6-1-2018

Balancing patient control and practical access policy for electronic health records via blockchain technology

Elena Horton
Dartmouth College

Follow this and additional works at: https://digitalcommons.dartmouth.edu/senior_theses

Part of the Computer Sciences Commons

Recommended Citation
https://digitalcommons.dartmouth.edu/senior_theses/136

This Thesis (Undergraduate) is brought to you for free and open access by the Theses and Dissertations at Dartmouth Digital Commons. It has been accepted for inclusion in Dartmouth College Undergraduate Theses by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.
Balancing patient control and practical access policy for electronic health records via blockchain technology

Senior Honors Thesis by: Elena Horton
1 June 2018

Advised by: Sean Smith
Department of Computer Science
Dartmouth College
elena.n.horton.18@dartmouth.edu

Abstract

Electronic health records (EHRs) have revolutionized the health information technology domain, as patient data can be easily stored and accessed within and among medical institutions. However, in working towards nationwide patient engagement and interoperability goals, recent literature adopts a very patient-centric model—patients own their universal, holistic medical records and control exactly who can access their health data. I contend that this approach is largely impractical for healthcare workflows, where many separate providers require access to health records for care delivery. My work investigates the potential of a blockchain network to balance patient control and provider accessibility with a two-fold approach. First, I conduct a survey investigation to identify patient concerns and determine the level of control patients would like over their health information. Second, I implement a blockchain network prototype to address the spectrum of patient control preferences and automate practical access policy. There are conflicting demands amongst patients and providers for EHR access—privacy versus flexibility. Yet, I find blockchain technology, when manipulated to model access states, automate an organizational role-based access scheme, and provide an immutable history of behavior in the network, to be a very plausible solution for balancing patient desires and provider needs. My approach is, to my knowledge, the first example of blockchain’s use for less patient-centric, nudge theory-based EHR access control, an idea that could align access control interests as academics, the government, and the healthcare industry make strides towards interoperable, universal patient records.
## Contents

1 Introduction ................................................. 4

2 Background .................................................. 7

   2.1 Blockchain ............................................. 8

   2.2 EHR access problems and blockchain’s potential .......... 10

3 Related Work ............................................... 13

4 Survey Investigation ........................................ 17

   4.1 Demographics ......................................... 18

   4.2 Results ............................................... 19

   4.3 Survey Conclusions .................................... 26

5 Blockchain-backed access control system prototype ........ 27

   5.1 Blockchain features and the EHR access problem ......... 27

   5.2 Assumptions .......................................... 30

   5.3 Requirements .......................................... 32

   5.4 Technologies Used ..................................... 33

   5.5 Access Model .......................................... 35

   5.6 Implementation Structure ................................ 39

   5.7 Smart contracts ........................................ 45

   5.8 Additional Programmability .............................. 50

   5.9 Issues ................................................. 51
6 Evaluation

6.1 Functional Analysis .................................................. 54
6.2 Practical Analysis ...................................................... 60
6.3 Limitations ............................................................... 62
6.4 Results ................................................................. 63

7 Future Work ............................................................. 65

8 Conclusion ............................................................... 66

9 Acknowledgements .................................................... 67

10 References .............................................................. 68

11 Appendix ............................................................... 77
1 Introduction

Much of the quality of healthcare delivery relies on providers’ abilities to easily obtain and record a patient’s health data and information. Electronic health records (EHRs) have revolutionized the health information technology domain, as patient data can now be digitally stored and accessed within and among medical institutions. However, despite significant adoption of EHRs, the failure of effective interoperability amongst varied providers has been identified as a key barrier to improving healthcare delivery [32, 41, 42]. Additionally, as patients leave scattered bits of information across the varied providers with whom they interact, recent literature addressing EHR improvement, which will be later explained as prior work, has taken a focus on patient-centric health records—these records give patients agency over their entire medical history and control to share their information with providers at their own discretion [1, 2, 3, 7, 8, 10, 11, 14, 15, 17, 18, 31, 36, 37]. As most EHR systems were not originally designed to be used holistically or cross-institutionally over a patient’s lifetime, this different schema of medical record—patient-owned and stored outside of any single medical institution—has also been coined the “Personal Health Record” (PHR), although terminology remains fuzzy within the healthcare IT realm. There is a spectrum of PHR versus EHR architectures. The ONC’s Shared Nationwide Interoperability Roadmap aims to melt this classification with their government-sponsored push to create holistic patient engagement and full interoperability amongst EHRs in America [43].

