Context*, who was the teen with at time of self-management as indicated through EMA (including parent, sibling, alone, casual friend, close

friend, other family, other person, strangers, and boyfriend/girlfriend), and location, including home, school, work, restaurant, friends house, or on the road

4. Feature Category: *Stress, Energy, Mood*, continuous values within range 0-100

5. Feature Category: *Barriers*, psychosocial indicators (including rushing, tired of diabetes, sick, on the road, hungry, wanting privacy, busy, without supplies, low, high, having fun)

After transforming the input data into various smaller subsets, the LFA created classification models for each predictor group using the same 75%/25% split for creating the training and test sets. Due to the imbalance of the dataset in this experiment, we employed the SMOTE and Tomek Link techniques to create artificial samples for the minority class and perform undersampling to remove noise that may have been introduced, both in the training data to ensure the integrity of the actual test data.

The final class distribution of all datasets had a majority-minority ratio between 1:1 and 1.2:1. After comparing the initial results of three classifiers (random forest, logistic regression, and support vector machine) on the training data, we chose the random forest classifier with a 10-fold cross validation that outperformed other models.

2.4.2.2 Classification Results

Tables 2.4 and 2.5 present the classification performance metrics of missing SMBG and missing mealtime IA against their respective tests, using our trained Random Forest classifier.

We configured the filter using the same approach to obtain the benchmark values and tolerance. As a result, the filter selected demographic data as the most predictive group of missing SMBG, while psychosocial barriers and the combination of stress, fatigue, mood values are stronger predictors in the missing IA analyses. We also identified stress, fatigue, mood group and social contexts as the next best predictor subsets for missing SMBG

Table 2.4: Missing Mealtime Blood Glucose Measurement Classification Performance Metrics

Feature Group	Accuracy	Precision	Recall	F1 Score
Demographics	85.5%	0.85	0.85	0.85
Time Variables	71.8%	0.61	0.72	0.64
Social Context	71.3%	0.73	0.71	0.72
Stress, Fatigue, Mood	73.1%	0.71	0.73	0.71
Barriers	75.4%	0.70	0.75	0.68
All	86.7%	0.87	0.87	0.87

Table 2.5: Missing Mealtime Insulin Administration Classification Performance Metrics

Feature Group	Accuracy	Precision	Recall	F1 Score
Demographics	65.9%	0.84	0.66	0.71
Time Variables	56.7%	0.79	0.57	0.63
Social Context	62.1%	0.78	0.62	0.67
Stress, Fatigue, Mood	72.5%	0.78	0.73	0.75
Barriers	75.6%	0.77	0.76	0.76
All	80.1%	0.84	0.80	0.82

because those values only marginally fell below the tolerance values for the performance metrics that we have configured for the filter.

2.5 Discussion

This section discusses our main findings and analyzes limitations regarding our work.

2.5.1 Main Findings

To gain a better understanding of the factors impacting self-management behavior of adolescents with T1DM, our study applied machine learning methods to construct a learning filter architecture (LFA) for novel momentary psychosocial data and other relevant demographic and physiological data. Based on feature similarities, we configured the LFA into meaningful subsets of variables: demographics; social context; stress, mood, fatigue levels; time variables; and psychosocial barriers.

As a benchmark, we compared the predictability of different subsets of data against the general predictability of the behavior using all the features combined. The LFA applied a 15% threshold to evaluate the performance metrics of all subsets, and the preliminary results indicated that (1) stress, fatigue, and mood levels were stronger predictors of both missed SMBG and IA and (2) demographics factors (such as parents education, family income, and race) was best at predicting average daily SMBG outcomes.

Our methods show promise to quantify the impact of psychosocial factors on self-management on a population level. We also employed a similar research approach in previous case studies [105, 106] in the context of identifying patterns of hand hygiene compliance monitoring, from which we obtained very useful initial insights into which type of features had the most impact on compliance behavior. Based on these promising findings, similar experiments are needed with larger samples to advance the assessment and analytic approaches utilized here.

2.5.2 Limitations

For small datasets that have disparities in the frequencies of observed classes or outcomes, applying an oversampling technique is a strategy to mitigate the negative impact this imbalance has on model fitting. Nevertheless, synthetic sampling (undersampling or oversampling) methods have the following drawbacks:

- **Overestimation of performance.** The trained model with synthetic samples may not reflect the class imbalance future studies may encounter, potentially leading to overly optimistic estimates performance.
- **Model uncertainty.** Synthetic samples could induce model uncertainty. Depending on how accurately the synthesized samples represent the actual samples, the prediction outcomes may be better or worse, so the model could appear more or less effective than it actually is.

With the above drawbacks notwithstanding, we did not consider these threats to validity crucial to our goals since we were relatively less focused on absolute levels of predictability for this pilot study compared to relative value of predictor groups. General patterns are harder to obscure by adding artificial samples using the algorithms we had chosen.

2.6 Conclusion

This chapter reports the results of a study that applied machine learning methods to better understand what triggers poor self-management behavior in adolescents with T1D through. We learned the following lessons from our study:

- **LFA can reduce the scale of EMA data collection**. By employing the learned filtering architecture (LFA), we systematically selected the more relevant information by filtering out data fields that had relatively less impact on the outcomes. As we collect larger-scale data, the filtering capability will be useful to reduce information yet guarantee relatively accurate clinical insights.

- **Combining EMA data with machine learning methods may result in enhanced clinical decision-making and just-in-time patient support.** The collection of primarily passive psychosocial and behavioral data streams combined with machine learning methods provides a population-based monitoring systems that can help guide clinical management and just-in-time guidance for self-management problem solving [107].

- **EMA data may be used to create personalized behavioral medicine targeting T1D**. Data from homogeneous sub-groups or even individuals can be used to tailor behavioral treatments and prevent blood glucose excursions and long-term consequences of poor glycemic control for personalized behavioral medicine.

Chapter 3

APPLYING LEARNED FILTERING ARCHITECTURE TO HAND HYGIENE COMPLIANCE MONITORING CASE STUDY

Hospital Acquired Infections (HAIs) are a global concern as they impose significant economic consequences on the healthcare systems. In the U.S. alone, HAIs have cost hospitals an estimated $9.8 billion a year. An effective measure to reduce the spread of HAIs is for Health Care Workers (HCWs) to comply with recommended hand hygiene (HH) guidelines. Unfortunately, HH guideline compliance is currently poor, forcing hospitals to implement controls. The current standard for monitoring compliance is overt direct observation of hand sanitation of HCWs by trained observers, which can be time-consuming, costly, biased, and sporadic. This chapter provides three contributions to research on hand hygiene compliance monitoring: (1) we describe the acquisition and analyses of 60-day real-time location data and handwashing dispenser activation events for the care providers in a 30-bed intensive care unit (ICU), (2) we pose five hypotheses that help identify key characteristics and predictablity of handwashing compliance and validate the hypotheses using the learned filtering architecture presented in Chapter 2, (3) we construct a hand hygiene compliance monitoring app, Hygiene Police (HyPo), based on the insights learned from our empirical studies that can be deployed as a service to alleviate the manual effort, reduce errors, and improve existing compliance monitoring practice.

3.1 Problem Overview

Emerging concerns in healthcare. The World Health Organization defines Hospital Acquired Infections (HAIs) to be either infections acquired by a patient in a hospital or other healthcare facilities that appear after patient discharge or occupational infections acquired by medical workers [108]. HAIs represent significant health problems, with a considerable economic impact on patients and hospitals worldwide, contributing to an increasing hospital re-admission rate. Hospital caregivers are often blamed for patient re-admissions because of their constant exposures to bacteria and diseases, and without good sanitary practices, their contaminated hands can become the primary carriers of infections which are often transmitted to patients or other staff through physical contact.

To prevent the spread of HAIs in healthcare facilities and reduce re-admission rates, healthcare professionals are expected to comply with recommended hand hygiene (HH) guidelines. The current standard practice for compliance monitoring has become employing trained secret shoppers, or covert human auditors, to directly observe and record hand hygiene compliance of medical workers unobstrusively. Unfortunately, this approach is costly and subject to bias [109] due to evidence of Hawthorne effect [110] (the process where human subjects of an experiment alter their behavior due to their awareness of being studied). An alternative approach is to use a real-time location system and smart dispensers to autonomously monitor handwashing compliance by tracking provider location and activation of dispensers.

Contribution. This paper analyzes two months of real-time location data and handwashing dispenser activation events for the care providers in a 30-bed intensive care unit (ICU). The goal of this study is to use machine learning to assess if there are location, time-based, or other behavioral characteristics that predict handwashing non-compliance events in advance. For example, having observed a provider with a non-compliant room entry, we can predict if the same provider will also be non-compliant when exiting the room. Using possible correlating factors to handwashing, we can predict at least one handwashing action ahead of time. We then leverage learned information to construct a hand hygiene compliance monitoring app, Hygiene Police (HyPo), which can be deployed as a service. The goal of this app is to mitigate the laborious and error-prone effort of direct observation and improve compliance by: (1) assisting the direct observation approach by deciding when and where to station manual auditors and (2) improving compliance by providing just-in-time

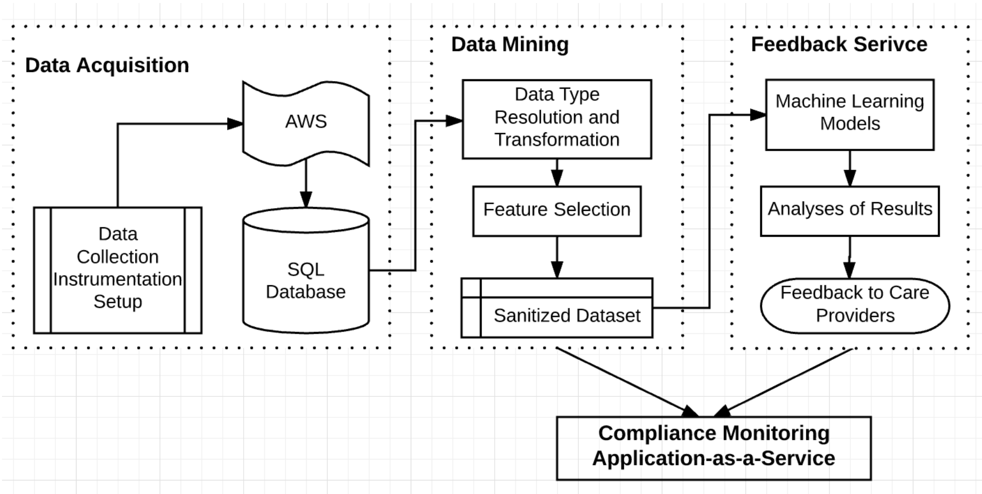

Data Acquisition
AWS
Data Collection Instrumentation Setup
SQL Database
Data Mining
Data Type Resolution and Transformation
Feature Selection
Sanitized Dataset
Feedback Serivce
Machine Learning Models
Analyses of Results
Feedback to Care Providers
Compliance Monitoring Application-as-a-Service

Figure 3.1: The Overall System Flow: from Data Collection to Post-prediction Analysis and Feedback Service.

alerts or potentially recommending training materials to predicted non-compliant staff.

App workflow overview. HyPo is implemented as a Java-based desktop app that communicates to and from Bluetooth Low-Energy (BLE) devices equipped at the facility from our previous study [111]. The schematic in Figure 3.1 depicts the overall app workflow, which is divided into the following three stages (the last two are the core components of HyPo):

1. **Data Acquisition**, where raw data is acquired from the BLE devices.

2. **Data Mining**, where the raw data undergoes a data mining process provisioned by HyPo to produce a set of features that is fed to Feature Selection algorithms to obtain a Sanitized Dataset. The Feature Selection is done to improve the execution performance of the Machine Learning (ML) methods that will follow by determining the most relevant features and removing the others from the Sanitized Dataset.

3. **Feedback Service**, where the ML models are run over the Sanitized Dataset to produce feature set that can be used to provide timely feedback to healthcare providers.

Chapter organization. The remainder of this chapter is organized as follows: Section 3.2 defines key terms frequently referenced throughout the chapter; Section 3.4 poses five hypotheses regarding compliance characteristics we investigated; Section 3.5 describes the data collection instrumentation setup; Section 3.6 details the data preparation and mining process; Section 3.7 evaluates the hypotheses with machine learning predictions and analyses of the preliminary classification results; Section 3.8 describes the construction of HyPo using learned results and how it complements the direct observation approach; and Section 3.9 presents concluding remarks and outlines future extensions of this work.

3.2 Hand Hygiene Compliance Overview

This section defines the following terms that are used frequently in the chapter:

1. **Hand hygiene opportunity**: an opportunity for hand cleaning is presented before each care provider's entry/exit of a patient room.

2. **Hand hygiene/handwashing compliance**: each hand hygiene opportunity requires one hand hygiene action, which should be a corresponding positive (compliance) or negative (non-compliance) action [112].

3. **Entry compliance**: hand hygiene compliance observed at staff's entry to a patient room, determined by *wash on entry*.

4. **Exit compliance**: hand hygiene compliance observed at staff's exit from a patient room, determined by *wash on exit*.

5. **Wash on entry**: hand hygiene action at patient room entry that determines entry compliance, true if performed and false otherwise.

6. **Wash on exit**: hand hygiene action at patient room exit that determines exit compliance, true if performed and false otherwise.

Our previous study collected 60 days of care providers' real-time location and handwashing data, from an intensive care unit (ICU) equipped with 30 beds, and observed two major correlating factors to compliance: (1) entry compliance has an 89% accuracy on predicting exit compliance and (2) exit compliance can predict entry compliance at the next visit (for the same staff) at an accuracy as high as 77%. Likewise, location data was observed to have a minor impact on predicting exit compliance [111].

Based on this previous study, in the HyPo app we compiled the following rules of hand hygiene compliance that ICU staff should abide by:

1. All on-duty staff at the ICU were required to wear a BLE badge.
2. All staff were required to sanitize their hands within a short interval of 2 minutes upon entering a patient room and before exiting the same room.
3. Each compliant action should be associated with an activation of a specific soap dispenser with disinfectant solution against Clostridium difficile, a common HAI spread through physical contact [113]. These dispensers are located both inside and outside each patient room.

These rules only apply to this ICU but can be configured to work with other caregiving settings. The rest of this chapter describes the application of HyPo using the same device-equipped 30-bed ICU from our previous study [111] as an example.

3.3 Related Work

3.3.1 Overview

Due to worldwide high demands of HAI prevention, a number of other researchers have studied approaches to improve hand hygiene compliance. Although the gold standard monitoring method is human-centric [108], [114], a wide rage of studies propose electronic or electronically assisted hand hygiene compliance monitoring and intervention systems [115], [116]. This section compares our work on the HyPo app with common electronic intervention systems including (1) technology-assisted direct human observation, (2) counting systems, and (3) automated monitoring systems.

3.3.1.1 Technology-Assisted Human Observation

Direct observation is the most precise way of controlling compliance. Several studies use technologies such as handheld devices and cameras to aid human observation, aiming at reducing input errors, costs, and time consumption. Handheld devices are used for data entry, and video cameras provide opportunities to reduce the Hawthorne effect and observe locations that are remote or hard to access.

Chen et al [117], used wireless data entry devices and a website to allow human observers to audit compliance. University of North Carolina Hospitals implemented a "clean-in, clean-out" system that allowed covert observers and designated nurses to track compliance using a mobile app and a web portal [118].

Cameras have been used by Armellino [119] to increase compliance in an ICU. The study connected motion sensors near the sinks that would activate cameras being monitored by remote auditors. The study by Davis [120] placed a discreet camera at the entrance of a ward and assessed compliance before and after a sink was placed pointing to the dispenser.

3.3.1.2 Counting Systems

Installing counting devices to measure the remaining sanitation product volume or the number of dispenser activation times is a quiet method that is not subject to the Hawthorne effect. A counter may detect usage patterns and frequency changes.

Marras [121] used dispenser counters along with direct observation to assess whether positive deviance in hand hygiene behaviors could have an impact on reducing HAIs. A downside to this approach, however, is that counter systems cannot tell who used the dispensers and therefore are unable to evaluate compliance by itself. Morgan et al [122] provided evidence to support the claim that dispenser usage data could be more reliable than direct human observation to estimate hand hygiene compliance.

3.3.1.3 Automated monitoring systems using wearables

Many automated monitoring systems are capable of producing feedback or reminders in real or near real time without human intervention, similar to our approach.

Fakhry used a motion-triggered system with audible hand washing reminders at each medical department entrance [123]. Sahud and Bhanot developed an electronic hand hygiene feedback device that reports real-time compliance rate on a liquid-crystal display visible to all staff in the intervention unit [124]. Edmond et al installed a sensor network using a credit-card sized sensor badge on each alcohol dispenser, which when not activated on room entry or exit beeped with a red indicating light [125]. Similarly, Marra et al employed a wireless network with sensors on the alcohol dispensers that provide real-time flashing light feedback to HCWs for hygiene activity detection [126]. Most recently, Ellison et al proposed a prospective electronic hand hygiene room entry/exit audible reminder system [127] that provides a combination of 24-hour-a-day recording of hand hygiene activities and real-time computer monitor performance feedback.

3.3.2 Gaps in Existing Research

All prior research we reviewed collected data to propose strategies that increased hand hygiene performance or gather conclusions regarding the efficacy of a specific approach. Technology-assisted human observation methods and counting systems still require human interaction and can bias the results, as the medical workers know they are being directly observed. Moreover, audits require trained personnel who are regularly monitored to ensure quality control. Although automated monitoring systems using wearables are able to provide reminders to providers, these approaches respond to already detected non-compliance behavior by notifying appropriate caregivers aftermath, which may improve compliance overall but does not prevent non-compliant behavior.

3.4 Summary of Hypotheses

This section poses five hypotheses that leverages the Learned Filtering Architecture described in Section 2.3 to identify key characteristics and predictablity of handwashing compliance *Entry/Exit compliance* is hand hygiene compliance observed at each caregiver's entry or exit to a patient room, determined by *wash on entry/exit*. To predict compliance we perform a binary classification of handwashing actions using features of the movement and handwashing history of a provider. Below we postulate how to evaluate these handwashing classifiers based on different features of a provider's movements and compliance history.

Hypothesis 1: Handwashing on room entry is indicative of washing on exit. Most auditing approaches evaluate handwashing behavior observed outside of a patient room, which may only show an entry or exit wash (*e.g.*, a provider may wash their hands inside the room on entry and outside the room on exit). An important question is how predictive observing one of the washing events is in predicting the other (*e.g.*, if a human auditor only sees a wash on exit, what does this tell us about wash on entry?). We hypothesize that handwashing on entry is indicative of washing on exit. Handwashing can be a habitual—and thus predictable—behavior for hospital caregivers, depending on whether they abide by hand hygiene guidelines.

Hypothesis 2: Time-related features may be indicative of handwashing. For instance, compliance may decrease when patients are asleep between midnight and 5am due to these likely reasons: (1) care providers have limited physical contact with patients, hence less need to sanitize, (2) to reduce noise from activating the dispensers that may disturb patients, and (3) reduced Hawthorne effect since patients are not awake to observe hand hygiene compliance.

Hypothesis 3: Location may affect handwashing behavior. We hypothesize that caregivers' compliance may be affected by which patient room they visit. The study in [110] recognizes the Hawthrone effect with the standard direct compliance observation approach. Likewise, care providers may perform better sanitation under observation when visiting locations that are clearly in view of other staff or supervisors, such as rooms closest to the nurses' stations.

Hypothesis 4: Staff's recent wash in/out behavior may affect entry/exit compliance. We speculate that if previously visited patients were infectious, then it is highly likely that the staff would wash their hands more frequently. Conversely, if these patients were *not* infectious, they may feel there is less need for hand hygiene. Previous handwashing behavior may therefore indicate current compliance.

Hypothesis 5: There may be other features that are possibly predictive of compliance. We postulate that the features selected based on our intuition may have excluded other correlating factors of compliance. To find other possible predictors, we therefore use *feature selection*, which is the process of selecting the most relevant subset of predictors for constructing classifiers.

3.5 Data Acquisition

This section describes the data acquisition process, where real-time location data and handwashing station activation data is recorded, and then provides an overview of the essential data fields extracted from the collection. The process described in this section is one approach of obtaining the hand hygiene compliance data to provide input for our app, but it is by no means the only option to acquire this type of data.

3.5.1 Instrumentation Configurations

The ICU with HyPo deployment was equipped with a Bluetooth Low-Energy (BLE) indoor positioning system that provides room-level accuracy for reporting staff locations in real-time. The system produced the location data for all staff with BLE badges.

The ICU also deployed Gojo Active Monitoring handwashing stations, which record each dispenser activation. These activation events were then combined with real-time location data to track individual staff handwashing compliance. The system expected to receive at least one handwashing event from either a sanitation station inside of the room or a station immediately outside the room within two minutes prior to entry, abiding the facility rules described in Section 3.2. Similarly, two minutes before or after room exit, the system expected one handwashing event from either sanitation stations.

Overall, the dataset collected at the studied ICU contains 8 weeks of events recording activities from 180+ soap dispensers activated by 60 badged nurses 24 hours a day. All raw event data was streamed to a data storage on Amazon Web Services (AWS), which was post processed and output to a SQL database. We then extrapolated the data fields of interest for compliance predictions and analyses.

3.5.2 Dataset Limitations

Although real-time location data was acquired and handwashing station activations recorded at the ICU, the dataset was still an estimate rather than a ground truth of hand hygiene compliance. The dataset collected has a number of key threats to validity as described next

Not all staff wore their BLE badges at all times. The main threat to validity of our work is that we based our findings upon some assumptions made about the data. For instance, we performed analyses on the data assuming that all on duty staff were using their badges at all time. In practice, however, some of staff were sporadically observed without badges. To minimize the impact of this behavior in our findings, we used location data to filter out dispenser activation events not associated with nearby caregivers, retaining all events that were associated with only badged staff.

The system could not differentiate activations from badged vs. non-badged visitors/staff. Unfortunately, there is also the possibility that a staff member without a badge activated the handwashing station while staying in the same room with another badged staff, making the system wrongly assign the event to the staff wearing the badge. Nevertheless, in our analysis of the data, we found it was uncommon for two (or more) caregivers to remain in the same room at the same time. We therefore believe these cases would only marginally skew our findings.

Subsets of the monitoring equipment went offline. Finally, we observed that some monitoring equipment were offline at some intervals. Although, the offline devices only prevented data capturing in certain rooms and did not affect our compliance observations in other patient rooms.

Nonetheless, we did not consider these limitations as fatal to our study results because we could either easily eliminate the data entries associated with these threats or discard the marginal impact that the threats had on our findings.

3.5.3 Dataset Schema

From the SQL database we obtained an initial dataset by omitting certain data fields with extraneous information, such as device IDs of the wearable badges, internally-used identifiers of the patient rooms, etc. The data fields associated to each patient room visit event that we deemed essential thus extracted from the database include:

1. *Staff ID* - ID of badge worn by the staff who has been associated with a patient room visit
2. *Location* - patient room number visited by the badged staff
3. *Entry time* - timestamp (in CDT) at which the badged nurse entered the patient room
4. *Exit time* - timestamp (in CDT) at which the badged nurse exited the patient room
5. *Wash on entry* - a boolean value indicating whether the staff properly performed hand hygiene on patient room entry
6. *Wash on exit* - a boolean value indicating if the staff properly performed handwashing on patient room exit
7. *Duration* - for how long (in milliseconds) the staff was in the patient room

3.6 Data Preparation

This section discusses how we prepared the collected data to maximize the utilization of our machine learning classifiers, which is an important capability offered by HyPo. This process employed to assist the analyses and characterization of hand hygiene compliance. Other influencing factors of hand hygiene compliance may be discovered as more relevant data becomes available, such as patient admittance details, medical records of admitted patients, facility regulations of compliance, etc.

Despite the specificity of the dataset used throughout this chapter, the data mining process provided by HyPo as described below can be generalized to support transformations of different forms of data collected in other facilities.

Most machine learning (ML) classifiers yield better results when the input dataset is structured in certain ways. For example, suppose we want to know if the day of week (Monday to Sunday) influences compliance, some ML classifiers will yield better results if we express date as a set of integers ranging from 1 to 7, as opposed to a real continuous stream of timestamps expressed in milliseconds.

As another example, our location data consists of room numbers, which provides little information regarding spatial distribution of the rooms. If we want to know whether compliance decreases in nearby locations, we must first transform the room numbers into coordinates on the facility's floor plan, for instance.

To obtain a transformed schema that can be better handled by our classifiers, we took the collected dataset and performed the following transformations over it:

1. We converted all event data from the original timestamp format into an integer field with range 1 to 7 to represent day of week, an integer field with range 1 to 4 to represent time of day in morning, afternoon, evening, bedtime, and another integer data field of 0-23 to represent hour of day. The numeric representations of the original nominal time stamp data will allow our classifiers to achieve higher classification

accuracy.

2. We mapped each patient room on the ICU floor plan to a set of *x* and *y* coordinates to identify the spacial location. We then extended each entry in the dataset to include these corresponding coordinates of the patient room.

3. For each data point we added new fields to include the previous record of the corresponding badged staff's handwashing data, *i.e.*, duration, location, washed on entry, and washed on exit. To ensure data integrity, we removed all entries that did not have previous handwashing records.

As a result of these transformations, we obtained a new schema consisting of a minimal set of features that our application expects to receive for best accuracy:

1. staff ID - integer

2. location (room number) - integer

3. washed on entry - TRUE/FALSE

4. washed on exit - TRUE/FALSE

5. duration (s) - length of patient room visit in seconds, integer

6. entry hour - hour of day on room entry, 0-23

7. exit hour - hour of day on room exit, 0-23

8. entry time - time of day on recorded room entry in Morning (1), Afternoon (2), Evening (3), and Bedtime (4)

9. exit time - time of day on recorded patient room exit, 1-4

10. entry day of week - day of week on recorded patient room entry, 1-7

11. exit day of week - day of week on room exit, 1-7

12. location X coordinate - x coordinate of patient room on the ICU floor plan
13. location Y coordinate - y coordinate of patient room on the ICU floor plan
14. previous duration (s) - duration of the same staff's previous patient room visit in seconds
15. previous washed on entry - dispenser activation on previous room entry TRUE/-FALSE
16. previous washed on exit - dispenser activation on previous room exit TRUE/FALSE
17. previous location - previously visited patient room number

3.7 Hypotheses Evaluation

In this section we first describe the machine learning models employed and then evaluate each of the hypotheses by setting up machine learning experiments and analyzing the results. The same machine learning models are also used to construct our HyPo app.

3.7.1 Overview of Machine Learning Models

After restructuring and sanitizing the data collected, we split the data to 65% for training, 10% for cross validation, and the remaining 25% for testing the ML models. Based on the compliance prediction observations from the previous study in [111], we employed the top three classifiers, one from Weka [128] and two deep nets from DeepLearning4J (DL4J) [129] to serve as our models for classifying *washed on entry* and *washed on exit*. HyPo then uses the results with highest accuracy.

- The **Sequential Minimal Optimization** (SMO) implementation of the Support Vector Machine (SVM), which uses heuristics to partition the training problem into smaller sub-problems and uses pairwise linear regression to classify. This method is usually resilient to data overfitting and by default normalizes the input data [130].

- The **Feed-Forward Neural Network** (FFNN), which is a one direction (from input to output) artificial neural network that performs classifications based on weight calculations of the network nodes [131]. Using the DL4J Java library, we developed a 3-layer FFNN with a random seed of 6, 1000 iterations, a learning rate of 0.1, and the Stochastic gradient descent optimization algorithm [132].
- The **Recurrent Neural Network** (RNN), which has a feedback loop whereby the immediately previous step's output is fed back to the net to affect the outcome of the current step. We used a 3-layer RNN with two Graves' Long Short-Term Memory (LSTM) layers [133] (input and hidden) and an output layer along with the same parameters as the FFNN.

Training models with all features. As a first step we examined how well handwashing can be predicted at least one step in advance (*e.g.*, if a care provider washed in on entry to a patient room, can we predict their wash out behavior). We therefore trained the ML models with all features in the dataset. The classification results are shown in Table 3.1 with a consistently high accuracy at 80%+ and other metrics above 0.8. These results indicate that some factors can be predictive of compliance. To identify the specifics, we conducted the following experiments to evaluate the hypotheses described in Section 3.4.