However, while having a patient-owned, universal, interoperable EHR system in America is a worthy goal, there exists a tradeoff between patient-centrism and provider practicality regarding the accessibility of PHR/EHR systems. Quality of care relies on various
individual providers having access to the patient information they need when they need it. Current medical organizations have developed their own access policies for their EHR systems, basing these primarily on role-based access control (RBAC) schemes [22, 23, 44]. These schemes allow for the easy flow of information amongst medical professionals, especially considering the sheer number of care providers patients may encounter on any given hospital visit—one health policy expert attributes her interest in the field to her experience of a simple procedure in which over twenty providers tended to her throughout the visit [24]. Consequently, once patients are given fine-grained autonomy over access to their medical history, the practical workflow of healthcare delivery disintegrates, as they likely cannot or will not grant specific access to the many individual providers who may require it. Very few patients understand their data and, more specifically, what data is critical to their care [25]. Also, such patient-centric systems are only plausible if they are widely adopted and will only be widely adopted if they provide value to the public [3, 4].

The commercial failures of Google Health and Microsoft HealthVault speak directly to this story. In these systems, patients had to record their own health data and could not easily share this information with their doctors, who likely would not trust the self-entered data anyways. Patient-owned health records must still be accessible and curated by their providers [9, 22, 24, 25]. Value appears in quality of care, improved patient outcomes, and patient engagement with their healthcare; all of these aspects require the maintaining of current provider workflows for care delivery [9, 48, 49, 50]. Therefore, a large challenge remains in striking a balance between patient-centrism and provider accessibility for PHR/EHR systems to maintain value through the healthcare ecosystem.

In my work, I explore the potential of a blockchain solution to address access policy
in a patient-owned PHR/EHR system. Despite blockchain’s introduction as a key part of
the cryptocurrency Bitcoin, further development has added programmability to the dis-
tributed ledger—logic known as *smart contracts* securely automate changes to the state of
the blockchain ledger, a distributed data structure, via defined rules [6, 27, 30, 47]. Blockchain
technology has been touted for its potential to improve healthcare in many facets due to
its features of decentralized management, immutable audit trails, data provenance, and
cryptographically enforced security [12, 35, 38, 39, 45, 46]. Specifically, ongoing projects
have been investigating its use for accelerating medical research, enhancing insurance claims,
and, most notably for this work, developing a decentralized, interoperable, patient-centered
EHR system [7, 38]. However, as these projects are in early stages, their actual practicality
in the complicated real world healthcare setting remains unclear.

As provider accessibility presents a major barrier to the practicality and eventual
success of universal, patient-owned health records, my work investigates the potential of
a blockchain network to balance patient control and provider accessibility with a two-fold
approach. First, I conduct a user survey to identify patient concerns and determine the
level of control patients would like over their health information. Second, I implement a
blockchain network prototype to address the spectrum of patient control preferences and
automate practical, existing access policy for providers on a distributed scale.

My work departs from recent literature and early-stage healthcare blockchain projects
by countering the direction of patient-centric control over EHRs and acknowledging the
practical needs of providers to access information. However, at the same time, by utilizing
a blockchain network for its immutable transaction history and securely enforced logic
via *smart contracts*, my system maintains patients’ ownership, access transparency, and
selective control over their health records. Thus, this work contributes to ongoing research regarding blockchain’s potential for healthcare delivery by incorporating an understanding of patient desires for access control into a more practical proof of concept to be considered while working towards the national goal of a holistic, interoperable PHR/EHR system. When looking at the larger problem of achieving universal, patient-owned, interoperable medical records, I focus specifically on how access to medical records can be managed. Simply put, in terms of real world access policy, my work addresses the need “to increase [patient] control without losing flexibility” [29].

The rest of this paper proceeds as follows. Chapter 2 provides background information on blockchain technology and current EHR access concerns. Chapter 3 outlines several examples of prior work that have informed this project. Chapter 4 explains my approach and results from a user survey executed via Amazon Mechanical Turk. Chapter 5 explains my approach and implementation of a blockchain EHR access control network prototype. Chapter 6 evaluates my solution for both functionality and practicality. Chapter 7 discusses goals for future work on this project, and Chapter 8 concludes.