3.7.2 Hypotheses Evaluations

3.7.2.1 Evaluating Hypothesis 1: Handwashing on room entry is indicative of washing on exit.

Experiment setup. We prepared two datasets for each class variable with one set including the counterpart class variable (*i.e.*, dataset with 16 features) and the other excluding it (*i.e.*, data with 15 features). To obtain the second set of training and test data, we applied an unsupervised remove attribute filter from the Weka library to remove the class variable not being predicted.

Table 3.1: Entry and Exit Compliance Classification Results Using All Features in the Dataset

	Class: Washed on Entry				
Classifier	Accuracy	Precision	Recall	F-Score	AUC
RandomForest	89.20%	0.896	0.892	0.893	0.927
SMO	89.35%	0.895	0.893	0.894	0.878
NaiveBayes	82.25%	0.858	0.822	0.829	0.907
FFNN	90.00%	0.879	0.878	0.877	0.869
RNN	91.20%	0.893	0.908	0.901	0.9

	Class: Washed on Exit				
Classifier	Accuracy	Precision	Recall	F-Score	AUC
RandomForest	88.83%	0.889	0.888	0.889	0.922
SMO	89.35%	0.893	0.893	0.893	0.869
NaiveBayes	79.88%	0.838	0.799	0.806	0.898
FFNN	88.80%	0.858	0.875	0.866	0.86
RNN	88.40%	0.861	0.864	0.862	0.853

Results. Fig. 3.1 shows the classification results produced using the dataset with 16 features, with a consistently high accuracy across classifiers at an average of 89% for *wash on entry* and 87% for *wash on exit*. Results in Table 3.2 correspond to the dataset with 15 features with an average *wash on entry* prediction accuracy of 75% and *wash on exit* of 73.5%.

Analysis of results. The overall classification accuracy of *wash on entry* is much higher when its counterpart, *wash on exit*, is taken into account and vice versa, meaning that *wash on entry* is highly predictive of *wash on exit*. With a provider's entry compliance, therefore, if they are predicted non-compliant on room exit, we can provide a hand hygiene reminder to the provider.

3.7.2.2 Evaluating Hypothesis 2: Time-related features may be indicative of hand-washing.

Experiment setup. For this study, we applied Weka's remove attribute filter to remove all features unrelated to time from the dataset and fed the generated dataset to the ML

Table 3.2: Compliance Prediction Results excluding the Counterpart Class Variable.

	Class: Washed on Entry				
Classifier	Accuracy	Precision	Recall	F-Score	AUC
RandomForest	69.08%	0.743	0.691	0.704	0.743
SMO	75.74%	0.76	0.757	0.759	0.713
NaiveBayes	69.38%	0.763	0.694	0.707	0.794
FFNN	79.20%	0.733	0.708	0.72	0.789
RNN	76.80%	0.706	0.721	0.713	0.782

	Class: Washed on Exit				
Classifier	Accuracy	Precision	Recall	F-Score	AUC
RandomForest	67.75%	0.72	0.678	0.689	0.709
SMO	74.56%	0.746	0.746	0.746	0.7
NaiveBayes	68.42%	0.76	0.684	0.697	0.786
FFNN	78.40%	0.726	0.734	0.73	0.699
RNN	76.00%	0.72	0.7	0.71	0.722

classifiers.

Results. The results shown in Table 3.3 have 60%+ accuracy in most cases for both class variables. Specifically, deep nets and SMO models achieved prediction accuracies around 71% for *wash on entry* and 69% for *wash on exit*.

Analysis of results. A closer analysis of the classification result metrics indicates that despite the classification accuracy being acceptable, the AUC (a valuable metric for evaluating classification) is around 0.5, meaning that the results are no better than random guesses. This result suggests that time factors have little impact on determining handwashing and cannot be used to forecast handwashing.

3.7.2.3 Evaluating Hypothesis 3: Location may affect handwashing behavior.

Experiment setup. Similar to the setup when evaluating Hypothesis 2, we altered the original dataset using Weka's remove attribute filter to exclude data unrelated to location information.

Results. The results shown in Table 3.4 have accuracies above 65% in all cases for both class variables. In particular, deep net ML models achieved an average prediction accuracy

Table 3.3: Compliance Classification Results Based on Time-related Features.

	Class: Washed on Entry				
Classifier	Accuracy	Precision	Recall	F-Score	AUC
RandomForest	61.46%	0.617	0.615	0.616	0.539
SMO	70.64%	0.499	0.706	0.585	0.5
NaiveBayes	66.12%	0.628	0.661	0.639	0.572
FFNN	70.80%	0.616	0.552	0.583	0.563
RNN	72.00%	0.609	0.542	0.574	0.551

	Class: Washed on Exit				
Classifier	Accuracy	Precision	Recall	F-Score	AUC
RandomForest	59.76%	0.597	0.598	0.597	0.524
SMO	69.45%	0.482	0.695	0.569	0.5
NaiveBayes	64.79%	0.622	0.648	0.631	0.587
FFNN	68.40%	0.59	0.532	0.559	0.52
RNN	70.00%	0.62	0.54	0.577	0.523

of 75% for *wash on entry* and 73% for *wash on exit*.

Analysis of results. The classification results output by the deep net ML models are more optimistic and consistent with medium accuracy. We therefore infer that location, unlike time-related factors, has more of an impact on predicting handwashing on entry and exit, although not as indicative as the class variables of each other.

Table 3.4: Compliance Classification Results Based on Location-related Features.

	Class: Washed on Entry				
Classifier	Accuracy	Precision	Recall	F-Score	AUC
RandomForest	68.57%	0.746	0.686	0.699	0.746
SMO	65.16%	0.78	0.652	0.665	0.717
NaiveBayes	65.90%	0.774	0.659	0.673	0.707
FFNN	75.20%	0.769	0.591	0.669	0.723
RNN	74.80%	0.766	0.569	0.653	0.71

	Class: Washed on Exit				
Classifier	Accuracy	Precision	Recall	F-Score	AUC
RandomForest	68.71%	0.739	0.687	0.699	0.733
SMO	65.01%	0.766	0.65	0.662	0.709
NaiveBayes	65.75%	0.762	0.658	0.67	0.704
FFNN	74.80%	0.662	0.55	0.601	0.642
RNN	71.60%	0.682	0.576	0.625	0.65

3.7.2.4 Evaluating Hypothesis 4: Staff's recent wash in/out behavior may affect entry/exit compliance.

Experiment setup. To include the previous wash in/out event, we sorted the dataset by *staff ID* and then *timestamp*. For each data entry we then added the immediate previous *wash on entry/exit* associated with the same staff and discarded all entries without any previous data.

Results. The classification results are shown in Table 3.5. Most classifiers produced an accuracy of 74%+ for both class variables.

Analysis of results. Most ML classifiers produced consistently optimistic prediction results of both class variables, and all performance metrics are above a confident value of 0.7. This result suggests that a provider's most recent handwashing behavior can be useful for predicting *wash on entry/exit* of the next visit.

Table 3.5: Predictions of Compliance Using Previous Handwashing Data

	Class: Washed on Entry				
Classifier	Accuracy	Precision	Recall	F-Score	AUC
RandomForest	64.05%	0.692	0.641	0.655	0.692
SMO	75.74%	0.76	0.757	0.759	0.713
NaiveBayes	75.07%	0.757	0.751	0.753	0.795
FFNN	77.60%	0.709	0.721	0.715	0.781
RNN	77.20%	0.74	0.729	0.734	0.774

	Class: Washed on Exit				
Classifier	Accuracy	Precision	Recall	F-Score	AUC
RandomForest	63.54%	0.681	0.635	0.648	0.682
SMO	74.56%	0.746	0.746	0.746	0.7
NaiveBayes	74.04%	0.745	0.74	0.742	0.784
FFNN	77.20%	0.735	0.706	0.72	0.763
RNN	78.80%	0.725	0.732	0.729	0.78

3.7.2.5 Evaluating Hypothesis 5: There may be other features that are possibly predictive of compliance.

Experiment setup. In this experiment we executed a feature selection process to automatically select feature subsets in our transformed data that best (1) reduced overfitting of data, (2) improved classification accuracy, and (3) decreased model training time [134]. Although we do not have a significantly large feature list produced for this ICU, it is still useful to apply this technique to select the most relevant subsets of features to help produce the most accurate feedback in the next step.

To automatically select features from the transformed dataset, HyPo applies a supervised attribute selection filter from the open source Weka Java library [128]. The filter is composed of two pieces: (1) a feature *Evaluator* to determine how features are evaluated and (2) a *Search Method* to navigate the feature's search space. Our app runs feature selection using the following pairs of *Evaluators* and *Search Methods*, as shown in Table 3.6:

1. *Evaluator*: CfsSubsetEval that evaluates a subset of features by considering each feature's predictive ability and the degree of redundancy between them.

   Search Method: GreedyStepwise with a backward search through the space of attribute subsets.

2. *Evaluator*: InfoGainAttributeEval that evaluates an attribute's worth by measuring the information gain with respect to the class variable to classify.

   Search Method: Ranker that ranks features by their individual evaluations with an optional parameter of 6 features in the output subset

3. *Evaluator*: WrapperSubsetEval [135] with NaiveBayes [136] as the basic learning scheme and a 10-fold cross validation to use for estimating accuracy.

   Search Method: GeneticSearch that performs a search using the simple genetic algorithm [137]

Table 3.6: Evaluator and Search Method Pairs Used in Feature Selection

Evaluator	Search Method
CfsSubsetEval	GreedyStepwise
InfoGainAttributeEval	Ranker
WrapperSubsetEval	GeneticSearch

Results. The features selected for class *wash on entry* are *wash on exit*, *previous wash on exit*, and *location x coordinate* and *wash on entry* for class *wash on exit*. The classification results are shown in Table 3.7 outputting an average accuracy of 88.5% and 87% for both classes.

Analysis of results. The results validated our previous observations made in Hypotheses 1, 2, 4 of *wash on entry* with a specific location factor being *location x coordinate* and Hypothesis 1 of *wash on exit*. They also indicate that no other feature can characterize compliance behavior.

Table 3.7: Compliance Predicted with Automatically Selected Features

	Class: Washed on Entry				
Classifier	Accuracy	Precision	Recall	F-Score	AUC
RandomForest	89.05%	0.891	0.892	0.925	0.692
SMO	89.35%	0.893	0.894	0.878	0.713
NaiveBayes	82.25%	0.822	0.829	0.907	0.795
FFNN	90.00%	0.878	0.877	0.869	0.781
RNN	91.20%	0.908	0.901	0.9	0.774

	Class: Washed on Exit				
Classifier	Accuracy	Precision	Recall	F-Score	AUC
RandomForest	89.05%	0.891	0.891	0.922	0.682
SMO	89.35%	0.893	0.893	0.869	0.7
NaiveBayes	79.88%	0.799	0.806	0.898	0.784
FFNN	88.80%	0.875	0.866	0.86	0.763
RNN	88.40%	0.864	0.862	0.853	0.78

3.8 Our Contribution: Feedback Service

This section presents HyPo's feedback service built upon learned results from the hypotheses evaluations in Section 3.7. Specifically, we demonstrate HyPo's capability to provide timely feedback, complement the direct observation approach, and provide appropriate training materials when necessary.

3.8.1 Just-in-Time Alerting

HyPo can provide just-in-time alerting to remind HCWs to perform hand hygiene when they are predicted not to comply, using either a *singular* prediction or a *chain-prediction* scheme, depending on if there is adequate time to provide such notifications between each hand hygiene opportunity.

Suppose that HyPo has just observed a staff nurse's compliance on a patient room entry, then the ML classifiers will predict the same staff's exit compliance. For instance, if the staff is predicted to be non-compliant, an alert of red flashing light can be sent to either the wearable badge or the device at the appropriate dispenser activation station as a reminder to the staff; otherwise, no alert is necessary.

If duration of the visit is too short of an interval to send the notification signal to the devices, then we can use the probability chain rule [138] to provide a backup alert to the same staff if necessary. In this case, the ML models will use the predicted entry compliance for the current visit (from the staff's exit compliance of the previous visit) to determine exit compliance of the current visit at a probability of 89% * 77% = 69%. It is less ideal, but the likelihood of the visit interval being too short is minimal because the grace period for compliance is set at two minutes, and if a room visit is within two minutes, hand hygiene compliance is not required.

3.8.2 Assist Direct Observation

The compliance prediction results can also be used to assist the current standard practice of direct observation. With predicted non-compliance reoccurring at a certain location (*i.e.*, a patient room), HyPo can deploy a human auditor (*e.g.*, by sending a notification) to observe compliance at the location that should be given most attention.

3.8.3 Recommend Training Material

If a staff member is frequently predicted as non-compliant over a long observation period, HyPo (with integrated email capabilities) can recommend hand hygiene guidelines or appropriate training materials to the staff via email. The goal is to improve compliance on an individual basis.

3.9 Conclusion

This chapter presented a hand hygiene monitoring app called Hygiene Police (HyPo) that can be deployed as a service to complement the current monitoring approach and improve compliance. We showed an example data collection process taken place at a 30-bed ICU where we acquired the handwashing compliance data. We also described the data transformation process HyPo employs to maximize the utilization of the selected machine learning (ML) classifiers.

Combining the results of real-time compliance predictions using the correlations identified from evaluating our hypotheses, HyPo can provide three types of services: (1) just-in-time alerting to remind predicted non-compliant staff to perform hand hygiene, (2) recommending training materials to habitually non-compliant staff via email, and (3) assisting the direct observation approach by deploying human auditors at the opportune time and place when and where non-compliance is frequently predicted to occur. We also compared our app to related research work and found that our approach *predicted* future compliance behavior instead of *reacted* to non-compliance as in other approaches. Our methodology using ML algorithms is unique and is the only work that evaluates ML prediction capabilities in this domain.

Chapter 4

FHIRCHAIN: APPLYING BLOCKCHAIN TO SECURELY AND SCALABLY SHARE CLINICAL DATA

Secure and scalable data sharing is essential for integrating the new generation of high-frequency and also for collaborative clinical decision making. Conventional clinical data efforts are often siloed, however, which creates barriers to efficient information exchange and impedes effective treatment decision made for patients. This chapter describes our study of applying blockchain technology to clinical data sharing in the context of technical requirements defined in the “Shared Nationwide Interoperability Roadmap” from the *Office of the National Coordinator for Health Information Technology* (ONC). Specifically, we provide four contribution to this research: (1) we analyze the ONC technical requirements and their implications for blockchain-based health IT systems, (2) we present FHIRChain, which is a blockchain-based architecture designed to meet ONC requirements by encapsulating the HL7 *Fast Healthcare Interoperability Resources* (FHIR) standard for clinical data exchange, (3) we demonstrate a FHIRChain-based decentralized app using digital health identities to authenticate participants in a case study of collaborative decision making in a remote tumor board, and (4) we highlight key lessons learned from our case study using decentralized blockchain technology.

4.1 Problem Overview

The importance of data sharing in collaborative decision making. Secure and scalable data sharing is essential to provide effective collaborative treatment and care decisions for patients. Patients visit many different care providers’ offices during their lifetime. These providers should be able to exchange health information about their patients in a timely and privacy-sensitive manner to ensure they have the most up-to-date knowledge about patient health conditions.

As another example, in telemedicine practice [139]—where patients are remotely diagnosed and treated—the ability to exchange data securely and scalably is particularly important for enabling clinical communications regarding remote patient cases. Data sharing helps improve diagnostic accuracy [140] by gathering confirmations or recommendations from a group of medical experts, as well as preventing inadequacies [141] and errors in treatment plan and medication [142, 143]. Likewise, aggregated intelligence and insights [144, 145, 146] helps clinicians understand patient needs and in turn apply more effective in-person and remote treatments.

Data sharing is also essential in cancer care, where groups of physicians with different specialties form tumor boards. These boards meet on a regular basis to analyze cancer cases, exchange knowledge, and collaboratively create effective treatment and care plans for each patient [147]. Regional virtual tumor boards are also being implemented via telemedicine [148, 149] for institutions that lack inter-specialty cancer care due to limited oncology expertise and resources [150].

Administrative support for coordinating health IT efforts. The Office of the National Coordinator for Health Information Technology (ONC) is a division of the Office of the Secretary within the United States Department of Health and Human Services. ONC is the principal federal entity to oversee and coordinate health IT efforts, including the development of interoperable, privacy-preserving, and secure nationwide health information systems and the promotion of widespread, meaningful use of health IT to improve healthcare.

Data sharing barriers to collaborative decision making. In practice, many barriers exist in the technical infrastructure of health IT systems today that impede the secure and scalable data sharing across institutions, thereby limiting support for collaborative clinical decision making. Examples of such barriers include the following:

- **Security and privacy concerns.** Despite the need for data sharing, concerns remain regarding protection of patient identity and confidentiality [151]. For instance, virtual medical interactions may increase the risk of clinical data breaches due to electronic transmission of data without highly secure infrastructures in place, which can result in severe financial and legal consequences [152]. Likewise, medical identity theft may occur more frequently, especially in telemedicine [151], where virtual (*i.e.*, networked) interactions are replacing face-to-face interactions between providers and patients.

- **Lack of trust relationships between healthcare entities**. Trust relationships between healthcare entities [153] (*e.g.*, care providers and/or healthcare institutions) are an important precondition to digital communications [154] and data sharing in the absence of custody over shared data. Larger healthcare facilities (such as enterprise hospital systems) may be networked [155], but communications between private or smaller practices may not be established.

- **Scalability concerns**. Large-scale datasets may be hard to transmit electronically due to restrictive firewall settings or limitations in bandwidth (which is still common in rural areas [156]). Lack of scalability can also impact overall system response time and data transaction speed [157].

- **Lack of interoperable data standards enforcement**. Without the enforcement of existing interoperable data standards (such as HL7's *Fast Healthcare Interoperability Resources* (FHIR)[28] for shared data), health data can vary in formats and structures that are hard to interpret and integrate into other systems [158].

What is needed, therefore, is a standards-based architecture that can integrate with existing health IT systems (and related mobile apps) to enable secure and scalable clinical data sharing for improving continuous, collaborative decision support.

Research focus and contributions → Architectural considerations for secure and scalable blockchain-based clinical data sharing systems. Blockchain technologies have recently been touted [159, 160, 161] as a technical infrastructure to support clinical data sharing that promotes care coordination. A key property of blockchains is their support for ”trustless disintermediation.” This property enables multiple parties who do not fully trust each other to exchange digital assets (such as the Bitcoin cryptocurrency [23]), while still protecting their sensitive, personal data from each other.

Our prior work [162] provided evaluation recommendations for blockchain-based health IT solutions on a high-level, focusing on common software patterns [106] that can be applied to improve the design of blockchain-based health apps. This chapter examines previously unexplored research topics related to alleviating the data sharing barriers described above, namely: *what are the architectural consideration associated with properly leveraging blockchain technologies to securely and scalably share healthcare data for improving collaborative clinical decision support*?

This chapter provides the following contributions to using blockchain technologies in clinical data sharing to improve collaborative decision support:

- We summarize key technical requirements defined in the “Shared Nationwide Interoperability Roadmap” [27] drafted by the *Office of the National Coordinator for Health Information Technology* (ONC) for creating an interoperable health IT system and analyze the implications for blockchain-based system design.

- We present the structure and funcationality of a blockchain-based architecture called FHIRChain that meets the ONC technical requirements for sharing clinical data between distributed providers. FHIRChain uses HL7’s FHIR data elements (which have uniquely identifying tags) in conjunction with a token-based design to exchange data resources in a decentralized and verifiable manner without requiring duplicated efforts of uploading data to a centralized repository.

- We demonstrate a FHIRChain-based *decentralized app* (DApp) that uses digital health identities to more readily authenticate participants and manage data access authorizations in a case study of clinical data sharing in remote cancer care. This DApp enables users to share specific and structured pieces of information (rather than an entire document), thereby increasing the readability of data and flexibility of sharing options.

- We highlight key lessons learned from our case study and discuss how our FHIRChain-based DApp can be further extended to support other technical requirements for improving advanced healthcare interoperability issues, such as coordinating other stakeholders (*e.g.*, insurance companies and pharmacies) across the industry and providing patients with direct and secure access to their own medical records. We also explore the data exchange issues that blockchains cannot yet address effectively, including semantic interoperability, healthcare malpractice, and unethical use of the data, which remain as future research problems in this space.

4.2 Overview of Blockchain

The most popular application of blockchain is the Bitcoin blockchain [23], which is a public distributed ledger designed to support financial transactions via the Bitcoin cryptocurrency. This blockchain operates in a peer-to-peer fashion with all transactions distributed to each network maintainer node (called a “miner”) for verification and admittance onto the blockchain. These miners validate available transactions and group them into blocks, as shown in Figure 4.1. Miners then compete in solving a computationally expensive cryptographic puzzle, known as “proof-of-work,” where a targeted hash value associated with the last valid block in the chain is calculated. The first miner to solve this puzzle receives a reward (*i.e.*, an amount of Bitcoin) and appends their block of validated transactions to the blockchain sequence.

The Bitcoin blockchain uses the proof-of-work process outlined above to achieve consensus (agreement on the shared state and order of transactions) by

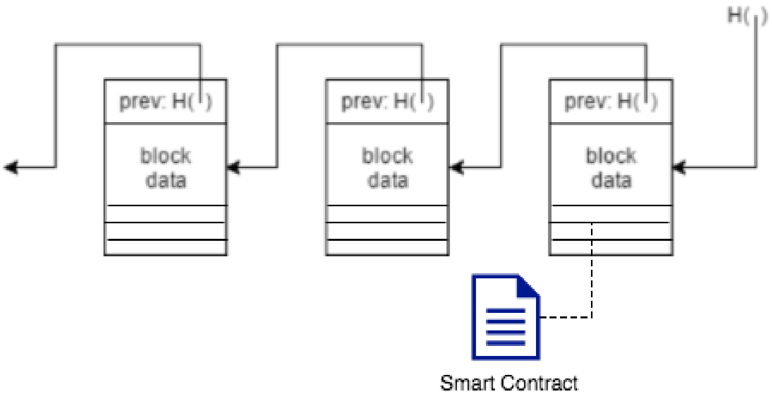


Figure 4.1: The Blockchain Structure: a Continuously Growing and Immutable List of Ordered and Validated Transactions

- incentivizing miners to contribute powerful hardware and electricity to the network with small amounts of cryptocurrency as rewards and
- discouraging rogue actors from attempting to manipulate or maliciously control the system.

After a block is added to the blockchain, its transaction history is secured from tampering via cryptography.

The Bitcoin blockchain is the most widely deployed example of this distributed ledger technology. In recent years, however, other types of blockchain technologies have emerged. For example, the Ethereum blockchain [24] provides a more generalized framework via "smart contracts" [163] that allow programs to run on the blockchain and store/retrieve information.

Smart contracts enable code to execute autonomously when certain conditions are met. They can also store information as internal state variables and define custom functions to manipulate or update this state. Operations in smart contracts are published as transactions and thus occur in a globally sequential order, in a similar fashion as shown in Figure 4.1. These operations are deterministic and verifiable by miners in the Ethereum blockchain to ensure their validity.

The mechanisms described above make a blockchain decentralized and immutable, thereby removing the need for a trusted central authority. These properties make blockchain technologies attractive to certain communities of health IT researchers and practitioners as means to improve clinical communications while protecting the privacy of healthcare participants. The remainder of this chapter examines how to effectively leverage blockchains for securely and scalably sharing clinical data that enables collaborative decision support.

4.3 Related Work

4.3.1 Overview

Due to the growing interest in using distribute ledger technologies for health IT systems, related work has explored various blockchain-based design considerations and prototypes. This section summarizes this related work and compares it with our research on FHIRChain and DApps that provide collaborative clinical decision support for remote patients.

4.3.1.1 Conceptual Blockchain-Based Design Considerations

Krawiec et al. [164] presented several existing pain points in current health information exchange systems and the corresponding opportunities provided by blockchain technologies. They also discussed how blockchain can be leveraged in the health IT systems so that patients, health providers, and/or health organisations can collaborate. Nichol et al [165] presented an analysis that assembles concepts in blockchain-related technologies and speculates on how blockchain can be used to solve common interoperability problems facing healthcare.

A team at IBM [166] took a broader approach by highlighting the challenges in the healthcare industry and providing concrete use cases to showcase potential applications of blockchain technologies. Our prior work also provided software design recommendations for creating general blockchain-based health IT systems [106] and proposed assessment metrics for blockchain-based health systems [162], which include a subset of the technical requirements defined in the ONC roadmap. This prior work of ours focused on providing more general or high-level recommendations for developers creating blockchain-based health IT systems.

The review paper by Kuo et al. [167] presented several blockchain applications in healthcare, such as improved medical record management and advanced healthcare data ledger, and their benefits for each described application. They then analyzed key challenges associated with using blockchain technology for healthcare, including issues like confidentiality, scalability, and treat of a 51% attack on the blockchain network. According to the authors, some example implementation techniques that may mitigate the challenges are (1) encryption of sensitive data or dissemination of only meta data and storing sensitive data off-chain to protect confidentiality, (2) keeping only partial, ongoing verified transactions on-chain rather than the entire transaction history to increase scalability of the blockchain network, and (3) the adoption of a virtual private network or HIPAA-compliant components to prevent the 51% attack.

4.3.1.2 Blockchain Prototype Designs

Azaria et al. [161] created a decentralized record management platform that enables patients to access their medical history across multiple providers. This platform used a so-called "permissioned" blockchain (which is only accessible by authorized users, rather than one that is open to the public) to manage authentication, data sharing, and other security properties in the medical domain. Their blockchain design integrated with existing provider data storage to enable interoperability by curating a representation of patient medical records. Medical researchers were incentivized to contribute to mining of the blockchain by collecting aggregated metadata as mining rewards.

Peterson et al. [168] presented a healthcare blockchain also considers the integration with FHIR standards. They proposed a merkle-tree based blockchain system that introduces ”Proof of Interoperability” as the consensus mechanism during block mining. Proof of interoperability is based on conformance to the FHIR protocol, meaning that miners must verify the clinical messages sent to their blockchain to ensure they are interoperable with known structural and semantic standards.

Dubovitskaya et al. [169] also proposed a permissioned blockchain framework on managing and sharing medical records for cancer patient care. Their design employed a membership service to authenticate registered users using a username/password scheme. Patient identity was created via a combination of personally identifying information (including social security number, date of birth, names, and zip code) and encrypted for security. Medical data files were uploaded to a secure cloud server, with their access managed by the blockchain logic.

Unlike other blockchain designs, Gropper’s ”HIE of One” system [170] focused on the creation and use of blockchain-based identities to credential physicians and address the patient matching challenge facing health IT systems. Patients are expected to install a digital wallet on their personal devices to create their blockchain-based IDs, which can then be used to communicate with the rest of the network. Instead of storing patient information, Gropper’s system would consume only the blockchain-based ID and use it to secure and manage access to patient data located in EHR systems.

4.3.2 Gaps in Existing Research

Existing evaluations and prototypes of blockchain-based healthcare architectures have not been analyzed against the technical requirements defined by health experts from the federal government. Traditional methods for identifying healthcare participants do not work well for blockchain-based approaches as they may position sensitive information at higher risks of being stolen due to the openness of data stored on the blockchain. In addition, there has been very limited discussion on potential remedies for lost or stolen identities.

Our goal with FHIRChain is to present a generalized architecture for healthcare data exchange that addresses the gaps in prior research. Particularly, we describe how our FHIRChain-based DApp demonstrates the use of digital health identities that do not directly encode private information and can thus be replaced for lost or stolen identities, even in a blockchain-based system. While our approach is similar to the use of digital IDs in the *HIE of One*[170] system, FHIRChain provides a more streamlined solution. In addition, we incorporate a token-based access exchange mechanism in FHIRChain that conforms with the FHIR clinical data standards. We also leverage public key cryptography to simplify secure authentication and permission authorizations, while simultaneously preventing attackers from obtaining unauthorized data access. FHIRChain differs from related work on blockchain infrastructures and associated consensus mechanisms because it is decoupled from any particular blockchain framework and instead focuses on design decisions of smart contract and other blockchain-interfacing components. The architecture is thus compatible with any existing blockchains that support the execution of smart contracts.

4.4 Technical Requirements for Blockchain-Based Clinical Data Sharing

The "Shared Nationwide Interoperability Roadmap" defines technical requirements and guiding principles for creating interoperable health IT systems [27]. Based on our experiences to date, we contend that crafting a blockchain architecture to meet these requirements necessitates overcoming significant challenges to utilize blockchain technology in healthcare most effectively.

This section first analyzes five key technical requirements fundamental to clinical data sharing systems and then discusses the implications of these requirements on blockchain-based architectures. Sections 4.5 and 4.6 subsequently describe how we developed and applied our FHIRChain blockchain-based architecture to create a decentralized app (DApp) that meets the ONC requirements in the context of collaborative clinical decision making.

4.4.1 Requirement 1: Verifying Identity and Authenticating All Participants

4.4.1.1 ONC requirement summary

The ONC requirements state that an identity ecosystem should be employed to minimize identity theft and provide redress in case of medical identity fraud, while complying with individual privacy regulations. Providers, hospitals, and their health IT systems should be easily identity-proofed and authenticated when exchanging electronic health information. Healthcare systems today, however, lack "consistently applied methods and criteria" for identity proofing and authentication across organizations [27]. For example, different network service providers have different policies or requirements and may not acknowledge the methods applied by other network service providers.

One of the most popular—and least complex—approaches to exchange data is through direct secure messaging [27]. For example, the Direct project [171] was launched to create a standard way for participants to send authenticated, encrypted health information directly to known, trusted recipients over the Internet. Providers or care centers using EHR systems *without* Direct integration, however, cannot benefit from the direct exchange capability.

4.4.1.2 Implications for blockchain-based system design

For a blockchain-based system, storing identification information (such as personal email) directly on-chain is problematic [172]. In particular, a property of blockchains is information "openness," *i.e.*, all data and associated modification records are immutably recorded and publicly available to all network participants. In the case of Bitcoin, data is open to everyone with Internet access [23], whereas in a non-public blockchain (such as a consortium blockchain [24]) data access is limited only to authenticated blockchain participants.

To meet the requirement of openness while complying to health privacy regulations [173], a blockchain-based system should thus support user identity-proofing and authentication while encapsulating sensitive personal information. Section 4.5.2.1 shows how FHIRChain addresses this identifiability and authorization requirement via digital health identities based on public key cryptography [174].

4.4.2 Requirement 2: Storing and Exchanging Data Securely

4.4.2.1 ONC requirement summary

The ONC requirements state that data should be shared securely and privately without unauthorized or unintended alteration, while making the information available to authorized parties. Data encryption is a recommended both when data is sent over networks (data-in-motion) and when it is stored (data-at-rest). Management and distribution of encryption keys must be ”secure and tightly controlled” [27].

4.4.2.2 Implications for blockchain-based system design

There has been recent interest [175, 176] in using blockchain technologies as decentralized storage for encrypted health data. As discussed in Section 4.2, however, the open and transparent nature of blockchain raises privacy concerns when attempting to integrate blockchain into the health IT domain. Although sensitive data can be encrypted, flaws in encryption algorithms or software implementations may expose the data contents in the future. To ensure long-term data security, therefore, a data storage design should be “simple” to minimize software bugs [177], *e.g.*, by not storing sensitive data (encrypted or not) on-chain, yet still enable data flow from one user to another [162].

Another implication of storing data on a blockchain is scalability. All blockchain transactions (such as storing data in a smart contract and modifying the data) and data records are distributed as an entire copy to all blockchain nodes. In a public blockchain, moreover, transaction fees are paid to miners to reward their validation efforts , as described in Section 4.2. As new data is added or modified, each change must be propagated to all nodes, raising scalability challenges and potentially incurring significant long-term operational costs. Section 4.5.2.2 shows how FHIRChain addresses this requirement via a hybrid on-chain/off-chain storage model.

4.4.3 Requirement 3: Consistent Permissioned Access to Data Sources

4.4.3.1 ONC requirement summary

The ONC advocates "computable privacy" that represents and communicates the permission to share and use identifiable health information [27]. Individuals should be able to document their permissions electronically, which are then honored as needed. Permission authorizations to receive or access an individual's clinical data should be accurate and trustworthy, requiring both the data requestor and holder to have a common understanding of what is authorized.

4.4.3.2 Implications for blockchain-based system design

Unfortunately, smart contract operations only occur in the blockchain space to ensure deterministic outcomes. Services (such as OAuth [178]) that exist off the blockchain therefore cannot be used. Given this constraint, incorporating other alternatives to provide data access permissioning should be a key component of a blockchain-based design. Section 4.5.2.3 shows how FHIRChain addresses this requirement via a token-based permission model.

4.4.4 Requirement 4: Applying Consistent Data Formats

4.4.4.1 ONC requirement summary

To satisfy interoperability needs, the ONC requirements state that health IT systems should be implemented with an "intentional movement and bias" [27] toward a clinical data standard identified by ONCs recently finalized *Interoperability Standards Advisory* [179]. The data exchanged should be structured, standardized, and contain discrete (granular [180]) information. Likewise, standards should use metadata to communicate their context along with pieces of structured data.

4.4.4.2 Implications for blockchain-based system design

To provide collaborative clinical decision support, health IT systems must present shared data to clinicians in a structured and readable format [181]. This requirement implies the enforcement of existing, commonly accepted clinical data standard(s), rather than introducing new data exchange formats. Section 4.5.2.4 shows how FHIRChain addresses this requirement by enforcing the FHIR standard.

4.4.5 Requirement 5: Maintaining Modularity

4.4.5.1 ONC requirement summary

The ONC requirements state that since technology inevitably changes over time, health IT system designs should be capable of evolving by maintaining modularity. When divided into connected, modular components, health IT systems become more resilient to change with increased flexibility. In turn, these properties enable the adoption of newer, more efficient technologies over time without rebuilding the entire system.

4.4.5.2 Implications for blockchain-based system design

Modularity requires a carefully crafted design to avoid "information lock-in" due to the immutability of smart contracts. Every change to a smart contract code creates a new contract instance on the blockchain, nullifying previous versions and their data. To minimize dependencies and the need to upgrade, therefore, smart contracts should be loosely coupled with other components in the system. Section 4.5.2.5 shows how FHIRChain addresses this requirement by applying the *model-view-controller* (MVC) pattern [182].

4.5 FHIRChain: a Blockchain-Based Architecture for Clinical Data Sharing

This section first presents an overview of FHIRChain, which is a blockchain-based architecture we designed to meet the ONC requirements for secure and scalable sharing of clinical data described in Section 4.4. We then explain why we made specific architectural decisions in FHIRChain to address each requirement and how they solve the five challenges facing blockchain technology described in Section 4.4.

4.5.1 FHIRChain Overview

Figure 4.2 shows the FHIRChain architecture we devised to address key ONC technical requirements. This architecture provides a general data sharing solution applicable to a wide range of health IT systems. It also serves as the basis for our decentralized app (DApp) prototype describe in Section 4.6, which customizes FHIRChain to support collaborative clinical decision making using a case study of cancer care in telemedicine.

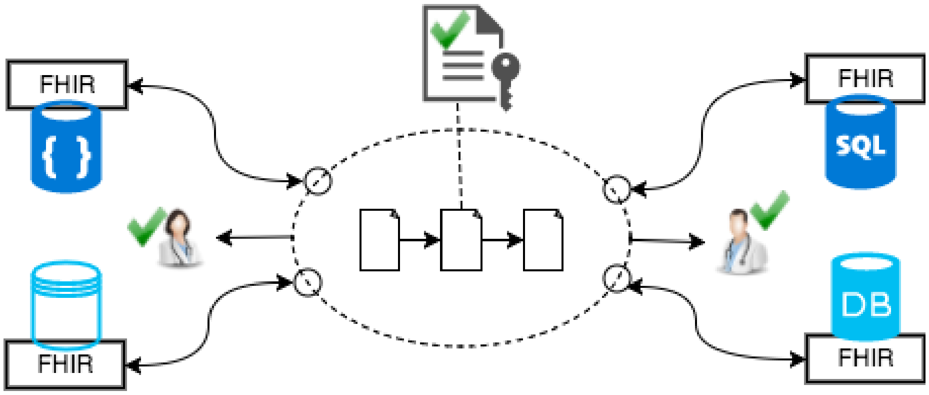


Figure 4.2: Architectural Components in FHIRChain

The dashed ellipse in Figure 4.2 represents a blockchain component that mediates data sharing between collaborating medical professionals (represented by providers with green check marks). Clinical data silos are represented by heterogeneous database symbols, which we normalized with the FHIR standards to enforce a common structure of shared data. Secure database connectors (represented as small circles) connect siloed data sources to the blockchain by exposing secure access tokens to data references that can be obtained only by authorized entities. The secure tokens are recorded in a smart contract (represented by linked documents) for decentralized access and also traceability.

In addition to storing secure access tokens, the smart contract also maintains an immutable timestamped transaction log (represented as a keyed file symbol) of all events related to exchanging and actually consuming these tokens. These logs include specific information regarding what access has been granted to which user by whom, who has consumed which token to access what resource, etc. To ensure the validity of shared data, FHIRChain can be configured to only approve participation from certified clinicians and healthcare organizations with a membership registry.

4.5.2 FHIRChain Architectural Decisions that Address Key ONC Technical Requirements

Below we explain why specific architectural decisions were made to address each ONC requirement presented in Section 4.4.

4.5.2.1 Addressing Requirement 1: Verifying Identity and Authenticating All Participants

Context Blockchains—such as Ethereum and Bitcoin—provide pseudo-anonymous personal accounts (*i.e.*, public addresses composed of random hash values) for users to transact cryptocurrencies. These native identities, however, do not address healthcare requirement for identifiability or authentication of all participants.

Problem By design, public blockchains are globally accessible to anyone with Internet access and allow users to hold any number of blockchain accounts to minimize the identifiability of account holders. This ONC requirement, however, specifies that all U.S. healthcare participants should be identifiable, implying the need for an entirely separate, traceable user base from blockchains' native identities. A key problem is thus how to properly define identities for healthcare users participating in clinical data sharing, while protecting sensitive personal information on the blockchain.

Design choice → use of a digital health identity Inspired by the success of secure shell (SSH) [183] and blockchain address generation mechanism, FHIRChain employs public key cryptography [174] to create and manage health identities. In public key cryptography, a pair of mathematically related public and private keys is used to create digital signatures and encrypt data. Since it is computationally infeasible to obtain the private key given its paired public key, these public keys can be shared freely, thereby allowing users to encrypt content and verify digital signatures. In contrast, private keys are kept secret to ensure only their owners can decrypt content and create digital signatures.

FHIRChain generates a cryptographic public/private key pair (also used for encryption, as described in Section 4.5.2.3) for each participating provider, *e.g.*, in-house providers and remote physicians in telemedicine clinics. The public keys represent users' digital health identities. These identities are recorded in the blockchain for both identity- and tamper-proofing, thereby ensuring that users holding the corresponding private keys can be authenticated to use FHIRChain's data sharing service.

FHIRChain's design applies a smart contract to maintain health users' identifiability without exposing personal information on the blockchain. It also replaces the need for a traditional username/password authentication scheme with the use of a public/private cryptographic key pair for authentication. In a general clinical setting, these digital health identities (*i.e.*, their private keys) would be hard to manage for patients. FHIRChain, however, only creates these identities for clinicians to facilitate data sharing, which enables more effective collaborative decision making for patients.

4.5.2.2 Addressing Requirement 2: Storing and Exchanging Data Securely

Context A key capability offered by blockchains is their support for "trustless transactions between parties who lack trust relationships established between them. Bitcoin is the most common example of this trustless exchange via its native cryptocurrency. Blockchains are peer-to-peer by nature and thus contribute to the ubiquitousness of digital assets being transacted.

Problem Health data represented via digital assets are more complex and harder to share *en masse*. There are also privacy and security concerns associated with its storage in an "open" peer-to-peer system (*i.e.*, public blockchains), such as encryption algorithms applied to protect data being decryptable in the future [162]. A key problem is thus how to design a blockchain-based health IT system so that it balances the need for ubiquitous store and exchange and the concerns regarding privacy of the data and scalability of the system.

Design choice → keeping sensitive data off-chain and exchanging reference pointers on-chain Rather than storing encrypted health data in the blockchain, a more scalable and secure alternative is to store and exchange encrypted metadata referencing protected data (*i.e.*, a reference pointer to a data set), which can be combined with an expiration configuration for short-term data sharing. Exchanging encrypted reference pointers allows providers to maintain their data ownership and choose to share data at will. This technique also prevents an attacker who intercepts the encrypted pointers from obtaining unauthorized data access.

FHIRChain attaches a secure connector to each database, as shown in Figure 4.2. Each connector generates appropriate reference pointers that grant access to the data. These reference pointers are digital health assets that can be transacted ubiquitously with reduced risks of exposing the data.

An added benefit of exchanging metadata *en masse* is more scalability compared to exchanging the original data source. As discussed in Section 4.4.2, each transaction or operation on the blockchain (*e.g.*, querying a smart contract state variable value or updating it) is associated with a small fee paid to the miner for verification and then included onto the blockchain. Transacting these lightweight reference pointers is more efficient in terms of time and cost in production because small changes to data generally require no modifications to reference pointers.

4.5.2.3 Addressing Requirement 3: Permission to Access Data Sources

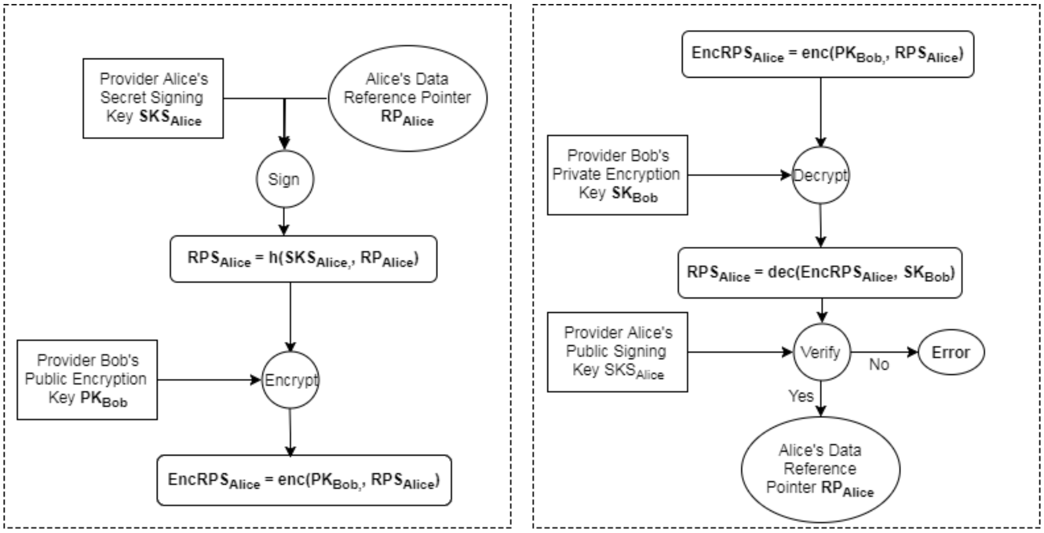
Context Data references can be stored on the blockchain for ubiquitous access via a smart contract. Access rights, however, must be granted only to authorized providers for viewing the data. As discussed in Section 4.4.3, OAuth is a popular platform for communicating permissions in web-based apps that are not based on blockchain.

Problem Smart contracts cannot directly use external services like OAuth since they do not produce deterministic outcomes that can be verified by blockchain miners. A key problem is thus how to design a mechanism that balances the need of permission authorization for clinical data and blockchain requirements for deterministic outcomes.

Design choice → token-based permission model To overcome the limitation with public blockchains, FHIRChain protects the shared content via a secure cryptographic mechanism called "sign then encrypt" [184]. This design employs the users' digital health identities to encrypt content so that only users holding the correct digital identity private keys can decrypt the content. FHIRChain also generates a new pair of signing keys for each participant and registers the public portion of signing keys alongside users' digital identities.

To concretely demonstrate this workflow, Figure 4.3 provides an example of using FHIRChain to create and retrieve an access token. Suppose provider *Alice* would like to initiate sharing of her patient's data, denoted as D_{Alice} (with a reference pointer, denoted as RP_{Alice}) with another provider *Bob*. FHIRChain creates a digital signature on the shared content RP_{Alice}, with *Alice*'s private signing key SKS_{Alice} for tamper-proofing as a first step. With *Bob*'s public encryption key, PK_{Bob}, FHIRChain encrypts the signed RPS_{Alice} to obtain an encrypted token $EncRPS_{Alice}$, and then stores $EncRPS_{Alice}$ in a smart contract for ubiquitous access.

When *Bob* wants to obtain the content *Alice* sent, he must use his corresponding pri-


Provider Alice's Secret Signing Key SKS_Alice
Alice's Data Reference Pointer RP_Alice
Sign
RPS_Alice = h(SKS_Alice,, RP_Alice)
Provider Bob's Public Encryption Key PK_Bob
Encrypt
EncRPS_Alice = enc(PK_Bob,, RPS_Alice)
EncRPS_Alice = enc(PK_Bob,, RPS_Alice)
Provider Bob's Private Encryption Key SK_Bob
Decrypt
RPS_Alice = dec(EncRPS_Alice, SK_Bob)
Provider Alice's Public Signing Key SKS_Alice
Verify
No
Error
Yes
Alice's Data Reference Pointer RP_Alice

(a) Creating encrypted assess token

(b) Decrypting to obtain original token

Figure 4.3: Example of the Creation and Retrieval of an Access Token Using FHIRChain.

vate encryption key SK_{Bob} to decipher the real content of $EncRPS_{Alice}$. *Bob* also verifies that this content was indeed provided by *Alice* with her public signing key PKS_{Alice}. This authentication process is automated by the DApp server component interfacing the smart contract, as discussed in Section 4.5.2.5.

Digital signing ensures that a resource is indeed shared by the sender and is not tampered with. Likewise, encryption protects the information against unauthorized access and spoofing. The data requestor's access to a resource can be approved or revoked at any time via a state update in the smart contract by the data holder where all permissions are logged.

Role-based or attribute-based permissions can also be implemented off-chain in the same manner as in a traditional centralized system (*e.g.*, via Active Directory). In this case, a meta-cryptographic key pair would be created for each role or type of attribute and securely stored within the systems database. The system can then be configured so that only allows users meeting certain permission criteria to use the key for data access, while shielding users from unessential details.

4.5.2.4 Addressing Requirement 4: Consistent Data Formats

Context Clinical research data can exist in various formats and structures, which may or may not be meaningful when shared with other providers from different organizations.

Problem Blockchain-based health IT systems should facilitate data sharing, while adhering to some existing standard(s) for representing the clinical data. A key problem is thus how to design a blockchain-based architecture to enforce the application of existing clinical data standard(s).

Design choice → enforcing FHIR standards FHIR, a proposed interoperability standard developed by HL7, is based on modern web services (*i.e.*, HTTP-based RESTful protocol) and supports the use of JSON [185], which is a popular format for exchanging information on the web. JSON is more compact and readable compared to the XML format used by other data formatting standards, thereby enabling more efficient transmission of JSON-encoded data. It is also compatible with many software libraries and packages. As more health IT systems upgrade their data exchange protocols to comply to FHIR standards, FHIRChain enforces the use of FHIR to shared clinical data by validating whether the generated reference pointers follow the FHIR API standards [28].

4.5.2.5 Addressing Requirement 5: Maintaining Modularity

Context Health IT system updates and/or upgrades are necessary to adopt more efficient, secure, or prevalent technology as it advances.

Problem If functions in a smart contract have too many dependencies on the rest of a health IT system, then each upgrade to the system must deploy a new contract, which requires restoring data from previous versions to prevent loss. A key problem is thus how to design a modular data sharing system that minimizes the need to create new versions of existing contracts when the system is upgraded. For example, when more user friendly features are needed, a good design should separate those updates from the underlying back-end services so that a change in the user interface does not require modifications of the server or blockchain component.

Design choice → applying the model-view-controller (MVC) pattern The *MVC* pattern [182] separates a system into three components: (1) the *model*, which manages the behavior and data of a system and responds to requests for information about its state and instructions to change state, (2) the *view*, which manages the display of information, and (3) the *controller*, which interprets user inputs into appropriate messages to pass onto the *view* or *model*.

The FHIRChain architecture applies the *MVC* pattern to separate concerns with individually testable modules as follows: (1) a model in the form of an immutable *blockchain component* is used to store necessary meta data via smart contracts; (2) a view provides a front-end *user interface* that accepts user inputs and presents data; (3) a controller is a *server* component with control logic that facilitates interactions with data between the *user interface* and *blockchain component*, such as queries, updates, encrypting and decrypting contents; and (4) a controller-invoked *data connector* service is used to validate the implementation of FHIR standards and create reference pointers for the data sources upon requests from the server.

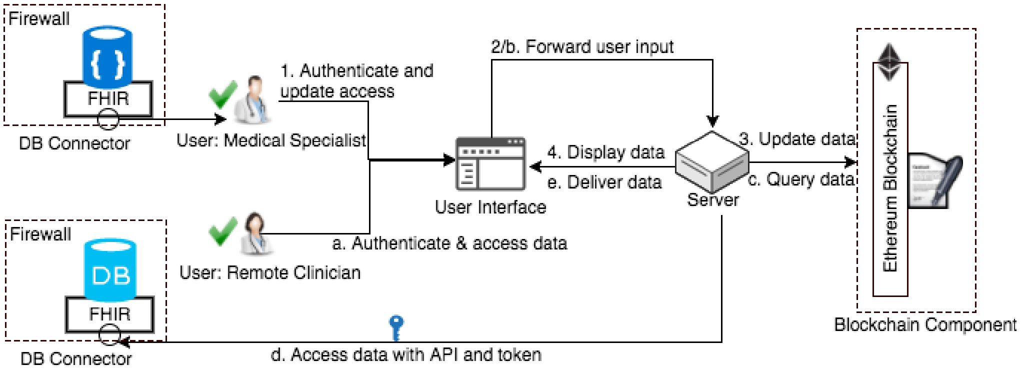


Figure 4.4: Composition and Structure of the FHIRChain Architecture with Modular Components.

The workflow for updating data access is shown in Figure 4.4 by the following steps 1-4:

1. A user first authenticates through the user interface (UI), and when successfully authenticated, data access permission request can be input to the system;

2. The UI forwards user's request to the server;

3. The server logs permissioned or revoked access in the blockchain component (BC); and

4. The server updates UI with proper response to notify the user.

Likewise, the workflow for accessing a data source is outlined in the following steps a-e:

a) The user first authenticates via the UI, and when successfully authenticated data access request can be input to the system;

b) UI forwards users request to the server;

c) The server queries BC for current user's access token(s);

d) When permission is valid, the server decodes the access token(s) with correct keys supplied by user and uses the decrypted reference pointer to obtain actual data from the DB connector to the proper database;

e) When data has been retrieved from the data source via DB connector, the server updates UI to display data in a readable format.

FHIRChain stores all relevant information in smart contracts, decoupling data store from the rest of the system. This decoupling enables future upgrades to all other components without losing access to—or locking out—existing users or their permission information.

4.6 Case Study: Applying FHIRChain to Create a Prototype DApp

This section first describes the structure and functionality of a *decentralized app* (DApp) that customizes the FHIRChain architecture described in Section 4.5 to support collaborative clinical decision making via a remote tumor board case study. We then analyze the benefits and limitations of our DApp case study.

4.6.1 Overview of the FHIRChain DApp Case Study

The FHIRChain DApp is written in Javascript. It consists of $\sim$1,000 lines of core app code that interacts with a private testnet of the Ethereum blockchain and three Solidity smart contracts, each containing $\sim$50 lines of code. Our DApp customizes the FHIRChain architecture in a private Ethereum testnet to address the various ONC requirements described in Section 4.4.

This DApp has an intuitive user interfacing portal that facilitates the sharing and viewing of patient cancer data for a remote tumor board to collaboratively create treatment plan for cancer patients. In addition, the DApp implements a notification service [106] that broadcasts events to appropriate event subscribers. The FHIRChain DApp notification service is used to alert collaborative tumor board members when new data access is available for review.

Verifying identity and authenticating participants with digital identities, as discussed in Section 4.5.2.1. Our DApp contains a *Registry* smart contract that maintains the digital health identities of providers who registered with our app. The registry maps provider email addresses (or phone numbers) from a public provider directory to both their public encryption (used as digital identity) and signing keys, which are generated automatically at user registration time. Figure 4.5 demonstrates the user registration and authentication workflow.

Storing and exchanging data securely with FHIR-based reference pointers, as dis-

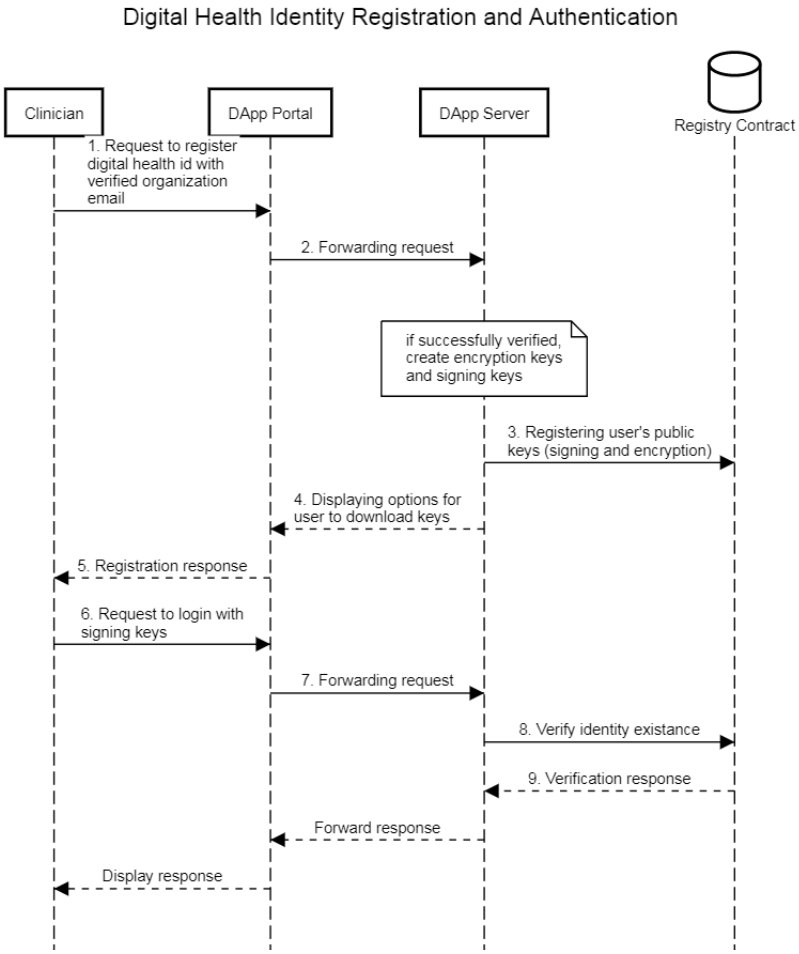

Digital Health Identity Registration and Authentication
Clinician
DApp Portal
DApp Server
Registry Contract
1. Request to register digital health id with verified organization email
2. Forwarding request
if successfully verified, create encryption keys and signing keys
3. Registering user's public keys (signing and encryption)
4. Displaying options for user to download keys
5. Registration response
6. Request to login with signing keys
7. Forwarding request
8. Verify identity existance
9. Verification response
Forward response
Display response

Figure 4.5: Workflow of the User Registration and Authentication Process in the FHIR-Chain DApp.

cussed in Sections 4.5.2.2 and 4.5.2.4. Our DApp defines two cancer patient databases and referencing paths to patient data entries using the open-source HapiFHIR [186] public test server. Validation of the FHIR implementation is performed via regular expression parsing of the paths against the FHIR APIs [28].