2 Background

This chapter provides an overview of blockchain—the central technology used in this work—and further introduces specific implementations of blockchains and smart contracts. It then offers a background of electronic health record implementation and access policy used amongst American healthcare organizations, stressing how blockchain technology can address current challenges.
2.1 Blockchain

Blockchain is a distributed data structure technology that allows various, untrusted members of a network to share and maintain a commonly recognized network state. The blockchain records chronological, verified transactions into an immutable chain of events, where each transaction updates a distributed ledger stored on all participating nodes within the network. Blockchain was originally introduced as a key component of the cryptocurrency, Bitcoin, in which the blockchain is a public ledger for all financial transactions amongst untrusted users [30]. However, a blockchain itself can easily be decoupled from its original cryptocurrency identity—it is simply a growing log of records, a series of transactions each updating the ledger, batched into time-stamped blocks. Each block is identified by a unique cryptographic hash that also incorporates the previous block, hence the chain aspect of blockchain. Any network participant can read through the ordered, back-linked list of blocks to determine the current state of the data ledger for the network.

Blockchain employs asymmetric cryptography for network participation and transaction validation. Users sign their transactions with their private key and address other participants by their public keys. These users interact with the blockchain via nodes—each
node holds a copy of the blockchain and can serve as a point of entry for one or more participants. Similarly, nodes can play distinct roles in the network depending on the blockchain implementation [51]. As users create, sign, and submit new transactions, nodes broadcast these transactions out to network peers who must work to reach a universal version of the ledger state by ordering and validating transactions; this process is known as reaching consensus. Again, the manner in which consensus is reached can vary by implementation. Most relevant to my work, a key implementation distinction exists between permissionless and permissioned blockchains.

A permissionless blockchain has anonymous or pseudonymous participants and allows all users to append transactions and new blocks to the ledger. On the other hand, permissioned blockchains know and control the participants’ identities via some form of identity management. This management framework is trusted to control access and users’ rights to participate in, create, or validate transactions. This distinction between permissioned and permissionless blockchains lends itself directly to two popular implementations of blockchain: Ethereum and Hyperledger.

Ethereum builds upon the original Bitcoin blockchain by introducing programmatic capabilities into a permissionless implementation [6]. Any user can create and execute transaction logic on the Ethereum Virtual Machine (EVM), and this code, known as a smart contract, executes autonomously on the blockchain via its own contract account. Transactions sent to such an account are paid for in ether, the platform’s cryptocurrency, and trigger the smart contract’s execution. Validation is achieved via a proof-of-work (PoW) process in which a nonce input is found to satisfy certain requirements of a new block’s resulting hash. This process, known as mining, has seen considerable criticism for its energy
costs and scalability concerns, and the Ethereum platform has made efforts to transition to alternative consensus mechanisms such as proof-of-stake (PoS) [52, 53, 54, 55]. Nonetheless, permissionless consensus mechanisms remain an open problem.

Hyperledger is an open source, permissioned, programmable blockchain platform sponsored by the Linux Foundation [27]. As a permissioned system, users are known and managed by the platform’s Membership Service, which issues Enrollment certificates upon user registration. As a programmable platform, smart contracts, known as chaincode, define how transactions will be executed and how the ledger state will change. The smart contracts run in their own Docker containers and their output is verified by the network’s chosen consensus mechanism, often a state machine replication algorithm such as Practical Byzantine Fault Tolerance (PBFT) [27, 51, 53, 56]. Hyperledger has fully separated the blockchain structure from a cryptocurrency, and it allows developers to create unique networks with diverse transactions, varied types of participants, and programmed business logic. My work utilizes the Hyperledger framework to develop a permissioned network for EHR access control. In this manner, known, registered users can interact with the network based on the permissions and capabilities associated with their cryptographically verified identities.

2.2 EHR access problems and blockchain’s potential

According to surveys published in 2017, 67% of healthcare providers reported using electronic health records, and 89% of office-based physicians had adopted EHRs; however, experts estimate that over 94% of hospitals and over 86% of individual providers
use EHRs as of 2018 [25, 57, 58]. Both statistics convey progress of the Office of the National Coordinator’s (ONC) full EHR adoption goals incentivized through Medicare and Medicaid payouts in 2009. Within healthcare organizations, these records are most often governed by role-based access control (RBAC) schemes specific to the organization [18]. These RBAC schemes base access policy on the roles that individual users have as part of their healthcare organization—for example, outpatient physician, inpatient nurse, hospital administrator, biller, physicians’ assistant, etc. Specific access rights to a particular record are granted to a role name, thus allowing access only to those individuals associated with that particular role [18, 23, 44]. An individual’s membership in roles can be revoked or added upon, commonly by hospital administrators, which helps to simplify the management of permissions by eliminating the need to update such permissions for every individual user. Also, RBAC schemes help satisfy the least privilege access protocol in which only the minimum set of access privileges required for the provider to perform their role’s job are granted [17].