Permissioning data access with token-based exchange, as discussed in Section 4.5.2.3. Our DApp also contains an *Access* smart contract that logs all user interactions and requests on the portal, *e.g.*, what resource is shared or no longer shared with which provider by whom and when. These access logs are structured as a mapping between user digital health identities (public encryption keys) and authorizations to custom-named access tokens (rep-

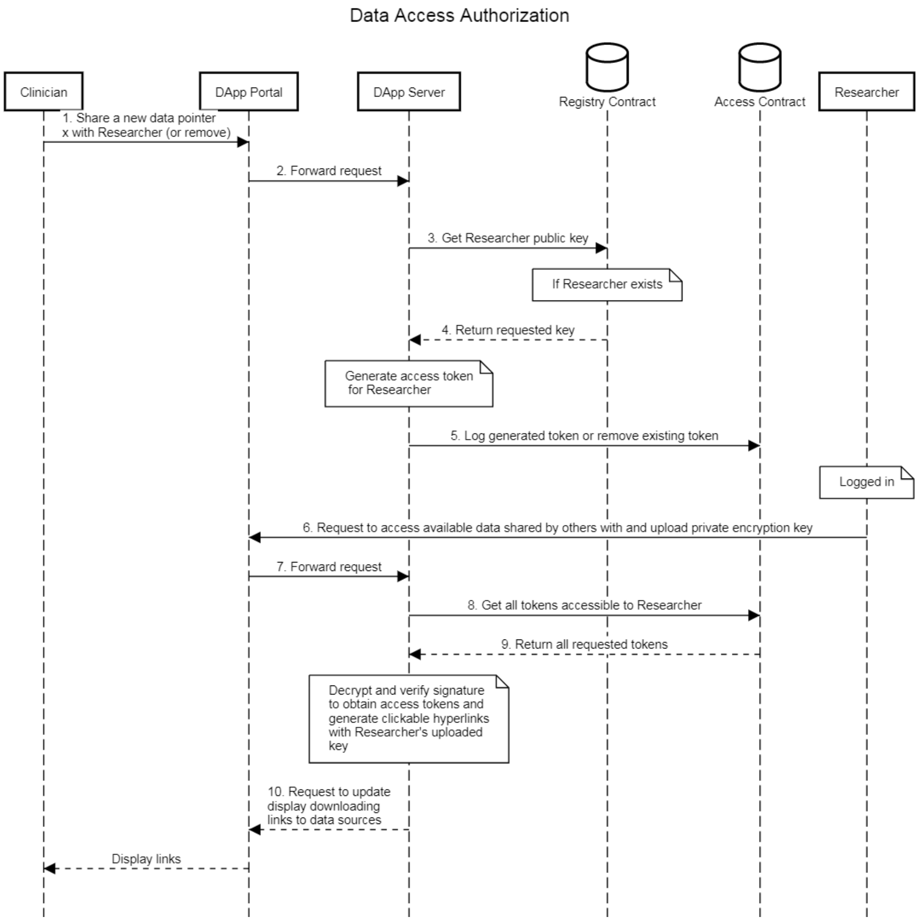
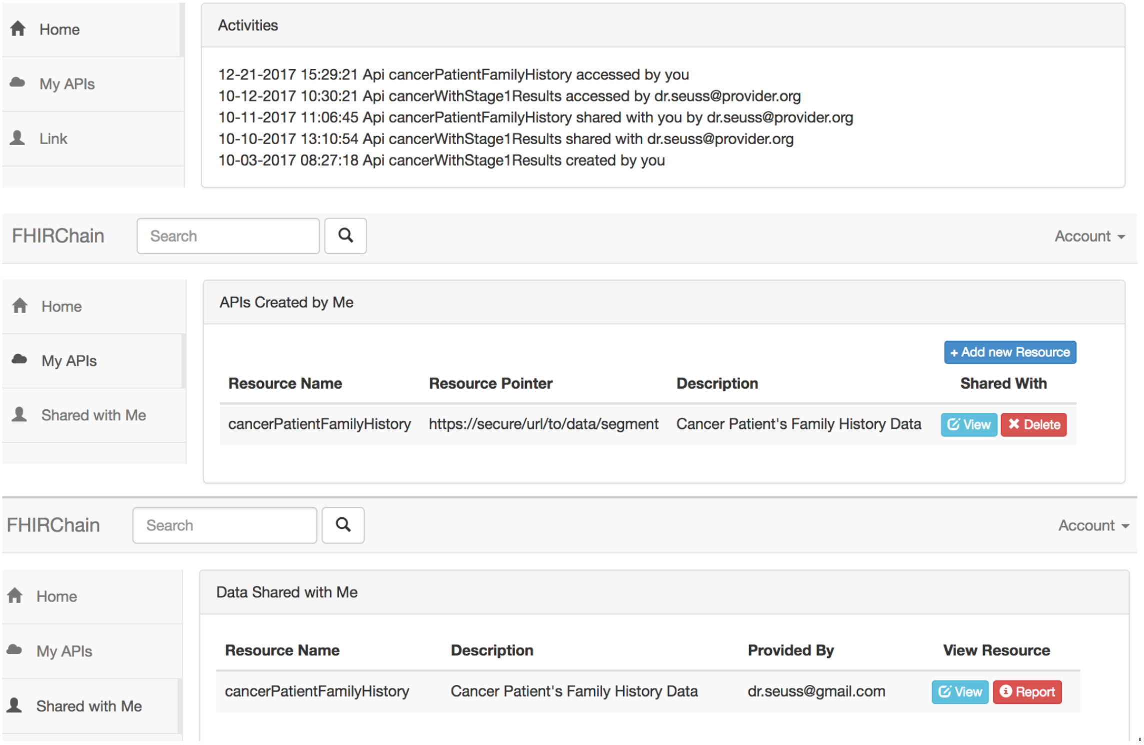

Data Access Authorization
Clinician
DApp Portal
DApp Server
Registry Contract
Access Contract
Researcher
1. Share a new data pointer x with Researcher (or remove)
2. Forward request
3. Get Researcher public key
If Researcher exists
4. Return requested key
Generate access token for Researcher
5. Log generated token or remove existing token
Logged in
6. Request to access available data shared by others with and upload private encryption key
7. Forward request
8. Get all tokens accessible to Researcher
9. Return all requested tokens
Decrypt and verify signature to obtain access tokens and generate clickable hyperlinks with Researcher's uploaded key
10. Request to update display downloading links to data sources
Display links

Figure 4.6: Workflow of Access Authorization in the FHIRChain DApp.

resented as a nested object associated with a *true/false* boolean value indicating if an access token access is granted for a provider). If an access revocation occurs, authorization is set to *false* and the associated token is set to an empty value. The workflow of this process is shown in Figure 4.6.

Maintaining modularity with the MVC pattern, as discussed in Section 4.5.2.5. The *view* component is a user interfacing portal that accepts provider user input, including registration and authentication credentials (corresponding keys) and data access information (*e.g.*, tumor board member email to query, a reference pointer to securely access data, and approval/revocation of access). Figure 4.7 is a screenshot of our DApp, presenting the following features (1) display recent sharing events related to the user, (2) display reference pointer APIs created by logged in user and available actions, and (3) display all references



```
12-21-2017 15:29:21 Api cancerPatientFamilyHistory accessed by you
10-12-2017 10:30:21 Api cancerWithStage1Results accessed by dr.seuss@provider.org
10-11-2017 11:06:45 Api cancerPatientFamilyHistory shared with you by dr.seuss@provider.org
10-10-2017 13:10:54 Api cancerWithStage1Results shared with dr.seuss@provider.org
10-03-2017 08:27:18 Api cancerWithStage1Results created by you
```

Resource Name	Resource Pointer	Description	Shared With
cancerPatientFamilyHistory	https://secure/url/to/data/segment	Cancer Patient's Family History Data	View Delete

Resource Name	Description	Provided By	View Resource
cancerPatientFamilyHistory	Cancer Patient's Family History Data	dr.seuss@gmail.com	View Report

Figure 4.7: Screenshot of Our FHIRChain-based DApp User Interface.

shared with logged in user and the option to view data.

The portal then forwards the user requests along with data input to the *sever* component, where all the complex logic is encapsulated.

Our FHIRChain DApp *server* performs all functions and control logic, including verifying provider user email account, generating cryptographic keys, token creation via signing and encryption, token retrieval via decryption and signature verification, forwarding requests and delegating tasks between the *portal* and *blockchain*. The *blockchain* component is an independent *model* component containing two smart contracts for ubiquitous storing and persisting event logs of data access.

4.6.2 Benefits of Our FHIRChain DApp Case Study

Our FHIRChain Dapp case study achieved the following benefits:

- **Increased modularity**. To increase modularity, we applied the "separation of concerns" principle [187] to decompose our DApp into independent components. FHIR-Chain employs a peer-to-peer API exchange protocol that references data pointers stored in a smart contract on the blockchain. In this design, exchanged information becomes lightweight, which increases scalability since system performance remains the same regardless of the original size of the data. Likewise, data is not transmitted electronically across institutional boundaries, thereby reducing the risk of data being compromised.

- **Scalable data integrity**. To ensure scalable data integrity, our design maintains a hash of the original data to exchange in addition to the reference pointer of the data. Suppose that the original data being exchanged is of size N and that the size of its reference pointer is ε. The total amount of data stored on-chain in terms of space complexity is then $O(hash(N)+\varepsilon)$. Since the hashed output of a variable-length input can be a fixed value, it consumes a constant amount of space. The size of a data reference pointer would be scalably smaller than the actual data size. This design therefore enhances scalability by using constant-sized representations of the data, rather than using the actual data.

- **Fine-grained access control**. To enable fine-grained access control, permissions to access a data source can be given or revoked at will by providers across various institutions regardless of their trust relationships. By implementing the FHIR standards, more granular access can be granted to selected pieces of data rather than an entire document, which also increases data readability. Moreover, all events related to data sharing and data access are logged in a transparent history for auditability.

- **Enhanced trust**. The DApp applies public key cryptography, which enhances trust to participants in the following ways:

  - **Identifiability and authentication**. Given the computation power today, it is infeasible to impersonate a user without knowing their private key, and the only way a user can be authenticated to use our service is to provide the correct private key paired with their public key registered on the blockchain. On the other hand, it is trivial to create a new public/private key pair in case of a user's private key being lost or stolen. This "digital identity" approach has been successfully adopted in Estonias government and healthcare infrastructure [188].

  - **Permission authorization**. With public key encryption securing their data reference pointers, users can trust that none other than the intended data recipient can view what they have shared. FHIRChain never shares the reference pointer with any user. Instead, RP is used to display the data content when it is decrypted with an authorized user's private key. In addition, users can approve or revoke data access at any time, and the request takes effect immediately.

4.6.3 Limitations of Our FHIRChain DApp Case Study

Since our FHIRChain DApp was designed based on several assumptions it incurs the following limitations:

- **Does not address semantic interoperability**. FHIRChain cannot address data exchange challenges related to semantic interoperability that are not yet fully captured by the FHIR standards. To provide semantics to clinical data, therefore, manual inspection and mapping of predefined ontologies from medical and health data experts are required, which remain the focus of our future research in this space.

- **May not be compatible with legacy systems not supporting FHIR.** Many legacy systems may use other messaging standards, such as the more prevalent HL7 v2 standards [189], and do not support FHIR protocols. The goal of this chapter, however, is to present the underlying representations and theories of our blockchain-based system. Although we advocate FHIR in this research because it has been used quite frequently and it supports fine-grained data exchange, the principles behind the system described here can also be used with other standards like HL7 v2 [189].

- **Cannot control clinical malpractice**. The intended users of FHIRChain are clinicians interested in collaboratively providing clinical decision support for remote patients. Our current design trusts that the data being exchanged using our DApp is not abused, misused, or unethically redistributed by users. Our future work will explore options to minimize these risks, such as tracking data credibility using cryptographic hashing or zero knowledge proofs [190] (ability to demonstrate the truth of a statement without revealing additional information beyond what it is trying to prove [191]) along with each reference pointer. Naturally, clinical malpractice may still occur (as in any other health IT system) since we cannot fully control these human behaviors.

- **DApp deployment costs**. Unlike existing public blockchain, such as Ethereum, our DApp is developed using a private testnet that imposes no interaction costs (*e.g.*, transaction fees). Our DApp would thus not be free of charge if deployed on a public blockchain. The convenience provided by a public blockchain, however, may justify the cost of usage versus the costs of licensing, running, and maintaining a private clinical data exchange infrastructure.

To overcome these limitations in future work, we will deploy our DApp in a permissioned consortium blockchain platform with trusted parties to ensure consensus through a variation of proof-of-work that incentivizes mining with cryptocurrency rewards. For instance, [161] proposes to use aggregated data as mining rewards in their system, while MultiChain [192] enforces a round-robin mining protocol in their blockchain. With the ability to replace monetary incentives to maintain consensus on the blockchain, the cost to use this blockchain-based service will be lower in the long run, although the initial deployment may still be expensive.

Although permissioned systems may be prone to collusion due to the 51% attack problem [24], the permissioned system used for healthcare would be maintained and managed by relatively large-scale entities/stakeholders within the healthcare industry. Unless majority of them (major hospitals, insurance companies, *etc.*) collude, therefore, the chance of experiencing this type of attack is quite low. Moreover, legal actions would most likely occur immediately upon the attack.

4.7 Conclusion

This chapter described the FHIRChain prototype we designed to provide patients with more collaborative clinical decision support using blockchain technology and the FHIR data standards. Complemented by the adoption of public key cryptography, our FHIRChain design addressed five key requirements provided by the ONC interoperability roadmap, including user identifiability and authentication, secure data exchange, permissioned data access, consistent data formats, and system modularity.

The following are the key lessons we learned from designing and implementing our DApp based on FHIRChain:

- **FHIRChain can provide trustless, decentralized storage for necessary meta information and audit logs**. FHIRChain alleviates proprietary vendor-lock found in conventional health IT systems by leveraging its blockchain component as a decentralized storage of necessary reference information as secure access points into those databases. It enables the sharing of clinical data without established trusts, providing clinicians with secure and scalable collaborative care decision support. In addition, each public key generated for a user is stored in the blockchain via a smart contract used to associate healthcare participants with their digital identities. Similarly, permission authorizations established between those participants are recorded in a smart contract as well, creating a traceable permission database with an audit log of data exchange history (*i.e.*, meta information involved during the data exchange and not the actual data). Storing these data on the blockchain ensures that our app is not subject to a single point of failure or corruption of records so that it is always accessible by healthcare participants.

- **FHIRChain facilitates data exchange without the need to upload/download data thus maintains data ownership.** The FHIR standards provide resource APIs to reference specific pieces of structured data while maintaining original data ownership. By adopting FHIR and combining it with blockchain technologies, FHIRChain creates lightweight reference pointers to siloed databases and exchange these pointers via the blockchain component instead of actual data. For telemedicine clinics or clinics in rural areas in particular, this approach can overcome network limitations by enabling scalable data sharing without requiring data to be uploaded to some other centralized repository, through which data can be shared and downloaded by other parties. In addition, this approach reduces risks of compromised data and ensures that original data ownership is respected. The reference pointers are encrypted with the intended recipients public key, *i.e.*, digital identity to permission data access. When successfully authenticated (*i.e.*, reference pointers are correctly decrypted) the data will be downloaded directly from the source and present properly formatted data to the user.

- **Public key cryptography can be effective for managing digital health identity in data sharing**. FHIRChain creates public keys as digital health identities associated with each collaborating care entity (provider or organization administrator). The benefits to this strategy include: (1) *easy authentication* since a clinician only needs to provide their private key associated with their identity, (2) *integrity* since by signing the exchanged reference pointers FHIRChain can easily verify that it was provided by the signed provider and has not been modified, and (3) *remedy to lost or stolen keys* since a new key can be created easily to replace the old key and associate with the same user. There is a drawback, however, to using digital identities for patients in a general clinical setting. Managing these identities—private keys—is hard because private keys are harder to remember than conventional passwords and require technical training for patients to manage their own keys. Nevertheless, there are approaches for managing private keys for larger populations, such as using key wallets [193, 23] or embedding private keys to physical medical ID cards [194].

In summary, our FHIRChain-based DApp demonstrates the potential of blockchain to foster effective healthcare data sharing while maintaining the security of original data sources. FHIRChain can be further extended to address other healthcare interoperability issues, such as coordinating other stakeholders (*e.g.*, insurance companies) across the industry and providing patients with easier (and secure) access to their own medical records.

Chapter 5

A PATTERN LANGUAGE FOR DESIGNING BLOCKCHAIN-BASED HEALTH IT SYSTEMS

The decentralization, transparency, and immutability properties make blockchains suitable for apps that require disintermediation through trustless exchange, consistent and incorruptible transaction records, and operational models beyond cryptocurrency. In particular, there has been growing interest in using blockchain and its programmable "smart contracts" to address healthcare interoperability challenges, such as enabling effective interactions between users and medical applications, delivering patient data securely across healthcare facilities, and improving the overall efficiency of medical communications. Despite the interest in applying blockchain technology to healthcare, little information is available on the concrete architectural styles and recommendations for designing blockchain-based apps with a healthcare focus.

The key contribution of this chapter is the introduction of a pattern sequence, using both traditional software patterns and novel patterns targeting blockchain-based apps, applicable in the design of blockchain-based healthcare systems focused on clinical communication and health data exchange. The application of this pattern sequence takes into account both the technical requirements specific to healthcare systems and the implications of these requirements on naive blockchain-based solutions. It provides a pattern-oriented reference architecture for software developers to create an interoperable (on the technical level) health IT system atop any blockchain infrastructure while minimizing storage requirements on the blockchain, preserving the privacy of sensitive information, facilitating scalable communication and maximizing evolvability of the system.

5.1 Problem Overview

Over the past several years, blockchain technology has attracted interest from computer scientists and domain experts in various industries, including finance, real estate, healthcare, and transactive energy. This interest initially stemmed from the popularity of Bitcoin [23], which is a cryptographic currency framework that was the first application of blockchain. Blockchain possesses certain properties, such as decentralization, transparency, and immutability, that have allowed Bitcoin to become a viable platform for ”trustless” transactions [195], which can occur directly between any parties without the intervention of a trusted intermediary.

Another blockchain platform, Ethereum, extended the capabilities of the Bitcoin blockchain by adding support for ”smart contracts” [24]. Smart contracts are code that directly controls the exchanges or redistributions of digital assets between two or more parties according to certain rules or agreements established between involved parties. Ethereum’s programmable “smart contracts” enable the development of decentralized apps (DApps) [163], which are autonomously operated services with data and records of operations cryptographically stored on the blockchain. DApps also enable direct interactions between end users and data on the blockchain.

Blockchain and its programmable smart contracts are being explored as a potential solution to address healthcare interoperability issues [27, 159]. Interoperability is defined as the ability for different information technology systems and software apps to communicate, exchange data, and effectively use the exchanged information [196]. In the healthcare domain, it is necessary to achieve both syntactic and semantic interoperability, such as enabling effective interactions between users and medical applications, delivering patient data securely across healthcare facilities, and improving the overall efficiency of medical communications [197]. Despite the interest in using blockchain technology for healthcare, however, little information is available on the concrete approaches for designing blockchain-based apps targeting healthcare-specific challenges.

This chapter focuses on addressing this unexplored research topic: *providing software patterns for designing blockchain-based healthcare DApps to help mitigate healthcare challenges*. The target audience of this chapter is thus health IT system developers interested in applying blockchain technologies in the design. Design patterns provide general solutions without tying specifics to a particular problem, allowing developers to communicate using well-known and well-understood names for software interactions [31]. By identifying these recurring patterns applicable to healthcare, this work intends to help the target audience more quickly adapt to this technology and create robust healthcare solutions with it.

5.2 Key Concepts of Blockchain and Its Role in Healthcare Apps

This section gives a general overview of blockchain and the open-source Ethereum implementation that provides additional support for smart contracts, which are computer protocols that enable different types of decentralized apps beyond cryptocurrencies. The general blockchain overview is followed by a discussion of Solidity, which is a programming language used to write Ethereum smart contracts.

5.2.1 Key Concepts of Blockchain

A blockchain is a decentralized computing architecture that maintains a growing list of ordered transactions grouped into blocks that are continually reconciled to keep information up-to-date, as shown in Figure 5.1. All transaction records are kept in the blockchain

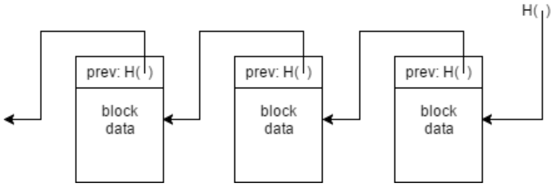


Figure 5.1: Blockchain Structure: a Continuously Growing List of Ordered and Validated Transactions

and are shared with all network nodes. Only one block can be added to the blockchain at a time. Each block is mathematically verified (using cryptography) to ensure that it follows in sequence from the previous block. The verification process is called ”mining” or *Proof of Work* [23], which allows network nodes (also called ”miners”) to compete to have their block be the next one added to the blockchain by solving a computationally expensive puzzle. The winner then announces the solution to the entire network to gain a mining reward paid via cryptocurrency. The mining process combines cryptography, game theory, and incentive engineering to ensure that the network reaches consensus regarding each block in the blockchain and that no tampering occurs in the transaction history. This process ensures properties of transparency, immutability, and decentralization (resilient to a single point of failure due to replicated storage) [198].

In the Bitcoin application, blockchain serves as a public ledger for all cryptocurrency transactions in bitcoins to promote trustless financial exchanges between individual users, securing all their interactions with cryptography. The Bitcoin blockchain has limitations, however, when supporting different types of applications involving contracts, equity, or other information, such as crowdfunding, identity management, and democratic voting registry [24]. To address the needs for a more flexible framework, Ethereum was created as an alternative blockchain, giving users a generalized trustless platform that can run smart contracts.

The Ethereum blockchain is a distributed state transition system, where state consists of accounts and state transitions are direct transfers of value and information between accounts. Two types of accounts exist in Ethereum: (1) *externally owned accounts* (EOAs), which are controlled via private keys and only store Ethereum’s native value-token ”ether” and (2) *smart contract accounts* (SCAs) that are associated with contract code and can be triggered by transactions or function calls from other contracts [24].

To protect the blockchain from malicious attacks and abuse (such as distributed denial of service attacks in the network or hostile infinite loops in smart contract code), Ethereum also enforces a payment protocol, whereby a fee (in terms of “gas” in the Ethereum lexicon) is charged for data storage and each data operation executed in a contract. These fees are collected by miners who verify, execute, propagate transactions, and then group transactions into blocks.

As in the Bitcoin network, the mining rewards provide an economic incentive for users to devote powerful hardware and electricity to the public Ethereum network. In addition, transactions have a gas limit field to specify the maximum amount of gas that the sender is willing to pay for. If gas used during transaction execution exceeds this limit, computation is stopped, but the sender still has to pay for the performed computation. This protocol also protects senders from completely running out of funds.

5.2.2 Overview of Solidity

Ethereum smart contracts can be built in a Turing-complete programming language, called Solidity [199]. This contract language is compiled by the Ethereum Virtual Machine (EVM), which enables the Ethereum blockchain to become a platform for creating DApps that provide potential solutions to certain healthcare interoperability challenges. Solidity has an object-oriented flavor and is intended primarily for writing contracts in Ethereum.

A ”class” in Solidity is realized through a ”contract,” which is a prototype of an object that lives on the blockchain. Just like an object-oriented class can be instantiated into a concrete object at runtime, a contract may be instantiated into a concrete SCA by a transaction or a function call from another contract. At instantiation, a contract is given a uniquely identifying address, similar to a reference or pointer in C/C++-like languages, with which it can then be called.

Contracts may contain persistent state variables that can be used as data storage and functions that interact with the states. Although one contract can be instantiated into many SCAs, it should be treated as a singleton to avoid storage overhead. After a contract is created, its associated SCA address is typically stored at some place (*e.g.*, a configuration file or a database) and used as a parameter by an app to access its internal states and functions [200].

Solidity supports multiple inheritance and polymorphism [201]. When a contract inherits from one or more other contracts, a single contract is created by copying all the base contracts code into the created contract instance. Abstract contracts in Solidity allow function declaration headers without concrete implementations. They cannot be compiled into an SCA but can be used as base contracts. Due to Solidity contracts' similarity to C++/Java classes, certain software patterns can be directly applied to smart contracts as well, as we describe in Section 4.

5.3 Related Work

5.3.1 Overview

Although relatively few papers focus on realizing software patterns in blockchains, some relate to healthcare blockchain solutions and design principles in this space. This section gives an overview of related research on (1) the challenges of applying blockchain-based technology in the healthcare space and innovative implementations of blockchain-based healthcare systems and (2) design principles and recommended practice for blockchain application implementations.

5.3.1.1 Challenges of healthcare blockchain and proposed solutions.

Azaria et al. [161] proposed MedRec as an innovative, working healthcare blockchain implementation for handling EHRs, based on principles of existing blockchains and Ethereum smart contracts. The MedRec system uses database ”Gatekeepers” for accessing a node's local database governed by permissions stored on the MedRec blockchain. Peterson et al. [168] presented a healthcare blockchain with a single centralized source of trust for sharing patient data, introducing *Proof of Interoperability* based on conformance to the

FHIR protocol as a means to ensure network consensus.

5.3.1.2 Prior efforts focused on software design practice for developing blockchain apps.

Porru et al. [202] highlighted evident challenges in state-of-the-art blockchain-oriented software development by analyzing open-source software repositories and addressed future directions for developing blockchain-based software. Their work focused on macro-level design principles such as improving collaboration, integrating effective testing, and evaluations of adopting the most appropriate software architecture. Bartoletti et al. [203] surveyed the usage of smart contracts and identified nine common software patterns shared by the studied contracts, *e.g.*, using ”oracles” to interface between contracts and external services and creating ”polls” to vote on some question. These patterns summarize the most frequent solutions to handle some repeated scenarios. More recently, a number of attacks on Ethereum smart contracts have been reported, including the infamous DAO attack [25] where $50 million worth of Ether was stolen and the critical Parity wallet hack [204] that incurred in $30 million worth of Ether being exploited. Atzei et al. surveyed existing attacks on Solidity smart contracts with code snippets showing related vulnerabilities [205]. Meanwhile, the blockchain community also compiled a number of software patterns and anti-patterns targeting Solidity programming around cryptocurrency transactions in order to maximize the security of Ethereum smart contract design [206].

5.3.2 Gaps in Existing Research

Many research and engineering ideas have been proposed to apply blockchain technology in healthcare, and implementation attempts are underway [161, 168, 202, 203]. As discussed in Section 5.3.1, prior research efforts have provided a number of design recommendations for implementing Solidity smart contracts involving cryptocurrency transactions. Few published studies, however, have addressed software design considerations needed to implement blockchain-based healthcare apps effectively. While it is crucial to understand the fundamental properties of blockchains and the smart contract programming language, it is also important to apply them properly so that healthcare-specific challenges are addressed. Even though a subset of principles from prior work may be relevant to the healthcare space, a systematic approach to document appropriate design practice that specifically target technical challenges in healthcare is still essential.

5.4 Healthcare Interoperability Challenges Faced by Blockchain-Based Apps

The US Office of the *National Coordination for Health Information Technology* (ONC) has outlined basic technical requirements for achieving interoperability [207]. Based on these requirements, this section summarizes key interoperability challenges faced by blockchain-based apps, focusing on four aspects: system evolvability, blockchain storage, information privacy, and scalability.

5.4.1 Evolvability Challenge: Maintaining Evolvability While Minimizing Integration Complexity

Many traditionally centralized apps are written with the assumption that data is easy to change. This assumption does not hold true for blockchain-based apps. Once stored on-chain, data is difficult to modify *en masse*. Not only is code manipulating the data is immutable, but data change history also persists on-chain and can be replayed due to the nature of blockchain. Healthcare data contains sensitive personal information protected by law [173], which, if compromised, would create severe legal, financial, and also social consequences. The vulnerability in smart contract code leading to the infamous DAO attack [25] must be avoided in a healthcare app.

At the same time, healthcare systems may be subject to updates or upgrades required by clinical workflow or healthcare regulations. This need for potential system evolution creates a tension for a blockchain-based design. As such, a critical design consideration when building blockchain apps for healthcare is to ensure that the data written into blockchain via smart contracts are designed to facilitate evolution where mandated.

Although evolution must be supported, healthcare data must often be accessible from a variety of deployed systems that cannot easily be changed over time. Apps should therefore be designed in a way that is loosely coupled and minimizes the usability impact of evolution on the clients, *i.e.*, user services that interact with data in the blockchain. Sections 5.5.1, 5.5.3, and 5.5.2 shows how using LAYERED RING, CONTRACT MANAGER, and GUARDED UPDATE patterns from the pattern sequence, respectively, can help avoid serious attacks like the DAO [25] and facilitate necessary system evolution, while minimizing the impact on dependent clients, focusing on the separation of concerns between data and logic and a type of attack called reentrancy, which will be described further in Section 5.5.2.

5.4.2 On-Chain Storage Challenge: Minimizing Data Storage Requirements on the Blockchain

Healthcare apps can serve thousands to millions of participants, which may incur enormous overhead when *large* volumes of data are stored in a blockchain–particularly if data normalization and denormalization techniques are not carefully considered. Considering storage scalability, not only is it costly to store data, such as the HFQ data we discussed earlier in Section 1.1, but data modifications and access operations may also fail if/when the cost of storage or execution exceeds the allowance in a blockchain, *e.g.*, gas limit defined for the Ethereum blockchain as discussed in Section 5.2.1. An important design consideration for blockchain-based healthcare apps is thus to minimize data storage requirements in addition to provide sufficient flexibility to manage individual health concerns. Sections 5.5.4 and 5.5.6 show how to design smart contracts with DATABASE CONNECTOR and ENTITY REGISTRY patterns from the pattern sequence, respectively, to improve interoperability by standardizing interfaces to storage access and maximizes on-chain scalability by capturing common intrinsic data sharing across entities while still allowing extrinsic data to vary in specific entity contracts.

5.4.3 On-Chain Privacy Challenge: Balancing Data Storage with Privacy Concerns

Blockchains and smart contracts can offer trustless digital health asset sharing, audit trails of data access, and decentralized and replicated storage, which are essential for improving healthcare interoperability by providing ubiquitous data store. Although there are substantial potential benefits to the availability of information if data is stored on-chain, there are also significant risks due to the transparency of blockchain. In particular, even when encryption is applied to sensitive data on-chain, it is still possible that the current encryption techniques may be broken in the future [208] or that vulnerabilities in the encryption implementations may later be exploited, rendering private information potentially decryptable in the future. To protect health information privacy, in Sections 5.5.5 and 5.5.7 we discuss how designing a blockchain-based app using the DATABASE PROXY and TOKENIZED EXCHANGE patterns from the pattern sequence, respectively, can facilitate data sharing while keeping sensitive patient data from being directly encoded in the blockchain.

5.4.4 Scalable Communication Challenge: Tracking Relevant Health Changes Scalably Across Large Patient Populations

Communication gaps and information sharing challenges are serious impediments to healthcare innovation and the quality of patient care. Providers, hospitals, insurance companies, and departments within health organizations experience disconnectedness caused by delayed or lack of information flow. Patients are commonly cared for by various sources, such as private clinics, regional urgent care centers, and enterprise hospitals. A provider may have hundreds or more patients whose associated health data must be tracked. Section 5.5.8 shows how a blockchain-based app design using the PUBLISHER-SUBSCRIBER pattern from our pattern sequence can be aid in scalably detecting and communicating relevant health changes.

5.5 A Key Pattern Sequence for Designing Blockchain-Based Health Apps

This section presents a key pattern sequence for creating blockchain-based health system designs that address the major challenges described earlier in Section 5.4. Our research approach that developed this sequence has three folds. First, because the topic of design patterns focused on using blockchain technology for healthcare has received limited attention in literature, we had extracted a subset of the patterns using commonality and variability analysis [32]. Specifically, we obtained a number of verified smart contract source code from Etherscan.io [33] to capture common portions repeatedly used across various contracts and/or supporting library contracts, which we codified into patterns of this sequence, such as LAYERED RING and CONTRACT MANAGER Second, based on our experience from previous work on researching healthcare data sharing solutions [106, 21] and our understanding of the healthcare domain and the technical requirements for its systems [162, 22], we codified the design practice we learned from prior research into several patterns in the key sequence, such as DATABASE CONNECTOR and TOKENIZED EXCHANGE. Third, given the extensiveness and maturity of existing research on centralized and distributed software engineering design practice, we have applied, wherever necessary for a blockchain-based healthcare system, design principles widely accepted to our pattern sequence with blockchain-focused design considerations, such as DATABASE PROXY, ENTITY REGISTRY, AND PUBLISHER-SUBSCRIBER. Additionally, due to the growing popularity of Solidity (which is the primary programming language for creating smart contracts) and attacks that have occurred to public smart contracts, the Ethereum community has captured a number of Solidity code patterns for preventing similar attacks. Although those code patterns were almost exclusively targeting cryptocurrency or other apps with financial incentives, we identified one code pattern that would be particularly critical in a healthcare system, namely, GUARDED UPDATE.

The remainder of this section applies a pattern form variant to motivate and show how our pattern sequence aids in designing blockchain-based healthcare apps. In particular, we present eight software patterns—LAYERED RING, GUARDED UPDATE, CONTRACT MANAGER, DATABASE CONNECTOR, DATABASE PROXY, ENTITY REGISTRY, TOKENIZED EXCHANGE, and PUBLISHER-SUBSCRIBER [209, 210]. We describe key healthcare challenges that they resolve in the blockchain platform and detail their structure and composition. [1].

Table 5.1 provides an overview of the pattern sequence, showing how the patterns relate to healthcare-specific challenges described previously in Section 5.4 and what specific sub-challenge each pattern aims to solve.

Table 5.1: Overview of Proposed Pattern Sequence for Designing Blockchain-Based Healthcare Apps

Pattern	Targeted Category	Specific Challenge to Solve
Layered Ring	Evolvability	Defining the data sharing systems base architecture
Guarded Update	Evolvability	Preventing unexpected reentrancy attacks that occurred in the DAO
Contract Manager	Evolvability	Separating data from logic to ensure data availability via clean separation of concerns
Database Connector	On-Chain Storage	Ensuring on-chain storage scalability and interoperability via standardized and minimal interfaces to off-chain storage
Database Proxy	On-Chain Privacy	Providing an additional layer of security by performing lightweight tasks before permitting access to database connectors
Entity Registry	On-Chain Storage	Managing healthcare entities on-chain and other types of common data at scale
Tokenized Exchange	On-Chain Privacy	Authorizing access to data storage and maintaining verifiable access logs
Publisher-Subscriber	Scalable Communication	Providing user notifications when events of interest occur across the decentralized network

Each of the patterns in this sequence is discussed in depth below.

[1] Naturally, there are other patterns relevant in this domain, which can be the focus of future work.

5.5.1 A Blockchain-Based Architecture for Health Data Sharing Systems

Design problem faced by blockchain-based apps. Healthcare data, as we saw earlier in Chapter 1 (Section 1.2), exists in siloed data warehouses across different healthcare organizations, private practices, and, more recently, mobile health app providers [211, 21]. Despite the adoption of certified EHRs or other data exchange solutions that can provide direct data exchange between providers within the same network (*e.g.*, using an EHR system provided by the same vendor), impediments for healthcare providers and researchers to access those heterogeneous data silos still exist.

Solution → Apply the LAYERED RING pattern to define the base architecture of the health data sharing system. The emerging blockchain technology that supports decentralized data storage and executable code via smart contracts, with Ethereum [24] being the most popular, has presented itself as a potential infrastructure to connect existing healthcare data silos [165, 212, 213] with its success in maintaining tamper-proof cryptocurrency transactions between worldwide Internet users [23, 214] and managing verifiable collectibles or rewards from cryptogaming like CryptoKitties [215].

At the architectural level, the healthcare data sharing problem is not too different from those successful use cases of blockchain. Figure 5.2 compares the high-level architecture of data sharing in healthcare with that of blockchain-based cryptocurrency exchange and cryptogaming. In this figure, the bottom level in both architectures 1 and 2 contains a number of heterogeneously represented objects, i.e., siloed healthcare data sources in *Architecture 1* and geographically dispersed Internet users in *Architecture 2*. Data sources generated by healthcare professionals via from diverse, centralized EHR systems on the left may or may not inter-operate, depending on if an authorized exchange service is available between the data sources. Whereas on the right, data (like identifiers of users/gamers) and data requests flow into and out of the same service implemented on the blockchain that is decentralized and widely accessible, with or without a user interface.

Consequently, a blockchain-based healthcare data sharing can apply the same basic pattern of *Architecture 2* in Figure 5.2, except that a user interface is needed for normal healthcare users who typically do not have advanced knowledge about how to execute smart contract functions. In fact, most blockchain apps, such as CryptoKitties (a cryptogame for collecting and breeding digial cats) [215], Fomo3D (a gambling game for winning cryptocurrency lotteries) [216], and IDEX (a cryptocurrency trading platform) [217], implement a user-friendly interface that encapsulates the blockchain component, providing users with familiar experience as if interacting with any other centralized web app.

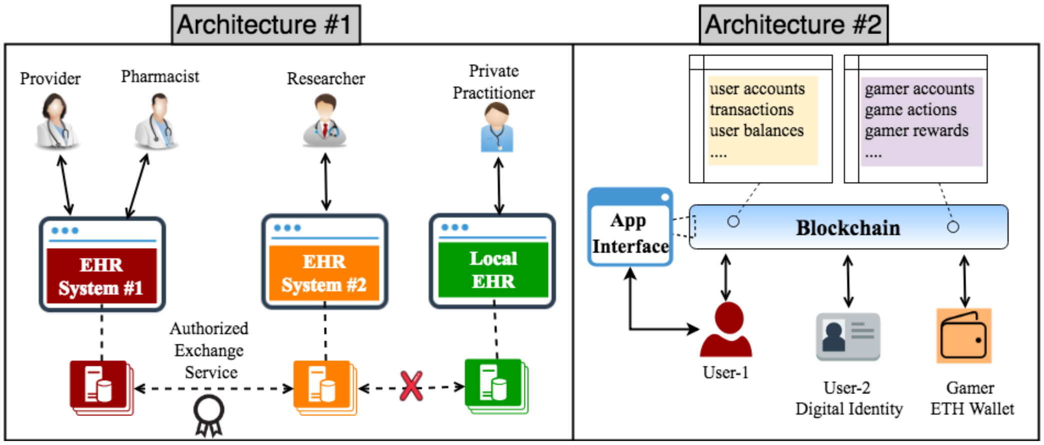


Figure 5.2: Comparing the Current State of Traditionally Centralized Healthcare Architecture with that of Popular Blockchain-Based Use Cases

In Figure 5.3 we present the first pattern in the sequence, LAYERED RING, generalized from *Architecture 2* above with a bird's eye view to better illustrate the scale of involved entities in each layer. The outermost layer is a *Storage Layer*, which contains a large number of data sources, each maintained by its owner (e.g., a private practitioner or a healthcare organization). The middle *Blockchain Layer* connects data sources from the outer layer and would be maintained by key stakeholders or mid- to large-size healthcare organizations in a consortium environment. The innermost *Web App Layer* provides a convenient interface for interacting with data and operations defined in the blockchain. It is also the most centralized piece of the system because a web app is usually hosted in a centralized server. Nevertheless, with a careful design of the web app server, this layer would not introduce additional dependency to the rest of the system and thus maximize the separation of concerns across the overall system.

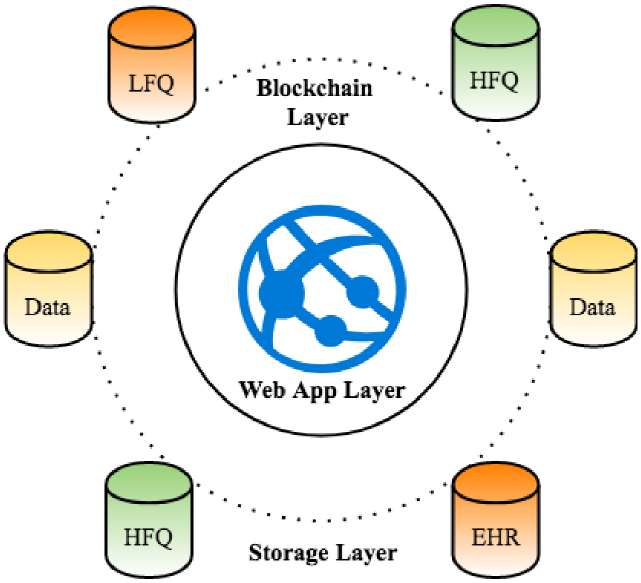


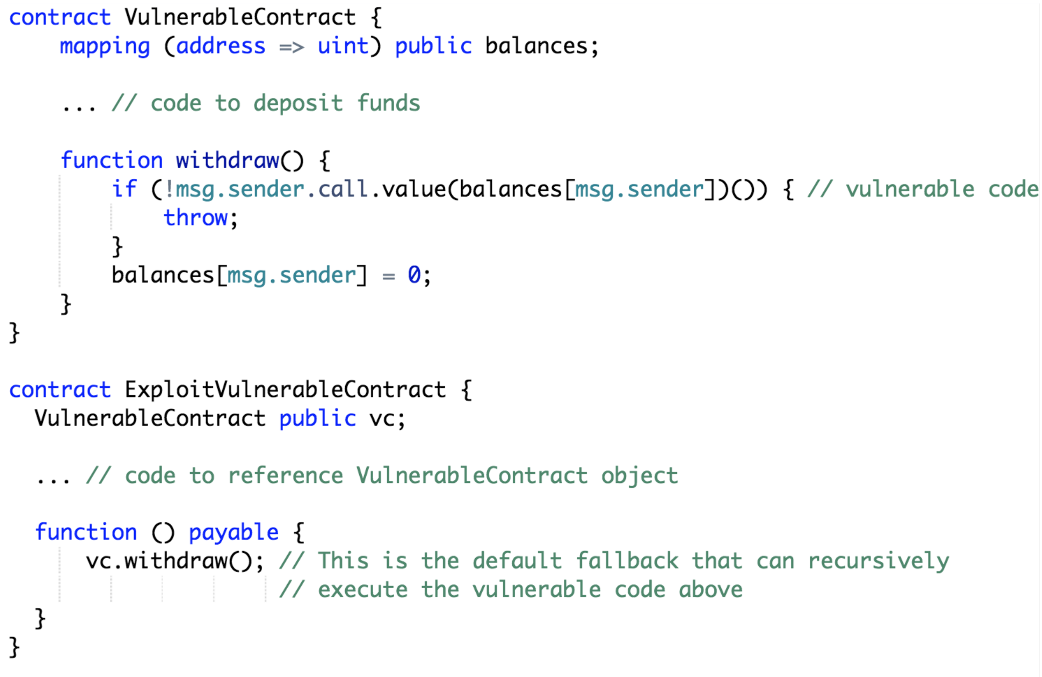
Figure 5.3: Structure of the LAYERED RING Pattern that Defines the Base Architecture of the Data Sharing System

After defining the base-line architecture for a blockchain-based data sharing system, we will present other patterns in the sequence applied in each layer to address healthcare-specific challenges described previously in Section 5.4. The resulting app provides a fundamental reference architecture for a health data sharing system, as well as other types of systems within the healthcare domain or other domains where similar design requirements apply.

5.5.2 Preventing Reentrancy Attack in the Blockchain

Design problem faced by blockchain-based apps. Despite the growing interest in using blockchain technology for healthcare, a lot of recent attacks on some of the major blockchain-based apps have raised security concerns regarding the use of this technology in especially the healthcare industry that requires compliance to strict security and privacy regulations. An infamous example of such attacks is the DAO attack [25] in which a reentrancy bug was discovered and exploited that causes then worth $30 million of Ethereum being stolen. Even though the immutability and decentralization properties of blockchain technology can provide tremendous value to the direct exchange of digital information, without proper design decisions made prior to deploying a system on-chain could yield destructive consequences.

Solution → Apply the Guarded Updated pattern to prevent unexpected reentrancy attacks. We deem attack prevention as the utmost important design consideration in the development cycle of a blockchain-based healthcare system and therefore introduce Guarded Updated as the second pattern in our pattern sequence after defining a base layer with the Layered Ring. The goal of this pattern is to provide software engineers with a pattern that prevents the serious reentrancy bug early on during the development cycle in order to help design the rest of the system wherever this pattern may apply.



```
contract VulnerableContract {
    mapping (address => uint) public balances;

    ... // code to deposit funds

    function withdraw() {
        if (!msg.sender.call.value(balances[msg.sender])()) { // vulnerable code
            throw;
        }
        balances[msg.sender] = 0;
    }
}

contract ExploitVulnerableContract {
  VulnerableContract public vc;

  ... // code to reference VulnerableContract object

  function () payable {
    vc.withdraw(); // This is the default fallback that can recursively
                   // execute the vulnerable code above
  }
}
```

Figure 5.4: Example Vulnerable Solidity Code of the Simplified Reentrancy Bug and its Exploitation

A simplified reentrancy bug that affected the DAO app appears in the code snippet shown in Figure 5.4. Function *withdraw* in the *VulnerableContract* sets the caller's balance after checking if the asset transfer to the caller (*msg.sender*) is successful. The attack in *ExploitVulnerableContract* exploits this vulnerability by calling the *withdraw* function in a fallback function that is executed by the *call.value* method, creating recursions that bypass the statement on line that sets the user balance supposedly after the vulnerable statement returns [206].

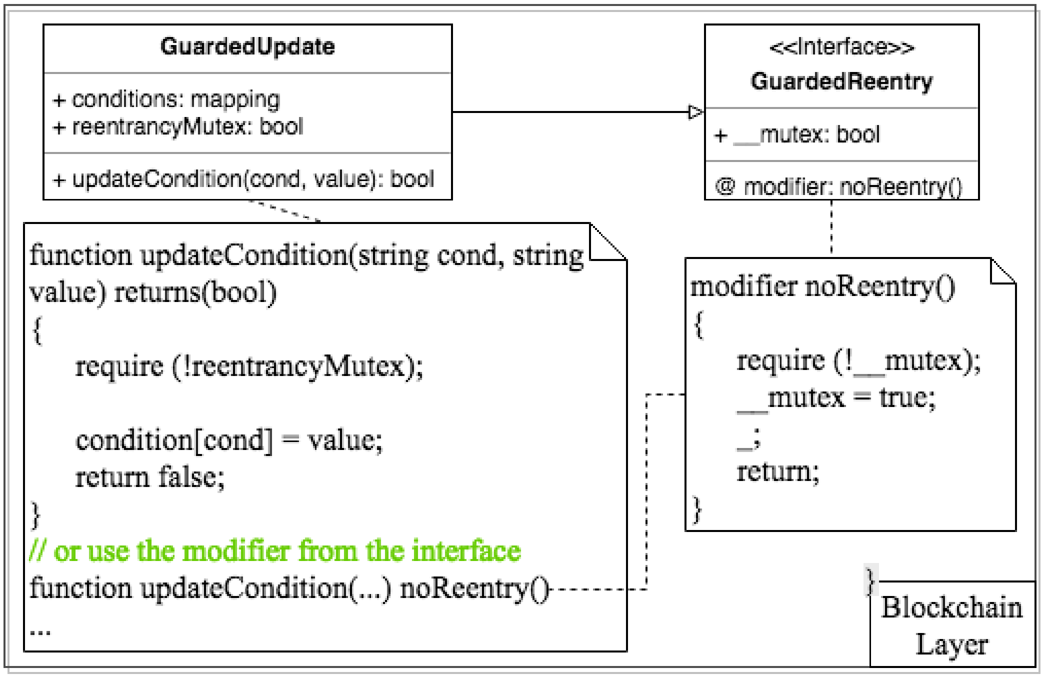
Although the reentrancy bug primarily targets cryptocurrencies in the interest of gaining financial returns, this bug could also plague systems designs for healthcare functions if prevention is not implemented in advance. As a key pattern in the sequence, GUARDED UPDATE aims to prevent reentrancy attack by ensuring atomic update to critical data in the blockchain-based healthcare system. The structure and code examples of this pattern appears in Figure 5.5[2]. As shown in the figure, a boolean guarding condition (i.e., reentrancyMutex) is used to control operations on protected state variable(s) (i.e., *conditions*). Once the variable(s) has been modified, the guarding condition can be reset to the initial state to permit other memory contexts to act upon the guarded data. Another, more systematic way to achieve this is to create a modifier in a Solidity interface contract, which can then be included in the declaration header of functions in other contracts.

Protecting atomic updates to state variables in the smart contracts prevents serious reentrancy attacks to occur, however, one major drawback is that atomic executions may slow down runtime performance of the system, particularly in a decentralized environment.

5.5.3 Separating Data from Logic to Ensure Data Availability via a Manager Contract

Design problem faced by blockchain-based apps. The immutability property of blockchains can ensure non-repudiation of data operations and/or transactions of data but can also become a major hurdle to data flow. On the one hand, immutability is important for

[2] The code examples are based on https://github.com/o0ragman0o/ReentryProtected



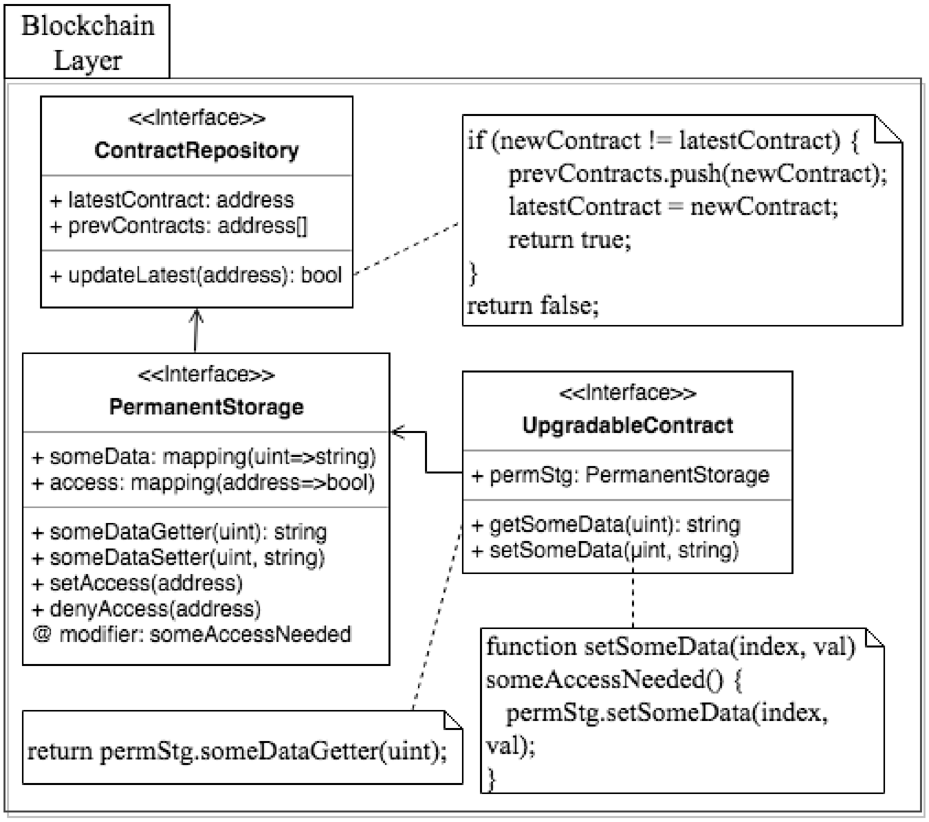
```
function updateCondition(string cond, string
value) returns(bool)
{
    require (!reentrancyMutex);

    condition[cond] = value;
    return false;
}
// or use the modifier from the interface
function updateCondition(...) noReentry()
...
```

```
modifier noReentry()
{
    require (!__mutex);
    __mutex = true;
    _;
    return;
}
```

Figure 5.5: Structure and Example Solidity Code Snippet of Guarded Update Pattern to Prevent Reentrancy Attacks on-Chain

achieving interoperability in a healthcare environment as it makes data objects (whether it is a reference pointer to a data store or an authorization request that grants a provider access to healthcare data) on the blockchain always available, even when one of the key maintainers of the network becomes unavailable. On the other hand, without a loosely-coupled design that focuses on clean separation of data and logic, immutability makes any upgrade to a blockchain-based health system hard to perform. Data, in such a system, does not only include information being exchanged across various network participants but also needs to contain meta data regarding the system that provides users with the most up-to-date knowledge regarding the system; whereas, logic refers to any operation or event that acts upon the data, typically implemented to read, update, or remove a data object.

Solution → Apply the Contract Manager to separate data from logic to ensure data availability via clean separation of concerns . The Contract Manager pattern aims to address the separation of data and logic via a *permanent storage* structure, which has been captured in [206]. Figure 5.6 presents the composition of this pattern.


Blockchain Layer
<<Interface>>
ContractRepository
+ latestContract: address
+ prevContracts: address[]
+ updateLatest(address): bool
<<Interface>>
PermanentStorage
+ someData: mapping(uint=>string)
+ access: mapping(address=>bool)
+ someDataGetter(uint): string
+ someDataSetter(uint, string)
+ setAccess(address)
+ denyAccess(address)
@ modifier: someAccessNeeded
<<Interface>>
UpgradableContract
+ permStg: PermanentStorage
+ getSomeData(uint): string
+ setSomeData(uint, string)

```
if (newContract != latestContract) {
     prevContracts.push(newContract);
     latestContract = newContract;
     return true;
}
return false;
```

```
return permStg.someDataGetter(uint);
```

```
function setSomeData(index, val)
someAccessNeeded() {
   permStg.setSomeData(index,
val);
}
```

Figure 5.6: Structure and Example Solidity Code of CONTRACT MANAGER Pattern for Maintaining Key Meta-Data on-Chain

Permanent storage maintains one or more data fields used throughout the system and provides permanent access to data with getter and setter functions for each one of the data fields. This ensures that all data used by the system remains readable even when logic contracts are outdated. Additionally, contract manager stores a *Contract Repository* of meta data that describes versions of the system (including but is not limited to addresses of the latest logic contract components and history contract addresses). To better ensure upgradeability of the system, CONTRACT MANAGER also defines access privilege of smart contracts by allowing the original owner of the storage contract to configure an access group for delegating or revoking certain or all rights of accessing or manipulating the data to other members to prevent data locking.

One disadvantage to the introduction of CONTRACT MANAGER is that all other logic contracts must execute additional calls to this contract for versioning checks and data queries.

5.5.4 Standardized On-Chain Interfaces to Off-Chain Storage Access

Design problem faced by blockchain-based apps. EHR systems have served the U.S. healthcare for decades and, unavoidably, have accumulated enormous amounts of valuable medical records that either exist in legacy systems or in more modern certified EHRs. Health data sharing today is only possible between healthcare professionals using the same EHR systems or compatible health information exchange services, which are exactly the third-party reliance that blockchain technology helps eliminate with its decentralized, trustless infrastructure. The direct exchange of digital information on the blockchain is only possible if such information or its representation is encoded on the blockchain with some degree of verifiable integrity. Due to the scale and privacy of healthcare data, it is unrealistic to store encrypted or hashed version of the actual data on the blockchain. Furthermore, it is impractical to create a blockchain-based system that completely replaces existing EHR systems or duplicates their functionality. The design of a scalable and standardized component that connects existing EHR data to a decentralized system offering interoperable data sharing is therefore needed.

Solution → Apply the DATABASE CONNECTOR pattern to ensure on-chain storage scalability and interoperability via standardized and minimal interfaces to off-chain storage. Figure 5.7 presents the composition of the DATABASE CONNECTOR pattern. The *Database Connector* component defines a standardized interface between the blockchain and storage layers. The interface provides an abstraction of the heterogeneous health data silos (e.g., EHR or other LFQ databases and HFQ data) to expose only minimal amount of information regarding each data source to the blockchain layer. As shown in Figure 5.7, the interface may only need to capture the name or description of a data source, some ”meta data" providing reference pointers to the data source, and a verifiable digital signature of the data source owner that provides some level of integrity. *Database Connector* is also closely associated with the DATABASE PROXY pattern (discussed next in Section 5.5.5) that uses a *Connector Handler* component in the blockchain layer to provide data access to the connector.

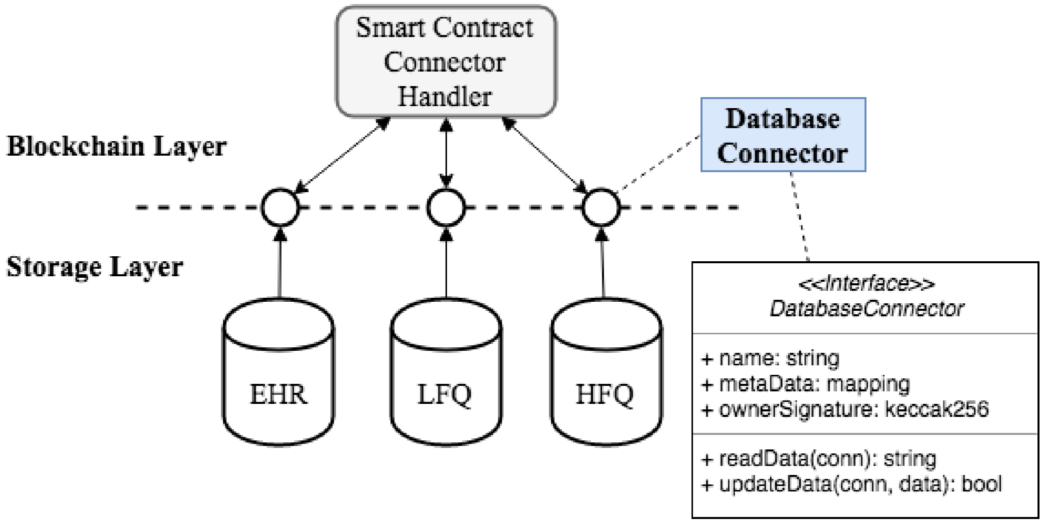


Figure 5.7: Structure of the DATABASE CONNECTOR Pattern Used to Standardize on-Chain Interfaces to off-Chain Storage Access

The main benefits of DATABASE CONNECTOR are (1) the storage scalability it provides on the blockchain that allows efficient sharing of connectors and (2) a standardized interface that unifies the on-chain representation of off-chain databases. The drawback is the additional implementations that are required for creating connectors to existing databases.

5.5.5 Sanity Checking before Accessing Off-Chain Storage

Design problem faced by blockchain-based apps. If a blockchain-based healthcare app must expose sensitive data or metadata (such as patient identifying information) on the blockchain, it must be designed to maximize health data privacy while facilitating health information exchange. In particular, a fundamental aspect of a blockchain is that data and all change history stored on-chain are public, immutable, and verifiable. For financial transactions focused on proving that transfer of an asset occurred, these properties are critical. When the goal is to store data in the blockchain, however, it is important to understand how these properties will impact the use case.

For example, storing patient data in the blockchain can be problematic since it requires that data be public and immutable. Although data can be encrypted before being stored, should all patient data be publicly distributed to all blockchain nodes? Even if encryption is used, the encryption technique may be broken in the future or defects in the implementation of the encryption algorithms or protocols used may make the data decryptable in the future. Immutability, on the other hand, prevents owners of the data from removing the data change history from the blockchain if a security flaw is found. Many other scenarios, ranging from discovery of medical mistakes in the data to changing data standards may necessitate the need to change the data over time.

In scenarios where the data may need to be changed, the public and immutable nature of the blockchain creates a fundamental tension that must be resolved. On the one hand, healthcare providers would like incorruptible data so its integrity is preserved. At the same time, providers want the data changeable and secure to protect patient privacy and account for possible errors. An interoperable app should protect patient privacy and also ensure data integrity.

Solution → Apply the DATABASE PROXY pattern to provide an additional layer of security by performing lightweight tasks before permitting access to database connectors. DATABASE PROXY is akin to the traditional PROXY pattern [209] with a slightly different focus unique to a blockchain-based design. To reduce computational costs on-chain, the *Database Proxy* interface defines some lightweight representation or placeholder for the real data object and encodes some lightweight security checks or auditing tasks until retrieval of the original data object is required.

Figure 5.8 illustrates the structure of DATABASE PROXY pattern and its interaction with the *Database Connector* object described previously in Section 5.5.4.

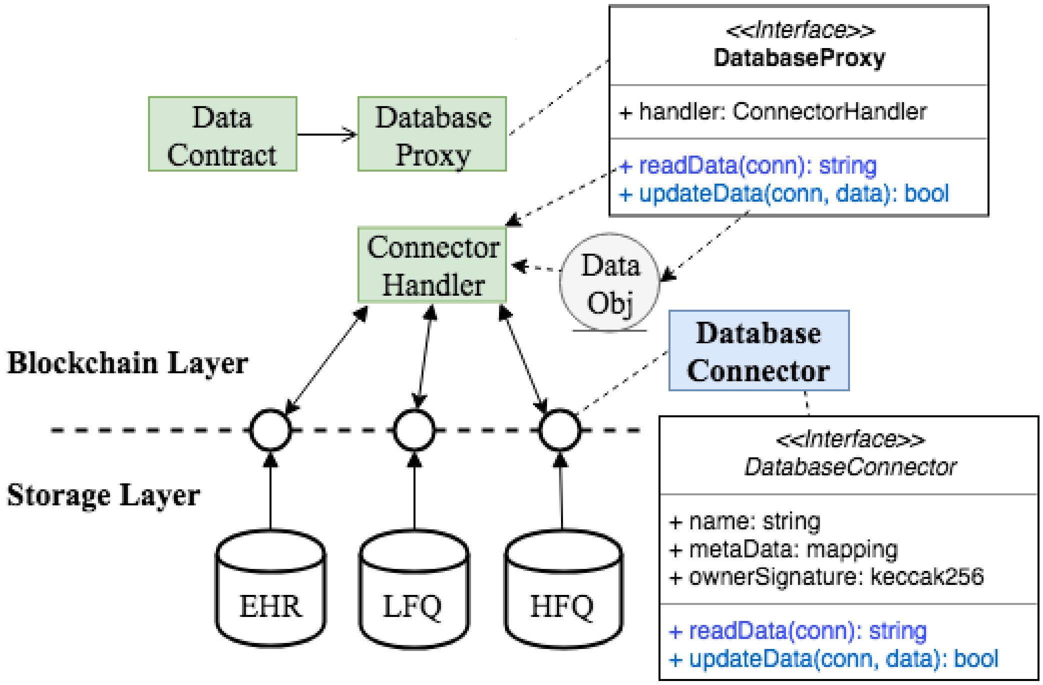


Figure 5.8: Composition of DATABASE PROXY Pattern for Performing Additional Security Checks before Accessing off-Chain Data Store

The *Database Proxy* interface maintains a reference to a *Connector Handler* object that forwards the read and write access to the appropriate *Database Connector* for access databases in the storage layer of the system. Each read request and modification operation through the *Connector Handler* can be logged in an immutable audit trail that is transparent to the entire blockchain network for verification against data corruption. In the case of a proxified contract (i.e., *Database Connector* that has somewhat heavyweight implementation) being updated with a new storage configuration (*e.g.*, when a data source has been introduced a new management system that requires some change in its *Database Connector* abstraction), the interface to the proxy contract can remain unchanged, encapsulating lower-level implementation variations.

As with the traditional PROXY pattern [209], a proxy object can perform lightweight housekeeping operations, such as security checks of administrative access and auditing tasks that log existing data requests, by storing some commonly used metadata in its internal states before retrieving the actual data. This component follows the same interface as the real object and can execute the original data object's function implementations as needed. It provides an additional layer for securing access to the real data object. However, *Database Proxy* may cause disparate behavior when the real object is accessed directly by some other component in the system while the proxy surrogate is accessed by others. It also creates an additional level of indirection for accessing actual data objects.

5.5.6 Managing Healthcare Entities and Other Types of Common Data on-Chain at Scale

Design problem faced by blockchain-based apps. All data and transaction records maintained in the blockchain are replicated and distributed to every node in the network. In a public blockchain, to compensate blockchain miners for contributing expensive hardware to store and maintain on-chain data, fees are charged based on the storage requirement of an application. Although a fee is not necessarily charged in a consortium blockchain with like-minded parties, other forms of compensation may exist to provide some incentives for the decentralized network maintainers. To minimize on-chain storage burden, a blockchain-based healthcare app that requires storage of some data on-chain must maximize data sharing among entities thus limit the amount of information stored.

In a large-scale healthcare setting, if a blockchain is used to store patient billing data, there will be millions of records replicated on all blockchain miner nodes. Moreover, billing data could include detailed patient insurance information, such as their ID#, insurance contact information, coverage details, and other aspects that the provider needs to bill for services. Capturing all this information for every patient can generate excessive amounts of data in the blockchain.

Suppose it is necessary to store a patient's insurance and billing information (encrypted) in the blockchain. Most patients are covered by one of a relatively small subset of insurers (in comparison to the total number of patients, *e.g.*, each insurance policy may cover

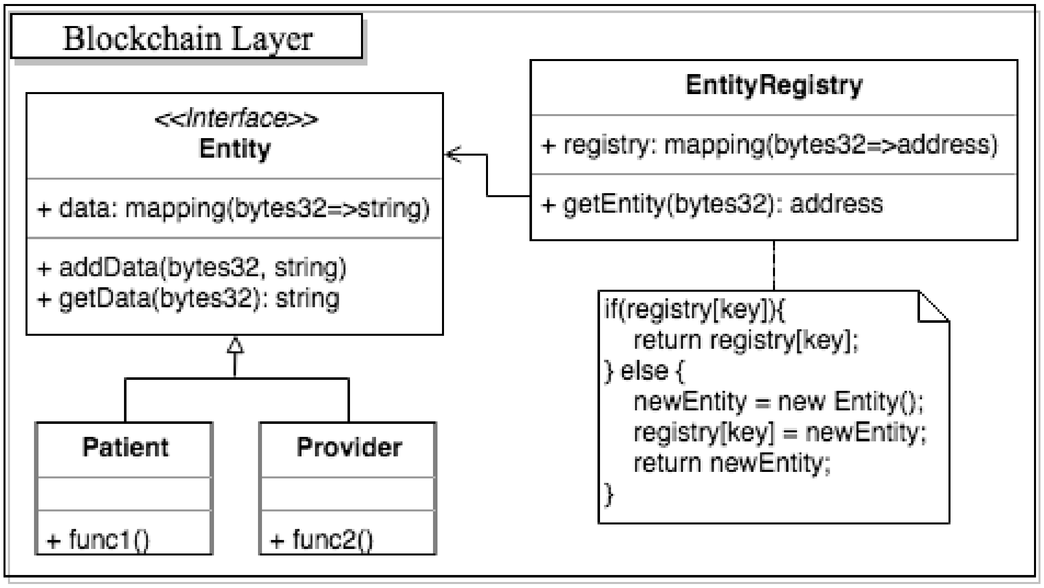
10,000s or 100,000s of patients). Therefore, a substantial amount of intrinsic, non-varying information is common across patients that can be reused and shared, such as details on what procedures are covered by an insurance policy. To bill for a service, however, this common intrinsic information must be combined with extrinsic information (such as the patient's ID#) that is specific to each patient.

A good design should maximize sharing of such common data to reduce on-chain storage cost and meanwhile have the capability to provide complete data objects on demand.

Solution → Apply the ENTITY REGISTRY pattern for managing healthcare entities on-chain at scale. As shown in Figure 5.9, the ENTITY REGISTRY mimics the traditional FLYWEIGHT pattern [209] with a factory [30] object to help manage healthcare entities on-chain at scale. In particular, *getEntity* uses a factory to create entity objects and maintain references (addresses) to created Entity objects in a common smart contract (i.e., *Entity Registry*). It internalizes common data across a number of *Entity*'s *data* field while externalizing varying data storage in entity-specific contracts (such as *Patient* or *Provider* entity). Using references (*i.e.*, addresses) to entity-specific contracts stored in the *registry*, combined extrinsic and intrinsic data can be retrieved upon request to return a complete dataset.

Applying this pattern to the earlier scenario, shared patient insurance information is stored only once in the *registry*, avoiding an exorbitant amount of memory usage from saving repeated data in all patient accounts. Varying, patient-specific billing information is stored in corresponding patient-specific entity contracts.

The registry can also maintain a mapping between unique entity identifiers and the referencing addresses of already deployed entity contracts to prevent account duplication. At account creation, only if no account with the specified entity identifier exists in the registry does it deploy a new entity contract; otherwise the registry retrieves the address associated with the existing entity contract. To retrieve complete insurance and billing information of a particular patient, clients need only invoke a function call from the registry


Blockchain Layer
<<Interface>>
Entity
+ data: mapping(bytes32=>string)
+ addData(bytes32, string)
+ getData(bytes32): string
EntityRegistry
+ registry: mapping(bytes32=>address)
+ getEntity(bytes32): address
Patient
+ func1()
Provider
+ func2()

```
if(registry[key]){
    return registry[key];
} else {
    newEntity = new Entity();
    registry[key] = newEntity;
    return newEntity;
}
```

Figure 5.9: ENTITY REGISTRY Pattern Used with a Factory to Manage Entities and Other Types of Common Data while Minimizing on-Chain Storage Requirements

with the patient identifier to obtain the combined intrinsic and extrinsic data object.

ENTITY REGISTRY provides better management for the large pool of objects (such as user accounts in the example above). It minimizes redundancy in similar objects by maximizing data and operation sharing. Particularly in the insurance example, if common insurance policy details are not extracted from each patient's contract, the cost to change a policy detail will be immense–it will require rewriting a huge number of impacted contracts. Data sharing with flyweight registry helps minimize the cost to change the common state in objects stored on-chain.

Although applying the ENTITY REGISTRY pattern creates an additional transaction to verify and include in the blockchain (*i.e.*, the flyweight object instantiation) before it can be used, this extra step can be outweighed by the resulted efficiency in data management.

5.5.7 Securing and Recording Data Access

Design problem faced by blockchain-based apps. Smart contracts are powerful for automating executions of predefined agreements directly between involved entities especially when the entities are registered on the blockchain using its native cryptographic keys and agreed terms are simple updates to cryptocurrency wallets/balances that are easy to update. The direct exchange of healthcare data unfortunately cannot easily be achieved on-chain due to the complexity and variability in the warehouses and management systems data resides in. Even when data sharing is made possible in such a decentralized environment, the shared information should not be available to the entire network, unlike an app that involves cryptocurrency. Instead, proper authorizations of sensitive health data access must be safeguarded.

Solution → Apply the TOKENIZED EXCHANGE pattern to authorize access to off-chain data storage and maintain a verifiable data access history. Variability of off-chain data sources can be encapsulated with a standardized interface that encodes a set of attributes describing the sources and some basic operations acting upon them (i.e., functions to retrieve the original data source and verify the digital signature to ensure data is originated from the expected sender.). Figure 5.5.7 presents the structure of the TOKENIZED EXCHANGE pattern in the sequence that defines a *Token* interface off-chain to represent each data source in a more consistent manner. With this interface, the *Database Connector Object* from the DATABASE CONNECTOR pattern discussed in Section 5.5.4 that references an off-chain data source can be ”tokenized” off-chain with access authorizations being encoded to a standard format using secure encryption and signing algorithms. Types of algorithms employed along with public keys used to generate the tokens are captured by the attributes defined in the interface. Tokens generated are then stored on-chain in a shared *Token Registry* smart contract. *Token Registry* builds an audit trail of the creation, update, deletion, and access requests to each of the tokens. To retrieve the *Database Connector Object*, the recipient must possess the authorized party’s secret key in order to decrypt and retrieve the original data source via the DATABASE PROXY pattern presented in Section 5.5.5.

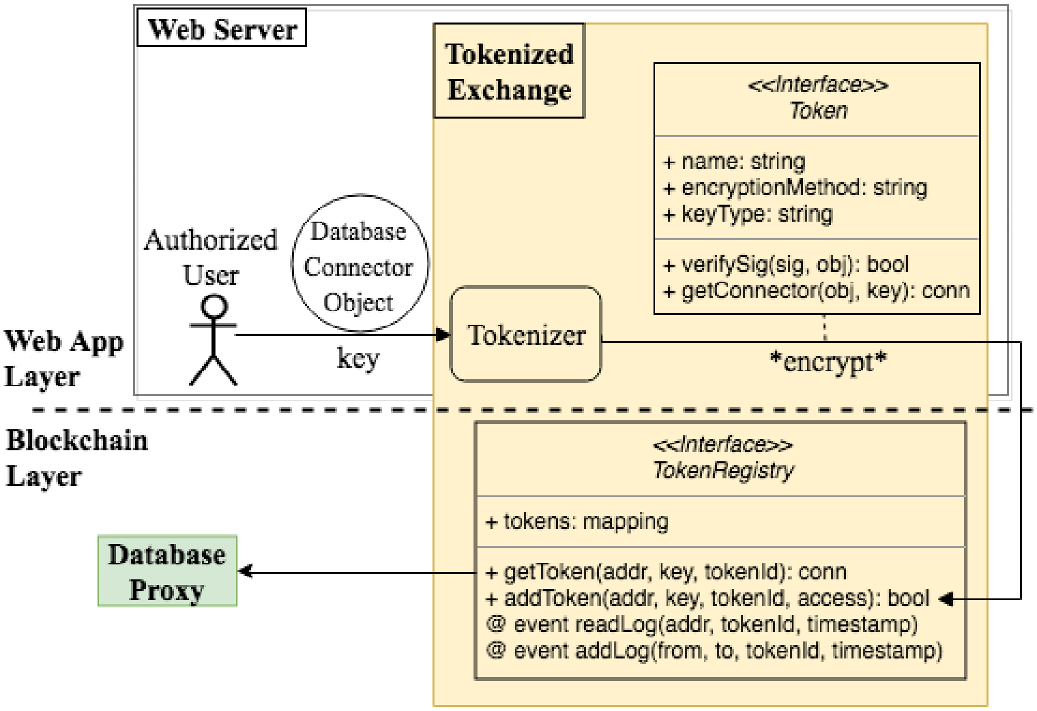

Web Server
Tokenized Exchange
<<Interface>>
Token
+ name: string
+ encryptionMethod: string
+ keyType: string
+ verifySig(sig, obj): bool
+ getConnector(obj, key): conn
Authorized User
Database Connector Object
key
Tokenizer
encrypt
Web App Layer
Blockchain Layer
<<Interface>>
TokenRegistry
+ tokens: mapping
+ getToken(addr, key, tokenId): conn
+ addToken(addr, key, tokenId, access): bool
@ event readLog(addr, tokenId, timestamp)
@ event addLog(from, to, tokenId, timestamp)
Database Proxy

Figure 5.10: Structure of TOKENIZED EXCHANGE Pattern for Authorizing off-Chain Data Access and Recording Verifiable Data Access Logs

This approach ensure that even when tokens carrying actual information of a particular data source are shared with a wide network, they can only be consumed by the intended recipient(s) with proper cryptographically paired keys. One drawback to this pattern is that there could be tokens that are not generalizable, in which case, implementations of other interfaces may be required.

5.5.8 Providing Notifications of Relevant Healthcare Activities at Scale

Design problem faced by blockchain-based apps. A blockchain-based healthcare system that needs to track relevant health changes across large patient populations must be designed to filter out useful health-related information from communication traffic (*i.e.*

transaction records) in the blockchain. For example, the Ethereum blockchain maintains an immutable record of contract creations and operation executions along with regular cryptocurrency transactions. The availability of this information makes blockchain a more autonomous approach to improve the coordination of patient care across different participants (*e.g.*, physicians, pharmacists, insurance agents, etc) who would normally communicate through various channels with a lot of manual effort, such as through telephoning or faxing. Due to the continually growing list of records on the blockchain, however, directly capturing any specific health-related topic from occurred events implies exhaustive transaction receipt lookups and topic filtering, which requires non-trivial computation and may result in delayed responses.

A good model should facilitate coordinated care and support relevant health information relays. For instance, health-related activities should be seamless communicated from the point when a patient self-reports illness (through a health DApp interface) to the point when they receive prescriptions created by their primary care provider; clinical reports and follow-up procedure should be relayed to and from the associated care provider offices in a timely manner.

Solution → Apply the *Publisher-Subscriber* pattern to manage user notifications at scale when events of interest occur across the decentralized network. Incorporating a notification service using the *Publisher-Subscriber* pattern [210] can facilitate scalable information filtering. In this design, changes in health activities are only broadcast to providers that subscribe to events relating to their patients. It alleviates tedious filtering of which care provider should be notified about patient activities as large volumes of transactions take place. It also helps maintain an interoperable environment that allows providers across various organizations to participate.

Due to the deterministic nature of blockchain that supports smart contracts, communications between the on-chain address space and off-chain services can only occur in two ways. The first way is a regular or constant poll, in which an off-chain server delegates a *Messenger* component to monitor changes and new events in the system. The second way pushes data out to an Oracle service, which is a trusted third-party that performs some computation off-chain and then forwards the results back to the blockchain address space via a callback function[3], such as in [218]. An Oracle service often charges a fee associated with its service provided to the blockchain and at its current stage today, it is not yet ideal for supporting large and sensitive data operations that are commonly experienced in a healthcare system.

The first variant avoids computation overhead on the blockchain because an off-chain server is responsible for querying and processing health events recorded on-chain. Specifically, when the publisher sends an update, its subscribers only need to do a simple update to an internal state variable that records the publisher's address, which the DApp server delegates a *Messenger* to actively monitor changes. When a change occurs, the responsibility for the heavy computational content filtering task (*e.g.,* retrieving the change activity from the publisher using the address) is delegated to the DApp server from the blockchain. The DApp server is context-aware at this point because each subscriber has an associated contract address accessible by the server. The *Messenger* can then filter the content based on subscribed topics and update the contract states of appropriate subscribers as needed.

The second variant shifts the responsibility of topic subscriptions and updates to the smart contract component on-chain. When a topic, such as a patient their provider wishes to be notified of any health-related activities, experiences a new event or has a value update, the smart contract logic that notifies the subscribers pushes the updated topic to an Oracle service, which executes some task related to the topic (e.g., sending a secure message to the subscriber regarding the updated event) and sends the result back to the smart contract caller upon task completion.

Figure 5.11 shows the two variants of PUBLISHER-SUBSCRIBER to provide the notification service.

[3] https://blockchainhub.net/blockchain-oracles/

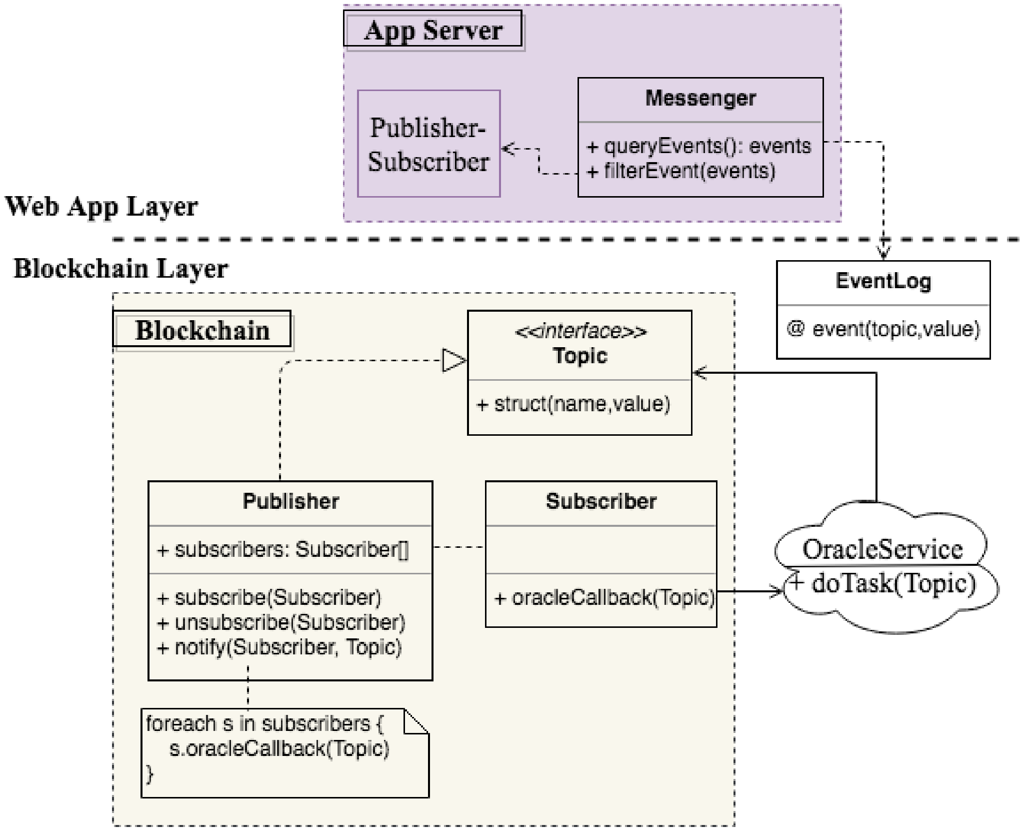


Figure 5.11: Two Variants of the PUBLISHER-SUBSCRIBER Pattern for Providing Clinical Notifications of Relevant Healthcare Activities at Scale

Implementing a notification service in a blockchain-based healthcare app is useful when a state change in the shared environment must be reported to interested parties without unmanaged many-to-many communications. The disadvantage to the ”poll” approach is the complexity in actual implementation of the messenger component that regularly monitors smart contract events, but it is much more efficient to unload the on-chain burden of topic filtering to off-chain services. The drawbacks to the ”push-to-oracle” approach are on-chain computation overhead and potential costs of Oracle services despite this approach being relatively easier to implement.

5.6 Conclusion

Blockchain and its programmable smart contracts provide a platform for creating decentralized apps that have the potential to improve healthcare interoperability. Leveraging this platform in a decentralized and transparent manner, however, requires that key design concerns be addressed. These concerns include—but are not limited to—system evolvability, storage requirements minimization, patient data privacy protection, and application scalability across large number of users. This chapter described these concerns and presented a key pattern sequence–LAYERED RING, GUARDED UPDATE, CONTRACT MANAGER, DATABASE CONNECTOR, DATABASE PROXY, ENTITY REGISTRY, TOKENIZED EXCHANGE, and PUBLISHER-SUBSCRIBER–that is designed to address these challenges.

Based on our experience developing the key pattern sequence, we learned the following lessons:

- The public, immutable, and verifiable properties of the blockchain enable a more interoperable environment that is not easily achieved using traditional approaches, which mostly rely on a centralized server or data storage.
- Each time a smart contract is modified, a new contract object is created on the blockchain. Important design decisions must therefore be made in advance to avoid the cost and storage overhead from contract interface change.
- To best leverage these properties of blockchain in the healthcare context, concerns regarding system evolvability, storage costs, sensitive information privacy, and application scalability must be taken into account.
- Combining time-proven design practices with domain-knowledge that focus on better leveraging properties of blockchain technology enables the creation of systems that are more modular, easier to integrate and maintain, and less susceptible to change.

Chapter 6

CONCLUDING REMARKS

6.1 Summary of Key Research Contributions

The design and applications of a learned filtering architecture (LFA) for extracting scalable insights from high-frequency, low-fidelity healthcare data:

- Proposed the generalized learned filter architecture based on advanced data science methods.

- Described the application of LFA in a case study focused on understanding the self-management behavior of adolescents with type 1 diabetes using novel data collected via the ecological momentary assessment methods.

- Showed that the LFA can help understand key psychosocial factors impacting self-management behavior in adolescents and at the same time, reduce the scale of EMA data collection.

- Demonstrated the potential value of EMA data in improving clinical decision-making and just-in-time patient support with positive results from the pilot study.

- Presented the application of LFA in the case study of hand hygiene compliance monitoring to characterize hand hygiene behavior in healthcare workers and move towards an intelligent compliance monitoring process.

- Detailed the process of data acquisition, data preprocessing, experiment setup, and posing then validating hypotheses regarding the hand hygiene compliance data leveraging the LFA.

- Proposed a hand hygiene compliance monitoring app (called ”HyPo”), based on insights learned from the empirical studies, as a service to alleviate the manual monitoring effort, reduce errors, and complement the compliance advocacy within a health organization.

The design of FHIRChain, a decentralized architecture enabling the secure and scalable sharing of healthcare data to improve collaborative clinical decision support using blockchain technology and the FHIR data standards:

- Provided an in-depth analysis of key ONC technical requirements and their implications for blockchain-based health IT systems

- Detailed the technical components of FHIRChain, a blockchain-based architecture designed to meet ONC requirements by encapsulating the HL7 FHIR standard for clinical data exchange

- Described a FHIRChain-based decentralized app using digital health identities to authenticate participants for a case study of collaborative decision making in a remote tumor board

- Demonstrated the potential of blockchain to foster effective healthcare data sharing while maintaining the security of original data sources

Well-documented software engineering practice via blockchain-focused design patterns and anti patterns to guide software developers in their design and implementation of evolvable and secure health IT systems:

- Presented and analyzed limitations of using naive blockchain solutions for healthcare.

- Demonstrated an end-to-end case study of a blockchain-based health IT system prototype as a motivating example before applying the software patterns we recommend.

- Provided detailed documentation of design patterns targeting blockchain-based healthcare apps.

- Illustrated how our proposed software patterns can help design blockchain-based apps to address specific challenges targeting healthcare and to avoid breaches to sensitive health data, using the same case study.

6.2 Future Work

6.2.1 Towards precision behavior medicine for adolescents with type 1 diabetes

Applying the learned filtering architecture to the pilot study of this research, we demonstrated the potential value of novel psychosocial data (collected via the MyDay mobile app) in understanding self-management behavior of adolescents with type 1 diabetes. As future work, these results can be extended to help enhance the MyDay system's ability to utilize unobtrusive indicators as much as possible. For example, experimental unobtrusive indicators of mealtimes are in development and if successful would greatly enhance our methodological approach [219]. In addition, the LFA machine learning methods employed in this research should be applied to a large diverse sample of patients to confirm and expand results reported in this chapter.

6.2.2 Towards autonomous monitoring of hand hygiene compliance in healthcare facilities

With the application of LFA, we were able to preliminarily characterize the hand hygiene behavior of healthcare workers. To expand upon these results, more compliance data should be collected, ideally, using the same process as detailed in Chapter 3. Additional data can be fed into the LFA to fine tune the parameters and also filter thresholds for various ML classifiers to increase classification accuracy and potentially further reduce the data size. Simulations of the HyPo app in other clinical environment(s) can also help observe compliance improvement rate and validate whether the improvement can be sustained over time in a range of caregiving settings.

6.2.3 Comparative analysis of FHIRChain implemented with different blockchain configurations

In our research, the initial FHIRChain prototype is implemented in the Ethereum blockchain, which is designed for apps and services open to the public. The limitation with the current version of the Ethereum blockchain for a healthcare system is transaction speed due to its underlying consensus mechanism (*i.e.*, Proof of Work) employed. In future work, simulations can be refined to evaluate and compare the performance of FHIRChain design implemented under a wide range of blockchain configurations beyond Ethereum. These simulations can be easily deployed to a testbed environment. For example, Amazon Web Services [220] may be used for such deployments following the blockchain template it provides.

6.2.4 More generalized and robust blockchain-focused software patterns for healthcare systems

The blockchain technology is still at its early stage with enhancements and new smart contract programmable features being introduced constantly. Despite new changes in designing blockchain-based apps, the fundamental property of a truly decentralized blockchain will remain, which means newer vulnerabilities may surface. Although the software patterns we proposed would theoretically be applicable to any blockchain infrastructure, we focused our discussions of the recommendations around the Ethereum implementation. One direction of future work may therefore be extracting more generalized patterns from ones proposed in this research to provide principles that are agnostic to specific blockchain infrastructure(s). Another future research direction is to evaluate the efficacy of applying these software patterns (*e.g.,* via performance metrics related to time and cost of computations or assessment metrics related to its feasibility) compared to other alternative designs (such as designs without using software patterns). Finally, generalized software patterns for creating healthcare-specific blockchains may also be worth investigating.

6.3 Summary of Publications

- Journal and Book Chapter Publications:

  1. **Peng Zhang**, Jules White, Douglas C. Schmidt, Gunther Lenz, S. Trent Rosenbloom, FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data, submitted to the Elsevier Computational and Structural Biotechnology Journal – Blockchain and Distributed Ledger Technologies in Biology, Medicine, and eHealth Special Issue, 2018 (to appear) [21].

  2. **Peng Zhang**, Douglas C. Schmidt, Jules White, and Gunther Lenz, Blockchain Technology Use Cases in Healthcare, Blockchain Technology: Platforms, Tools, and Use Cases, edited by Ganesh Deka, 2018 [22].

  3. **Peng Zhang**, Breck Stodghill, Cory Pitt, Cavan Briody, Douglas C. Schmidt, Jules White, Alan Pitt, and Kelly Aldrich, OpTrak: Tracking Opioid Prescriptions via Distributed Ledger Technology, submitted to the International Journal of Information Systems and Social Change (IJISSC), Special Issue On: Blockchain Technology: Platforms, Tools, and Use Cases, IGI Global, 2018 (to appear) [29].

- Conference Publications:

  1. Zhongwei Teng, **Peng Zhang**, Xiao Li, William Nock, Marcelino Rodriguez-Cancio, Denis Gilmore, Jules White, Douglas C. Schmidt, and Jonathan C. Nesbitt, Authentication and Usability in mHealth Apps, 2018 IEEE International Conference on E-health Networking, Application & Services (Healthcom), 17-20 September 2018, Ostrava, Czech Republic (to appear) [221].

  2. **Peng Zhang**, Douglas C. Schmidt, Jules White, and Shelagh A. Mulvaney,

Towards Precision Behavioral Medicine with IoT: Iterative Design and Optimization of a Self-Management Tool for Type 1 Diabetes, proceedings of the 2018 IEEE International Conference on Healthcare Informatics (ICHI 2018), New York, NY, USA, June 4-7, 2018 [70].

3. **Peng Zhang**, Douglas C. Schmidt, Jules White, Gunther Lenz, and Mike Walker, Metrics for Assessing Blockchain-based Healthcare Decentralized Apps, Proceedings of the IEEE Healthcom 2017, October 12-15, 2017, Dalian, China [162].

4. **Peng Zhang**, Jules White, Douglas C. Schmidt, and Gunther Lenz, Design of Blockchain-Based Apps Using Familiar Software Patterns to Address Interoperability Challenges in Healthcare, the 24th Pattern Languages of Programming conference, October 22-25, 2017, Vancouver, Canada [106].

5. Fangzhou Sun, **Peng Zhang**, Jules White, Douglas C. Schmidt, Jacob Staples, and Lee Krause, A Feasibility Study of Autonomically Detecting In-process Cyber-Attacks, Proceedings of the 3rd IEEE International Conference on Cybernetics (CYBCONF-2017), Special Session on Cyber Security [222].

6. **Peng Zhang**, Jules White, Douglas C. Schmidt, and Tom Dennis, Discussions of a Preliminary Hand Hygiene Compliance Monitoring Application-as-a-Service, 10th International Conference on Health Informatics - HEALTHINF 2017, 21 - 23 February, 2017, Porto, Portugal [105].

7. **Peng Zhang**, Jules White, Douglas C. Schmidt, and Tom Dennis, Applying Machine Learning Methods to Predict Hand Hygiene Compliance Characteristics, Proceedings of the Biomedical and Health Informatics Conference, Orlando, Florida, February 16-19, 2017 [111].

8. **Peng Zhang**, Sandeep Neema, and Ted Bapty, A Study of Collaborative Efforts and Proposed Visualizations in Domain-Specific Modeling Environment, 8th International Conference on Cyber-Enabled Distributed Computing and Knowl-

edge Discovery, Chengdu, China, October 13-15, 2016 [223]

9. **Peng Zhang**, Jules White, and Douglas C. Schmidt, HoliCoW: Automatically Breaking Team-based Software Projects to Motivate Student Testing, Proceedings to the Software Engineering Education and Training track at the 38th International Conference on Software Engineering Austin, TX, May 14 - 22, 2016 [224]

10. **Peng Zhang**, Z. Lattmann, J. Klingler, S. Neema, and T. Bapty, Visualization Techniques in Collaborative Domain-Specific Modeling Environment, Proceedings of SoutheastCon in Fort Lauderdale, FL, 9-12 April 2015 [225]

- Doctoral Consortium and Poster Publications:

  1. **Peng Zhang**, Jules White, Douglas C. Schmidt, Architectures and Patterns for Leveraging High-Frequency Low-Fidelity Data in Healthcare, Doctoral Consortium: Proceedings of the Sixth IEEE International Conference on Healthcare Informatics (ICHI 2018), New York, NY, USA, June 4-7, 2018 (Long talk) [12].

  2. **Peng Zhang**, Douglas C. Schmidt, Jules White, and Shelagh A. Mulvaney, Machine Learning Techniques Predict Diabetes Self-Care Behaviors in Adolescents: Precision Behavioral Medicine, Vanderbilt Diabetes Day Thursday, November 16, 2017.

  3. **Peng Zhang**, Jules White, Tom Dennis, Preliminary Experiment with Using Deep Learning to Characterize Hand Hygiene Compliance, Invited to Present at Stanford Medicine X 2017 Research Track Poster Session, Palo Alto, California, September 15-17, 2017

  4. **Peng Zhang**, Jules White, Ted Bapty, Tom Dennis, Logan Buchanan, Jeff Kimble, and Priscilla Knolle, An Initial Feasibility Study on Algorithmically Influencing Hand Washing Behavior by Characterizing Hand Hygiene Compliance Related Data, Poster session presented at the Tennessee Emerging Infections

Program 17th Annual Scientific Presentation Day, Nashville, TN, USA, October 6, 2016.

BIBLIOGRAPHY

[1] David F Doolan, David W Bates, and Brent C James. The use of computers for clinical care: a case series of advanced us sites. *Journal of the American Medical Informatics Association*, 10(1):94–107, 2003.

[2] Molla S Donaldson, Janet M Corrigan, Linda T Kohn, et al. *To err is human: building a safer health system*, volume 6. National Academies Press, 2000.

[3] Micky Tripathi. Ehr evolution: policy and legislation forces changing the ehr. *Journal of AHIMA*, 83(10):24–29, 2012.

[4] US Department of Health, Human Services, et al. Hitech act enforcement interim final rule. *US Department of*, 2009.

[5] David Blumenthal and Marilyn Tavenner. The meaningful use regulation for electronic health records. *N Engl J Med*, 2010(363):501–504, 2010.

[6] Dustin Charles, Meghan Gabriel, and Michael F Furukawa. Adoption of electronic health record systems among us non-federal acute care hospitals: 2008–2013. *ONC data brief*, 9:1–9, 2013.

[7] Mobile phone ownership over time. *Web. http://www.pewinternet.org/fact-sheet/mobile/*.

[8] Healthcare apps battle to be taken seriously. *Web. https://www.ft.com/content/ed3268f2-e620-11e5-a09b-1f8b0d268c39*, Last Accessed: 2018-08-29.

[9] Chang-Se Oh, Min-Seok Seo, Jung-Hyuck Lee, Sang-Hyun Kim, Young-Don Kim, and Hyun-Ju Park. Indoor air quality monitoring systems in the iot environment. *The Journal of Korean Institute of Communications and Information Sciences*, 40(5):886–891, 2015.

[10] Patient generated health data: Challenges & opportunities. *Web. https://www.cleardata.com/knowledge-hub/patient-generated-health-data-challenges-opportunities/*.

[11] Patient-generated health data. *Web. https://www.healthit.gov/topic/scientific-initiatives/patient-generated-health-data*, Last Accessed: 2018-08-29.

[12] Peng Zhang, Jules White, and Douglas Schmidt. Architectures and patterns for leveraging high-frequency, low-fidelity data in the healthcare domain. In *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 463–464. IEEE, 2018.

[13] Michael Shapiro, Douglas Johnston, Jonathan Wald, and Donald Mon. Patient-generated health data. *RTI International, April*, 2012.

[14] The shift to managing more patients with fewer resources. *Web. http://www.healthcareitnews.com/sponsored-content/shift-managing-more-patients-less-resources-0*, Last Accessed: 2018-08-29.

[15] Your next blood test could cost $10,000. *Web. http://time.com/3112985/your-next-blood-test-could-cost-10000/*.

[16] CostHelper.com. How much does a glucose meter cost? *Web. https://health.costhelper.com/glucose-meter.html*, Last Accessed: 2018-08-29.

[17] C Lee Ventola. Mobile devices and apps for health care professionals: uses and benefits. *Pharmacy and Therapeutics*, 39(5):356, 2014.

[18] William A Wood, Antonia V Bennett, and Ethan Basch. Emerging uses of patient generated health data in clinical research. *Molecular oncology*, 9(5):1018–1024, 2015.

[19] Joshua R Vest and Larry D Gamm. Health information exchange: persistent challenges and new strategies. *Journal of the American Medical Informatics Association*, 17(3):288–294, 2010.

[20] EHRIntelligence. Ehr compatibility and connectivity: two obstacles to patient care. *Web. https://ehrintelligence.com/news/ehr-compatibility-and-connectivity-two-obstacles-to-patient-care*, 2012.

[21] Peng Zhang, Jules White, Douglas C. Schmidt, Gunther Lenz, and S. Trent Rosenbloom. Fhirchain: Applying blockchain to securely and scalably share clinical data. *Computational and Structural Biotechnology Journal*, 16:267 – 278, 2018.

[22] Peng Zhang, Douglas C. Schmidt, Jules White, and Gunther Lenz. Blockchain technology use cases in healthcare. Advances in Computers. Elsevier, 2018.

[23] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[24] Vitalik Buterin et al. Ethereum white paper, 2013.

[25] David Siegel. Understanding the dao attack. *Web. http://www.coindesk.com/understanding-dao-hack-journalists*, 2016.

[26] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3), 2006.

[27] K DeSalvo and E Galvez. Connecting health and care for the nation: a shared nationwide interoperability roadmapversion 1.0. *Health IT Buzz*, 2015.

[28] Duane Bender and Kamran Sartipi. Hl7 fhir: An agile and restful approach to healthcare information exchange. In *Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium on*, pages 326–331. IEEE, 2013.

[29] Peng Zhang, Breck Stodghill, Cory Pitt, Cavan Briody, Douglas C. Schmidt, Jules White, Alan Pitt, and Kelly Aldrich. Optrak: Tracking opioid prescriptions via distributed ledger technology. *the International Journal of Information Systems and Social Change (IJISSC), Special Issue On: Blockchain Technology: Platforms, Tools, and Use Cases*, 2018.

[30] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Design patterns: Abstraction and reuse of object-oriented design. In *European Conference on Object-Oriented Programming*, pages 406–431. Springer, 1993.

[31] Alexander Shvets. Design patterns explained simply. *sourcemaking. com*, pages 80–84, 2015.

[32] James Coplien, Daniel Hoffman, and David Weiss. Commonality and variability in software engineering. *IEEE software*, 15(6):37–45, 1998.

[33] Etherescan - the ethereum blockchain explorer. *Web. https://etherscan.io/accounts/c*, Last Accessed: 2018-08-29.

[34] Andrea T Borchers, Raivo Uibo, and ME Gershwin. The geoepidemiology of type 1 diabetes. *Autoimmunity reviews*, 9(5):A355–A365, 2010.

[35] Dana Dabelea, Elizabeth J Mayer-Davis, Sharon Saydah, Giuseppina Imperatore, Barbara Linder, Jasmin Divers, Ronny Bell, Angela Badaru, Jennifer W Talton, Tessa Crume, et al. Prevalence of type 1 and type 2 diabetes among children and adolescents from 2001 to 2009. *JAMA*, 311(17):1778–1786, 2014.

[36] Li Wen, Ruth E Ley, Pavel Yu Volchkov, Peter B Stranges, Lia Avanesyan, Austin C Stonebraker, Changyun Hu, F Susan Wong, Gregory L Szot, Jeffrey A Bluestone, et al. Innate immunity and intestinal microbiota in the development of type 1 diabetes. *Nature*, 455(7216):1109–1113, 2008.

[37] Jamie R Wood, Kellee M Miller, David M Maahs, Roy W Beck, Linda A DiMeglio, Ingrid M Libman, Maryanne Quinn, William V Tamborlane, Stephanie E Woerner, T1D Exchange Clinic Network, et al. Most youth with type 1 diabetes in the t1d exchange clinic registry do not meet american diabetes association or international society for pediatric and adolescent diabetes clinical guidelines. *Diabetes care*, 36(7):2035–2037, 2013.

[38] Patricia A Cleary, William Dahms, David Goldstein, John Malone, and William V Tamborlane. Beneficial effects of intensive therapy of diabetes during adolescence: outcomes after the conclusion of the diabetes control and complications trial (dcct). *J Pediatr*, 139:804–812, 2001.

[39] Diabetes Control, Complications Trial Research Group, et al. The effect of intensive treatment of diabetes on the development and progression of long-term complications in insulin-dependent diabetes mellitus. *N Engl j Med*, 1993(329):977–986, 1993.

[40] Marisa E Hilliard, Maartje De Wit, Rachel M Wasserman, Ashley M Butler, Meredyth Evans, Jill Weissberg-Benchell, and Barbara J Anderson. Screening and support for emotional burdens of youth with type 1 diabetes: Strategies for diabetes care providers. *Pediatric Diabetes*, 2017.

[41] Deborah J Wiebe, Vicki Helgeson, and Cynthia A Berg. The social context of managing diabetes across the life span. *American Psychologist*, 71(7):526, 2016.

[42] Shelagh A Mulvaney, Korey K Hood, David G Schlundt, Chandra Y Osborn, Kevin B Johnson, Russell L Rothman, and Kenneth A Wallston. Development and initial validation of the barriers to diabetes adherence measure for adolescents. *Diabetes research and clinical practice*, 94(1):77–83, 2011.

[43] Stephanie L Fitzpatrick, Kristina P Schumann, and Felicia Hill-Briggs. Problem

solving interventions for diabetes self-management and control: a systematic review of the literature. *Diabetes research and clinical practice*, 100(2):145–161, 2013.

[44] Saul Shiffman, Arthur A Stone, and Michael R Hufford. Ecological momentary assessment. *Annu. Rev. Clin. Psychol.*, 4:1–32, 2008.

[45] Genevieve Dunton, Eldin Dzubur, Marilyn Li, Jimi Huh, Stephen Intille, and Rob McConnell. Momentary assessment of psychosocial stressors, context, and asthma symptoms in hispanic adolescents. *Behavior modification*, 40(1-2):257–280, 2016.

[46] Beth S Linas, Andrew Genz, Ryan P Westergaard, Larry W Chang, Robert C Bollinger, Carl Latkin, and Gregory D Kirk. Ecological momentary assessment of illicit drug use compared to biological and self-reported methods. *JMIR mHealth and uHealth*, 4(1), 2016.

[47] Erin E Brannon, Christopher C Cushing, Christopher J Crick, and Tarrah B Mitchell. The promise of wearable sensors and ecological momentary assessment measures for dynamical systems modeling in adolescents: A feasibility and acceptability study. *Translational behavioral medicine*, 6(4):558–565, 2016.

[48] Arianna Dagliati, Simone Marini, Lucia Sacchi, Giulia Cogni, Marsida Teliti, Valentina Tibollo, Pasquale De Cata, Luca Chiovato, and Riccardo Bellazzi. Machine learning methods to predict diabetes complications. *Journal of Diabetes Science and Technology*, page 1932296817706375, 2017.

[49] Youqing Wang, Jinping Zhang, Fanmao Zeng, Na Wang, Xiaoping Chen, Bo Zhang, Dong Zhao, Wenying Yang, and Claudio Cobelli. learning can improve the blood glucose control performance for type 1 diabetes mellitus. *Diabetes technology & therapeutics*, 19(1):41–48, 2017.

[50] Tao Zheng, Wei Xie, Liling Xu, Xiaoying He, Ya Zhang, Mingrong You, Gong Yang, and You Chen. A machine learning-based framework to identify type 2 dia-

betes through electronic health records. *International journal of medical informatics*, 97:120–127, 2017.

[51] Karina W Davidson and Ying Kuen Cheung. Envisioning a future for precision health psychology: innovative applied statistical approaches to n-of-1 studies. *Health Psychology Review*, 11(3):292–294, 2017.

[52] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[53] Centers for Disease Control, Prevention, US Department of Health, Human Services, et al. National diabetes fact sheet: national estimates and general information on diabetes and prediabetes in the united states, 2011. *Available from Accessed*, 3, 2012.

[54] Payal H Marathe, Helen X Gao, and Kelly L Close. American diabetes association standards of medical care in diabetes 2017. *Journal of diabetes*, 9(4):320–324, 2017.

[55] Shelagh Mulvaney, Russell Rothman, Chandra Osborn, Cindy Lybarger, Mary S Dietrich, and Kenneth Wallston. Self-management problem solving for adolescents with type 1 diabetes: Intervention processes associated with an internet program. 85:140–2, 10 2010.

[56] Sally A Huston and Christopher P Houk. Common sense model of illness in youth with type 1 diabetes or sickle cell disease. *The Journal of Pediatric Pharmacology and Therapeutics: JPPT*, 16(4):270, 2011.

[57] Shelagh A. Mulvaney. Improving patient problem solving to reduce barriers to diabetes self-management. *Clinical Diabetes*, 27(3):99–104, 2009.

[58] Russell E Glasgow, Lawrence Fisher, Marilyn Skaff, Joe Mullan, and Deborah J

Toobert. Problem solving and diabetes self-management. *Diabetes Care*, 30(1):33–37, 2007.

[59] Felicia Hll-Briggs, Denise C Cooper, Kimberly Loman, Frederick L Brancati, and Lisa A Cooper. A qualitative study of problem solving and diabetes control in type 2 diabetes self-management. *The Diabetes Educator*, 29(6):1018–1028, 2003.

[60] Tim Wysocki, Ronald Iannotti, Jill Weissberg-Benchell, Lori Laffel, Korey Hood, Barbara Anderson, Rusan Chen, and Family Management of Childhood Diabetes Steering Committee. Diabetes problem solving by youths with type 1 diabetes and their caregivers: measurement, validation, and longitudinal associations with glycemic control. *Journal of Pediatric Psychology*, 33(8):875–884, 2008.

[61] Saul Shiffman, Arthur A Stone, and Michael R Hufford. Ecological momentary assessment. *Annu. Rev. Clin. Psychol.*, 4:1–32, 2008.

[62] Kay Connelly, Karen F Stein, Beenish Chaudry, and Nicole Trabold. Development of an ecological momentary assessment mobile app for a low-literacy, mexican american population to collect disordered eating behaviors. *JMIR public health and surveillance*, 2(2), 2016.

[63] James Russell Pike, Bin Xie, Nasya Tan, Melanie Dee Sabado-Liwag, Annette Orne, Tupou Toilolo, Steven Cen, Vanessa May, Cevadne Lee, Victor Kaiwi Pang, et al. Developing an internet-and mobile-based system to measure cigarette use among pacific islanders: An ecological momentary assessment study. *JMIR mHealth and uHealth*, 4(1), 2016.

[64] Shelagh A Mulvaney, Russell L Rothman, Mary S Dietrich, Kenneth A Wallston, Elena Grove, Tom A Elasy, and Kevin B Johnson. Using mobile phones to measure adolescent diabetes adherence. *Health Psychology*, 31(1):43, 2012.

[65] Genevieve Fridlund Dunton, Audie A Atienza, Cynthia M Castro, and Abby C King. Using ecological momentary assessment to examine antecedents and correlates of physical activity bouts in adults age 50+ years: a pilot study. *Annals of Behavioral Medicine*, 38(3):249–255, 2009.

[66] Jacob Anhøj and Claus Møldrup. Feasibility of collecting diary data from asthma patients through mobile phones and sms (short message service): response rate analysis and focus group evaluation from a pilot study. *Journal of medical Internet research*, 6(4), 2004.

[67] Emilia Bielli, Fabio Carminati, Stella La Capra, Micaela Lina, Cinzia Brunelli, and Marcello Tamburini. A wireless health outcomes monitoring system (whoms): development and field testing with cancer patients using mobile phones. *BMC medical informatics and decision making*, 4(1):7, 2004.

[68] Vicki S Helgeson, Lindsey C Lopez, and Thomas Kamarck. Peer relationships and diabetes: retrospective and ecological momentary assessment approaches. *Health Psychology*, 28(3):273, 2009.

[69] Anja Hilbert, Winfried Rief, Brunna Tuschen-Caffier, Martina de Zwaan, and Julia Czaja. Loss of control eating and psychological maintenance in children: An ecological momentary assessment study. *Behaviour research and therapy*, 47(1):26–33, 2009.

[70] Peng Zhang, Douglas C Schmidt, Jules White, and Shelagh A Mulvaney. Towards precision behavioral medicine with iot: Iterative design and optimization of a self-management tool for type 1 diabetes. *2018 IEEE International Conference on Healthcare Informatics*.

[71] Marisa E Hilliard, P Joyce, Danielle Hessler, Ashley M Butler, Barbara J Anderson,

and Sarah Jaser. Stress and a1c among people with diabetes across the lifespan. *Current diabetes reports*, 16(8):1–10, 2016.

[72] Ran A Cai, Dominik Beste, Hema Chaplin, Socrates Varakliotis, Linda Suffield, Francesca Josephs, Debajit Sen, Lucy R Wedderburn, Yiannakis Ioannou, Stephen Hailes, et al. Developing and evaluating jiapp: Acceptability and usability of a smartphone app system to improve self-management in young people with juvenile idiopathic arthritis. *JMIR mHealth and uHealth*, 5(8), 2017.

[73] Amy Hughes Lansing, Cynthia A Berg, Jonathan Butner, and Deborah J Wiebe. Self-control, daily negative affect, and blood glucose control in adolescents with type 1 diabetes. *Health Psychology*, 35(7):643, 2016.

[74] Oksana Pugach, Donald Hedeker, Melanie J Richmond, Alexander Sokolovsky, and Robin Mermelstein. Modeling mood variation and covariation among adolescent smokers: application of a bivariate location-scale mixed-effects model. *Nicotine & Tobacco Research*, 16(Suppl_2):S151–S158, 2013.

[75] Tao Li and Ge Lin. Examining the role of location-specific associations between ambient air pollutants and adult asthma in the united states. *Health & place*, 25:26–33, 2014.

[76] J Schabert, JL Browne, K Mosely, and J Speight. Social stigma in diabetes: a framework to understand a growing problem for an increasing epidemic. patient 2013; 6: 1–10.

[77] Moshe Phillip, Thomas Danne, Shlomit Shalitin, Bruce Buckingham, Lori Laffel, William Tamborlane, and Tadej Battelino. Use of continuous glucose monitoring in children and adolescents. *Pediatric diabetes*, 13(3):215–228, 2012.

[78] Eleni I Georga, Vasilios C Protopappas, and Dimitrios I Fotiadis. Glucose prediction

in type 1 and type 2 diabetic patients using data driven techniques. In *Knowledge-oriented applications in data mining*. InTech, 2011.

[79] Youqing Wang, Xiangwei Wu, and Xue Mo. A novel adaptive-weighted-average framework for blood glucose prediction. *Diabetes technology & therapeutics*, 15(10):792–801, 2013.

[80] Fredrik Ståhl. *Diabetes mellitus glucose prediction by linear and Bayesian ensemble modeling*. PhD thesis, Department of Automatic Control, Lund University Sweden, 2012.

[81] Jorge Bondia, Cristina Tarín, Winston García-Gabin, Eduardo Esteve, José Manuel Fernández-Real, Wifredo Ricart, and Josep Vehí. Using support vector machines to detect therapeutically incorrect measurements by the minimed cgms®. *Journal of diabetes science and technology*, 2(4):622–629, 2008.

[82] Meysam Bastani. *Model-free intelligent diabetes management using machine learning*. PhD thesis, University of Alberta, 2014.

[83] Torben Biester, Olga Kordonouri, Martin Holder, Kerstin Remus, Dorothee Kieninger-Baum, Tanja Wadien, and Thomas Danne. let the algorithm do the work: Reduction of hypoglycemia using sensor-augmented pump therapy with predictive insulin suspension (smartguard) in pediatric type 1 diabetes patients. *Diabetes technology & therapeutics*, 19(3):173–182, 2017.

[84] Daniela Elleri, Janet M Allen, Martina Biagioni, Kavita Kumareswaran, Lalantha Leelarathna, Karen Caldwell, Marianna Nodale, Malgorzata E Wilinska, Carlo L Acerini, David B Dunger, et al. Evaluation of a portable ambulatory prototype for automated overnight closed-loop insulin delivery in young people with type 1 diabetes. *Pediatric diabetes*, 13(6):449–453, 2012.

[85] Michael J O'grady, Adam J Retterath, D Barry Keenan, Natalie Kurtz, Martin Cantwell, Glenn Spital, Michael N Kremliovsky, Anirban Roy, Elizabeth A Davis, Timothy W Jones, et al. The use of an automated, portable glucose control system for overnight glucose control in adolescents and young adults with type 1 diabetes. *Diabetes care*, 35(11):2182–2187, 2012.

[86] Boris P Kovatchev, Eric Renard, Claudio Cobelli, Howard C Zisser, Patrick Keith-Hynes, Stacey M Anderson, Sue A Brown, Daniel R Chernavvsky, Marc D Breton, Anne Farret, et al. Feasibility of outpatient fully integrated closed-loop control. *Diabetes care*, 36(7):1851–1858, 2013.

[87] Aaron E Carroll, Linda A DiMeglio, Stephanie Stein, and David G Marrero. Using a cell phone-based glucose monitoring system for adolescent diabetes management. *The Diabetes Educator*, 37(1):59–66, 2011.

[88] Miroslav Rusin, Eirik Årsand, and Gunnar Hartvigsen. Functionalities and input methods for recording food intake: a systematic review. *International journal of medical informatics*, 82(8):653–664, 2013.

[89] Joseph Tran, Rosanna Tran, and John R White. Smartphone-based glucose monitors and applications in the management of diabetes: an overview of 10 salient apps and a novel smartphone-connected blood glucose monitor. *Clinical Diabetes*, 30(4):173–178, 2012.

[90] Bharath Sudharsan, Malinda Peeples, and Mansur Shomali. Hypoglycemia prediction using machine learning models for patients with type 2 diabetes. *Journal of diabetes science and technology*, 9(1):86–90, 2014.

[91] Juan Li and Chandima Fernando. Smartphone-based personalized blood glucose prediction. *ICT Express*, 2(4):150–154, 2016.

[92] Yoshiyuki Kawano and Keiji Yanai. Real-time mobile food recognition system. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 1–7. IEEE, 2013.

[93] Katherine Ellis, Jacqueline Kerr, Suneeta Godbole, Gert Lanckriet, David Wing, and Simon Marshall. A random forest classifier for the prediction of energy expenditure and type of physical activity from wrist and hip accelerometers. *Physiological measurement*, 35(11):2191, 2014.

[94] Maged N Kamel Boulos, Shauna Gammon, Mavis C Dixon, Sandra M MacRury, Michael J Fergusson, Francisco Miranda Rodrigues, Telmo Mourinho Baptista, and Stephen P Yang. Digital games for type 1 and type 2 diabetes: underpinning theory with three illustrative examples. *JMIR Serious Games*, 3(1), 2015.

[95] Mulvaney SA, Vaala SV, Hood KK, Lybarger C, Carroll R, Williams L, Schmidt DC, Johnson K, Dietrich MS, and Laffel L. Mobile momentary assessment and biobehavioral feedback for adolescents with type 1 diabetes: feasibility and engagement patterns. *Diabetes Technology and Therapeutics*, 2018.

[96] iHealth Labs Inc. ihealth labs. *Web. https://ihealthlabs.com*, Last Accessed: 2018-08-29.

[97] Kellee M Miller, Nicole C Foster, Roy W Beck, Richard M Bergenstal, Stephanie N DuBose, Linda A DiMeglio, David M Maahs, and William V Tamborlane. Current state of type 1 diabetes treatment in the us: updated data from the t1d exchange clinic registry. *Diabetes care*, 38(6):971–978, 2015.

[98] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, 6(1):1–6, 2004.

[99] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[100] Ivan Tomek. An experiment with the edited nearest-neighbor rule. *IEEE Transactions on systems, Man, and Cybernetics*, (6):448–452, 1976.

[101] Nguyen Thai-Nghe, DT Nghi, and Lars Schmidt-Thieme. Learning optimal threshold on resampling data to deal with class imbalance. In *Proc. IEEE RIVF International Conference on Computing and Telecommunication Technologies*, pages 71–76, 2010.

[102] Lara Lusa et al. Smote for high-dimensional class-imbalanced data. *BMC bioinformatics*, 14(1):106, 2013.

[103] T Elhassan, M Aljurf, F Al-Mohanna, and M Shoukri. Classification of imbalance data using tomek link (t-link) combined with random under-sampling (rus) as a data reduction method. *Journal of Informatics and Data Mining*, 2016.

[104] Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas, et al. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36, 2006.

[105] Peng Zhang, Marcelino Rodriguez-Cancio, Douglas C Schmidt, Jules White, and Tom Dennis. Discussions of a preliminary hand hygiene compliance monitoring application-as-a-service. In *HEALTHINF*, pages 537–544, 2017.

[106] Peng Zhang, Jules White, Douglas C Schmidt, and Gunther Lenz. Applying software patterns to address interoperability in blockchain-based healthcare apps. *arXiv preprint arXiv:1706.03700*, 2017.

[107] Inbal Nahum-Shani, Shawna N Smith, Bonnie J Spring, Linda M Collins, Katie Witkiewitz, Ambuj Tewari, and Susan A Murphy. Just-in-time adaptive interventions (jitais) in mobile health: key components and design principles for ongoing health behavior support. *Annals of Behavioral Medicine*, 2016.

[108] WHO. A guide to the implementation of the WHO multimodal hand hygiene improvement strategy. http://www.who.int/iris/handle/10665/70030, 2009.

[109] John M. Boyce, Timothea Cooper, and Michael J. Dolan. Evaluation of an Electronic Device for Real-Time Measurement of Alcohol-Based Hand Rub Use. *Infection Control & Hospital Epidemiology*, 30(11):1090–1095, November 2009.

[110] Tim Eckmanns, Jan Bessert, Michael Behnke, Petra Gastmeier, and Henning Rüden. Compliance with antiseptic hand rub use in intensive care units the hawthorne effect. *Infection Control*, 27(09):931–934, 2006.

[111] Peng Zhang, Jules White, Douglas Schmidt, and Tom Dennis. Applying machine learning methods to predict hand hygiene compliance characteristics. (ISIS-16-101), 11/2016 2016.

[112] B EFORE. Hand hygiene technical reference manual. 2009.

[113] Subarna K Shrestha, Venkata CK Sunkesula, Sirisha Kundrapu, Myreen E Tomas, Michelle M Nerandzic, and Curtis J Donskey. Acquisition of clostridium difficile on hands of healthcare personnel caring for patients with resolved c. difficile infection. *Infection Control & Hospital Epidemiology*, 37(04):475–477, 2016.

[114] D. J. Gould, N. S. Drey, and S. Creedon. Routine hand hygiene audit by direct observation: has nemesis arrived? *The Journal of Hospital Infection*, 77(4):290–293, April 2011.

[115] Katherine Ellingson, Janet P. Haas, Allison E. Aiello, Linda Kusek, Lisa L. Maragakis, Russell N. Olmsted, Eli Perencevich, Philip M. Polgreen, Marin L. Schweizer, Polly Trexler, Margaret VanAmringe, and Deborah S. Yokoe. Strategies to Prevent Healthcare-Associated Infections through Hand Hygiene. *Infection Control and Hospital Epidemiology*, 35(8):937–960, 2014.

[116] Melissa A. Ward, Marin L. Schweizer, Philip M. Polgreen, Kalpana Gupta, Heather S. Reisinger, and Eli N. Perencevich. Automated and electronically assisted hand hygiene monitoring systems: A systematic review. *American Journal of Infection Control*, 42(5):472–478, May 2014.

[117] Luke F Chen, Charlene Carriker, Russell Staheli, Pamela Isaacs, Brandon Elliott, Becky A Miller, Deverick J Anderson, Rebekah W Moehring, Sheila Vereen, Judie Bringhurst, et al. Observing and improving hand hygiene compliance implementation and refinement of an electronic-assisted direct-observer hand hygiene audit program. *Infection Control & Hospital Epidemiology*, 34(2):207–210, 2013.

[118] Emily E. Sickbert-Bennett, Lauren M. DiBiase, Tina M. Schade Willis, Eric S. Wolak, David J. Weber, and William A. Rutala. Reducing health careassociated infections by implementing a novel all hands on deck approach for hand hygiene compliance. *American Journal of Infection Control*, 44(5, Supplement):e13–e16, May 2016.

[119] Donna Armellino, Manish Trivedi, Isabel Law, Narendra Singh, Mary Ellen Schilling, Erfan Hussain, and Bruce Farber. Replicating changes in hand hygiene in a surgical intensive care unit with remote video auditing and feedback. *American Journal of Infection Control*, 41(10):925–927, October 2013.

[120] C. R. Davis. Infection-free surgery: how to improve hand-hygiene compliance and

eradicate methicillin-resistant Staphylococcus aureus from surgical wards. *Annals of the Royal College of Surgeons of England*, 92(4):316–319, May 2010.

[121] Alexandre R. Marra, Luciana Reis Guastelli, Carla Manuela Pereira de Arajo, Jorge L. Saraiva dos Santos, Luiz Carlos R. Lamblet, Moacyr Silva, Gisele de Lima, Ruy Guilherme Rodrigues Cal, ngela Tavares Paes, Miguel Cendoroglo Neto, Luciana Barbosa, Michael B. Edmond, and Oscar Fernando Pavo dos Santos. Positive Deviance A New Strategy for Improving Hand Hygiene Compliance. *Infection Control &amp; Hospital Epidemiology*, 31(1):12–20, January 2010.

[122] Daniel J. Morgan, Lisa Pineles, Michelle Shardell, Atlisa Young, Katherine Ellingson, John A. Jernigan, Hannah R. Day, Kerri A. Thom, Anthony D. Harris, and Eli N. Perencevich. Automated hand hygiene count devices may better measure compliance than human observation. *American Journal of Infection Control*, 40(10):955–959, December 2012.

[123] Morkos Fakhry, George B. Hanna, Oliver Anderson, Alison Holmes, and Dinesh Nathwani. Effectiveness of an audible reminder on hand hygiene adherence. *American Journal of Infection Control*, 40(4):320–323, May 2012.

[124] Andrew G. Sahud and Nitin Bhanot. Measuring hand hygiene compliance: a new frontier for improving hand hygiene. *Infection Control and Hospital Epidemiology*, 30(11):1132, November 2009.

[125] M. B. Edmond, A. Goodell, W. Zuelzer, K. Sanogo, K. Elam, and G. Bearman. Successful use of alcohol sensor technology to monitor and report hand hygiene compliance. *The Journal of Hospital Infection*, 76(4):364–365, December 2010.

[126] Alexandre R. Marra, Thiago Zinsly Sampaio Camargo, Thyago Pereira Magnus, Rosangela Pereira Blaya, Gilson Batista Dos Santos, Luciana Reis Guastelli, Rodrigo Dias Rodrigues, Marcelo Prado, Elivane da Silva Victor, Humberto Bogos-

sian, Julio Cesar Martins Monte, Oscar Fernando Pavo dos Santos, Carlos Kazume Oyama, and Michael B. Edmond. The use of real-time feedback via wireless technology to improve hand hygiene compliance. *American Journal of Infection Control*, 42(6):608–611, June 2014.

[127] Richard T. Ellison, Constance M. Barysauskas, Elke A. Rundensteiner, Di Wang, and Bruce Barton. A Prospective Controlled Trial of an Electronic Hand Hygiene Reminder System. *Open Forum Infectious Diseases*, page ofv121, August 2015.

[128] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[129] Deeplearning4j Development Team. Deeplearning4j: Open-source distributed deep learning for the jvm. *Apache Software Foundation License*, 2, 2016.

[130] John Platt et al. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[131] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[132] William A Gardner. Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique. *Signal Processing*, 6(2):113–133, 1984.

[133] Alex Graves. Supervised sequence labelling. In *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 5–13. Springer, 2012.

[134] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[135] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997. Special issue on relevance.

[136] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, San Mateo, 1995. Morgan Kaufmann.

[137] David E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.

[138] David A Schum. *The evidential foundations of probabilistic reasoning*. Northwestern University Press, 1994.

[139] Matthew Berman and Andrea Fenaughty. Technology and managed care: patient benefits of telemedicine in a rural health care network. *Health economics*, 14(6):559–573, 2005.

[140] Christian Castaneda, Kip Nalley, Ciaran Mannion, Pritish Bhattacharyya, Patrick Blake, Andrew Pecora, Andre Goy, and K Stephen Suh. Clinical decision support systems for improving diagnostic accuracy and achieving precision medicine. *Journal of clinical bioinformatics*, 5(1):4, 2015.

[141] Hardeep Singh, Traber Davis Giardina, Ashley ND Meyer, Samuel N Forjuoh, Michael D Reis, and Eric J Thomas. Types and origins of diagnostic errors in primary care settings. *JAMA internal medicine*, 173(6):418–425, 2013.

[142] Rainu Kaushal, Kaveh G Shojania, and David W Bates. Effects of computerized physician order entry and clinical decision support systems on medication safety: a systematic review. *Archives of internal medicine*, 163(12):1409–1416, 2003.

[143] Gordon D Schiff, Omar Hasan, Seijeoung Kim, Richard Abrams, Karen Cosby, Bruce L Lambert, Arthur S Elstein, Scott Hasler, Martin L Kabongo, Nela Kros-

njar, et al. Diagnostic error in medicine: analysis of 583 physician-reported errors. *Archives of internal medicine*, 169(20):1881–1887, 2009.

[144] Darren B Taichman, Joyce Backus, Christopher Baethge, Howard Bauchner, Peter W De Leeuw, Jeffrey M Drazen, John Fletcher, Frank A Frizelle, Trish Groves, Abraham Haileamlak, et al. Sharing clinical trial data: A proposal from the international committee of medical journal editorssharing clinical trial data. *Annals of internal medicine*, 164(7):505–506, 2016.

[145] Elizabeth Warren. Strengthening research through data sharing. *New England Journal of Medicine*, 375(5):401–403, 2016.

[146] Nophar Geifman, Jennifer Bollyky, Sanchita Bhattacharya, and Atul J Butte. Opening clinical trial data: are the voluntary data-sharing portals enough? *BMC medicine*, 13(1):280, 2015.

[147] Gary Edward Gross. The role of the tumor board in a community hospital. *CA: a cancer journal for clinicians*, 37(2):88–92, 1987.

[148] J Ricke and H Bartelink. Telemedicine and its impact on cancer management. *European Journal of Cancer*, 36(7):826–833, 2000.

[149] Christy L Marshall, Nancy J Petersen, Aanand D Naik, Nancy Vander Velde, Avo Artinyan, Daniel Albo, David H Berger, and Daniel A Anaya. Implementation of a regional virtual tumor board: a prospective study evaluating feasibility and provider acceptance. *Telemedicine and e-Health*, 20(8):705–711, 2014.

[150] Laura Levit, Alison P Smith, Edward J Benz Jr, and Betty Ferrell. Ensuring quality cancer care through the oncology workforce. *Journal of Oncology Practice*, 6(1):7–11, 2010.

[151] Mark Terry. Medical identity theft and telemedicine security. *Telemedicine and e-Health*, 15(10):1–5, 2009.

[152] Autumn S Downey, Steve Olson, et al. *Sharing clinical research data: workshop summary*. National Academies Press, 2013.

[153] George Hripcsak, Meryl Bloomrosen, Patti FlatelyBrennan, Christopher G Chute, Jim Cimino, Don E Detmer, Margo Edmunds, Peter J Embi, Melissa M Goldstein, William Ed Hammond, et al. Health data use, stewardship, and governance: ongoing gaps and challenges: a report from amia's 2012 health policy meeting. *Journal of the American Medical Informatics Association*, 21(2):204–211, 2014.

[154] Gunnar Hartvigsen, Monika A Johansen, Per Hasvold, Johan Gustav Bellika, Eirik Arsand, Eli Arild, Deede Gammon, Sture Pettersen, Steinar Pedersen, et al. Challenges in telemedicine and ehealth: lessons learned from 20 years with telemedicine in tromso. *Studies in health technology and informatics*, 129(1):82, 2007.

[155] Marlene Maheu, Pamela Whitten, and Ace Allen. *E-Health, Telehealth, and Telemedicine: a guide to startup and success*. John Wiley & Sons, 2002.

[156] Robert LaRose, Sharon Strover, Jennifer L Gregg, and Joseph Straubhaar. The impact of rural broadband development: Lessons from a natural field experiment. *Government Information Quarterly*, 28(1):91–100, 2011.

[157] André B Bondi. Characteristics of scalability and their impact on performance. In *Proceedings of the 2nd international workshop on Software and performance*, pages 195–203. ACM, 2000.

[158] Rachel L Richesson and Jeffrey Krischer. Data standards in clinical research: gaps, overlaps, challenges and future directions. *Journal of the American Medical Informatics Association*, 14(6):687–696, 2007.

[159] Reenita Das. Does blockchain have a place in healthcare. *Web. https://www.forbes.com/sites/reenitadas/2017/05/08/does-blockchain-have-a-place-in-healthcare/*.

[160] Matthias Mettler. Blockchain technology in healthcare: The revolution starts here. In *e-Health Networking, Applications and Services (Healthcom), 2016 IEEE 18th International Conference on*, pages 1–3. IEEE, 2016.

[161] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. Medrec: Using blockchain for medical data access and permission management. In *Open and Big Data (OBD), International Conference on*, pages 25–30. IEEE, 2016.

[162] P. Zhang, M. A. Walker, J. White, D. C. Schmidt, and G. Lenz. Metrics for assessing blockchain-based healthcare decentralized apps. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–4, Oct 2017.

[163] David Johnston, Sam Onat Yilmaz, Jeremy Kandah, Nikos Bentenitis, Farzad Hashemi, Ron Gross, Shawn Wilkinson, and Steven Mason. The general theory of decentralized applications, dapps. *GitHub, June*, 9, 2014.

[164] RJ Krawiec, Dan Housman, Mark White, Mariya Filipova, Florian Quarre, Dan Barr, Allen Nesbitt, Kate Fedosova, Jason Killmeyer, Adam Israel, et al. Blockchain: Opportunities for health care. In *Proc. NIST Workshop Blockchain Healthcare*, pages 1–16, 2016.

[165] Jeff Brandt Peter B. Nichol. Co-creation of trust for healthcare: The cryptocitizen. framework for interoperability with blockchain. 2016.

[166] IBM Global Business Services Public Sector Team. Blockchain: The chain of trust and its potential to transform healthcare our point of view. 2016.

[167] Tsung-Ting Kuo, Hyeon-Eui Kim, and Lucila Ohno-Machado. Blockchain distributed ledger technologies for biomedical and health care applications. *Journal of the American Medical Informatics Association*, 24(6):1211–1220, 2017.

[168] Kevin Peterson, Rammohan Deeduvanu, Pradip Kanjamala, and Kelly Boles. A blockchain-based approach to health information exchange networks, 2016.

[169] Alevtina Dubovitskaya, Zhigang Xu, Samuel Ryu, Michael Schumacher, and Fusheng Wang. Secure and trustable electronic medical records sharing using blockchain. *arXiv preprint arXiv:1709.06528*, 2017.

[170] Adrian Gropper. Powering the physician-patient relationship with hie of one blockchain health it, 2016.

[171] Direct project. *Web. https://www.healthit.gov/policy-researchers-implementers/direct-project*, Last Accessed: 2018-08-29.

[172] Gideon Greenspan. Blockchains vs centralized databases. *Web. https://www.multichain.com/blog/2016/03/blockchains-vs-centralized-databases*, Last Accessed: 2018-08-29.

[173] Centers for Disease Control, Prevention, et al. Hipaa privacy rule and public health. guidance from cdc and the us department of health and human services. *MMWR: Morbidity and mortality weekly report*, 52(Suppl. 1):1–17, 2003.

[174] Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

[175] Abdullah Al Omar, Mohammad Shahriar Rahman, Anirban Basu, and Shinsaku Kiyomoto. Medibchain: A blockchain based privacy preserving platform for healthcare data. In *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pages 534–543. Springer, 2017.

[176] Xiao Yue, Huiju Wang, Dawei Jin, Mingqiang Li, and Wei Jiang. Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *Journal of medical systems*, 40(10):218, 2016.

[177] Ryan Shea. Simple contracts are better contracts: What we can learn from the meltdown of the dao. *Web. https://medium.com/@ryanshea/simple-contracts-are-better-contracts-what-we-can-learn-from-the-dao-6293214bad3a*, Last Accessed: 2018-08-29.

[178] Dick Hardt. The oauth 2.0 authorization framework. 2012.

[179] Introduction to the isa. *Web. https://www.healthit.gov/isa/*, Last Accessed: 2018-08-29.

[180] MT Kim Futrell. Structured data. *Web. http://www.orchardsoft.com/files/white_paper_structured_data.pdf*, 2013.

[181] Kensaku Kawamoto, Tonya Hongsermeier, Adam Wright, Janet Lewis, Douglas S Bell, and Blackford Middleton. Key principles for a national clinical decision support knowledge sharing framework: synthesis of insights from leading subject matter experts. *Journal of the American Medical Informatics Association*, 20(1):199–207, 2013.

[182] Avraham Leff and James T Rayfield. Web-application development using the model/view/controller design pattern. In *Enterprise Distributed Object Computing Conference, 2001. EDOC'01. Proceedings. Fifth IEEE International*, pages 118–127. IEEE, 2001.

[183] Tatu Ylonen and Chris Lonvick. The secure shell (ssh) protocol architecture. 2006.

[184] Hugo Krawczyk. The order of encryption and authentication for protecting commu-

nications (or: How secure is ssl?). In *Advances in CryptologyCRYPTO 2001*, pages 310–331. Springer, 2001.

[185] Douglas Crockford. The application/json media type for javascript object notation (json). 2006.

[186] Hapi-fhir. *Web. http://fhirtest.uhn.ca/*, Last Accessed: 2018-08-29.

[187] Harold Ossher and Peri Tarr. Using multidimensional separation of concerns to (re) shape evolving software. *Communications of the ACM*, 44(10):43–50, 2001.

[188] R Michael Alvarez, Thad E Hall, and Alexander H Trechsel. Internet voting in comparative perspective: the case of estonia. *PS: Political Science & Politics*, 42(3):497–505, 2009.

[189] Robert H Dolin, Liora Alschuler, Sandy Boyer, Calvin Beebe, Fred M Behlen, Paul V Biron, and Amnon Shabo. Hl7 clinical document architecture, release 2. *Journal of the American Medical Informatics Association*, 13(1):30–39, 2006.

[190] Charles Rackoff and Daniel R Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Annual International Cryptology Conference*, pages 433–444. Springer, 1991.

[191] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE symposium on security and privacy (SP)*, pages 839–858. IEEE, 2016.

[192] Gideon Greenspan. Multichain private blockchain white paper. *Web. https://www.multichain.com/download/MultiChain-White-Paper.pdf*, 2015.

[193] Shimon Even, Oded Goldreich, and Yacov Yacobi. Electronic wallet. In *Advances in Cryptology*, pages 383–386. Springer, 1984.

[194] Gary Anthes. Estonia: a model for e-government. *Communications of the ACM*, 58(6):18–20, 2015.

[195] Adrian Blundell-Wignall. The bitcoin question: Currency versus trust-less transfer technology. *OECD Working Papers on Finance, Insurance and Private Pensions*, (37):1, 2014.

[196] Anne Geraci, Freny Katki, Louise McMonegal, Bennett Meyer, John Lane, Paul Wilson, Jane Radatz, Mary Yee, Hugh Porteous, and Fredrick Springsteel. *IEEE standard computer dictionary: Compilation of IEEE standard computer glossaries*. IEEE Press, 1991.

[197] Sandy Weininger, Michael B Jaffe, Michael Robkin, Tracy Rausch, David Arney, and Julian M Goldman. The importance of state and context in safe interoperable medical systems. *IEEE journal of translational engineering in health and medicine*, 4:1–10, 2016.

[198] Blockchain Hub. Blockchain oracles. *Web. https://insights.sei.cmu.edu/sei_blog/2017/07/what-is-bitcoin-what-is-blockchain.html*, 2017.

[199] Ethereum Foundation. Solidity. *Web. https://solidity.readthedocs.io/en/develop/*, 2015.

[200] Jules Dourlens. Ethereum smart contracts lifecycle. *Web. https://ethereumdev.io/ethereum-smart-contracts-lifecycle/*.

[201] Ethereum.io. Contracts. *Web. http://solidity.readthedocs.io/en/develop/contracts.html*, 2017.

[202] Simone Porru, Andrea Pinna, Michele Marchesi, and Roberto Tonelli. Blockchain-oriented software engineering: challenges and new directions. In *Proceedings of the*

39th International Conference on Software Engineering Companion, pages 169–171. IEEE Press, 2017.

[203] Massimo Bartoletti and Livio Pompianu. An empirical analysis of smart contracts: platforms, applications, and design patterns. *arXiv preprint arXiv:1703.06322*, 2017.

[204] Santiago Palladino. The parity wallet hack explained. *Web. https://blog.zeppelin.solutions/on-the-parity-wallet-multisig-hack-405a8c12e8f7*, 2017.

[205] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. A survey of attacks on ethereum smart contracts (sok). In *Principles of Security and Trust*, pages 164–186. Springer, 2017.

[206] ConsenSys. Recommendations for smart contract security in solidity. *Web. Recommendations for Smart Contract Security in Solidity*, 2018.

[207] ONC. Connecting health and care for the nation: A 10-year vision to achieve an interoperable health it infrastructure. 2014.

[208] Steven Rich and Barton Gellman. Nsa seeks to build quantum computer that could crack most types of encryption. *The Washington Post*, 2, 2014.

[209] Erich Gamma, John Vlissides, Ralph Johnson, and Helm. Richard. *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education, 1995.

[210] Frank Buschmann, Kelvin Henney, and Douglas Schimdt. *Pattern-oriented Software Architecture: on patterns and pattern language*, volume 5. John wiley & sons, 2007.

[211] Sima Ajami and Tayyebe Bagheri-Tadi. Barriers for adopting electronic health records (ehrs) by physicians. *Acta Informatica Medica*, 21(2):129, 2013.

[212] C Broderson, B Kalis, C Leong, E Mitchell, E Pupo, and A Truscott. Blockchain: Securing a new health interoperability experience, 2016.

[213] Alevtina Dubovitskaya, Zhigang Xu, Samuel Ryu, Michael Schumacher, and Fusheng Wang. Secure and trustable electronic medical records sharing using blockchain. *arXiv preprint arXiv:1709.06528*, 2017.

[214] Coin Market Cap. Cryptocurrency market capitalizations. *Web. https://coinmarketcap.com/*.

[215] Cryptokitties. *Web. https://www.cryptokitties.co/about*, Last Accessed: 2018-08-29.

[216] Fomo3d. *Web. https://exitscam.me/play*, Last Accessed: 2018-08-29.

[217] Idex - decentralized ethereum asset exchange. *Web. https://idex.market/*, Last Accessed: 2018-08-29.

[218] Ethereum Foundation. Oraclize limited. *Web. http://www.oraclize.it/*, 2015.

[219] Sediqeh Samadi, Mudassir Rashid, Kamuran Turksoy, Jianyuan Feng, Iman Hajizadeh, Nicole Hobbs, Caterina Lazaro, Mert Sevil, Elizabeth Littlejohn, and Ali Cinar. Automatic detection and estimation of unannounced meals for multivariable artificial pancreas system. *Diabetes technology & therapeutics*, 20(3):235–246, 2018.

[220] Amazon AWS. Aws blockchain templates. *Web. https://aws.amazon.com/blockchain/templates/*, Last Accessed: 2018-08-29.

[221] Zhongwei Teng, Peng Zhang, Xiao Li, William Nock, Marcelino Rodriguez-Cancio, Denis Gilmore, Jules White, Douglas C Schmidt, and Jonathan C Nesbitt. Authentication and usability in mhealth apps. In *International Conference on E-health Networking, Application & Services (Healthcom)*. IEEE, 2018.

[222] Fangzhou Sun, Peng Zhang, Jules White, Douglas Schmidt, Jacob Staples, and Lee Krause. A feasibility study of autonomically detecting in-process cyber-attacks. In

2017 3rd IEEE International Conference on Cybernetics (CYBCONF), pages 1–8. IEEE, 2017.

[223] Peng Zhang, Sandeep Neema, and Ted Bapty. A study of collaborative efforts and proposed visualizations in domain-specific modeling environment. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2016 International Conference on*, pages 146–152. IEEE, 2016.

[224] Peng Zhang, Jules White, and Douglas C Schmidt. Holicow: automatically breaking team-based software projects to motivate student testing. In *Software Engineering Companion (ICSE-C), IEEE/ACM International Conference on*, pages 436–439. IEEE, 2016.

[225] Peng Zhang, Zsolt Lattmann, James Klingler, Sandeep Neema, and Ted Bapty. Visualization techniques in collaborative domain-specific modeling environment. In *SoutheastCon 2015*, pages 1–6. IEEE, 2015.