However, despite the prevalence of RBAC schemes for EHR access policy amongst healthcare providers, several concerns have been raised regarding patient privacy and control. First, as mandated by the 1996 HIPAA privacy rule, patients have the right to control who can access their information and how that information is managed [40]. However, in reality, patients sign waivers during registration and have little transparency into how their records are actually being used—each organization has their own specific access policies and concerned patients can only make requests for internal audits, a very lengthy process. This is especially problematic considering that patients themselves are often the only ones to recognize improper access; most such cases are perpetrated by a known rel-
ative/friend/acquaintance within the organization [23, 24, 29]. Thus, patients would be the most qualified auditor. It is worth noting that there are also opposing concerns that the sheer number of accessors would alarm patients with full visibility over the access logs of their medical records [23, 25, 29]. Furthermore, each organization holds its own version of a patient’s health record; sharing records outside of an organization poses a couple key challenges: 1) printed/electronic text copies can be sent, yet access cannot be monitored once doing so; 2) this process is relatively time and labor intensive; 3) differing data standards and EHR software usage are huge barriers to interoperability amongst existing cross-institutional EHR systems [2, 25, 41, 42]. Some national data standards, namely HL7 [5], have been attempting to improve this cross-institutional EHR interoperability challenge, but as evident by the above-mentioned challenges, that is just one piece of the puzzle.

As work continues to improve EHRs for healthcare delivery, a general consensus on key requirements has emerged. In terms of access, Alhabani and Fidge consolidate these requirements around patients’ ability to control granting or blocking access from certain providers, healthcare organizations’ ability to define access policies within their own domain, and providers’ ability to access information when they need it or in the case of emergency [17]. In terms of interoperability, the ONC has emphasized through its Shared Nationwide Interoperability Roadmap goals of full interoperability amongst healthcare providers, accurate and holistic patient data, consistent data formats, and secure data transport and delivery [32, 43]. Blockchain technology holds potential to address these requirements by leveraging its immutable audit trails, smart contract automation, design flexibility, decentralized data structures, and cryptographic security. Researchers have begun exploring blockchain-backed EHR systems, which, along with other efforts at patient-
owned, interoperable EHR solutions, will be outlined in the next section.

3 Related Work

As electronic health records have gained prominence in the healthcare landscape, efforts to harness technology for more holistic, patient-owned, interoperable medical records have also emerged. Within already existing EHR infrastructure—in 2016, Epic Systems was in particular reported to be used by 26% of medium to large sized clinics, yet this statistic is likely much higher now [58]—there is ongoing platform-specific work being done to standardize data, engage patients via online portals (Epic offers a portal called MyChart), and share patient records across participating institutions [22, 23, 24]. For example, Epic is enabling two separate hospitals, both using Epic for their EHR management, to share medical data across the Epic platform. However, these efforts, as a product of their top-down approach, do not solve problems of record fragmentation across institutions, lack of patient ownership over their records, or varied software implementations (systems like Epic are still highly customizable) [23]. Consequently, external, bottom-up approaches of EHR/PHR systems have also garnered attention, and these systems are the basis of my work.

Prior to blockchain’s introduction, tech giants Google and Microsoft entered the healthcare sphere with Google Health and Microsoft HealthVault. Both were marketed as PHRs where users could collect and store their medical information on one centralized account. However, in both cases, users entirely controlled access to their health records and could only share them with other platform users [25, 48, 49, 61]. Consequently, without
the engagement of healthcare providers, the main creators and curators of patient medical records, user adoption of these PHR systems remained low—Google Health was discontinued in 2011 as a result. Sunyaev highlighted the functional conflict of patients’ privacy and providers’ data accessibility by assessing the patient-centrism of Google Health and Microsoft HealthVault [61]. Additionally, Spil and Klein’s user study [9] concluded that patients viewed the primary use of Google Health and HealthVault, creating their personal health record, to be time consuming and unnecessary; they would rather rely on the records created by their providers. Users also did not trust the privacy of their records on the internet, leading to the authors’ assertion that “low trust and high risk emerge as two additional significant reasons for the failure of personal health records” [9].

Ekonomou et. al. [2] bridged the gap of PHRs and EHRs by combining the Microsoft HealthVault system with the DACAR EHR system on a cloud-based infrastructure. Their solution uses role-based access based on DACAR’s authentication system and a “translation gateway” between DACAR and HealthVault, yet does not allow the patient to specifically grant, block, or monitor access once enabling the translation gateway with his HealthVault account. Their work demonstrates the need to “share health data, user identities, and access rights across health domains” [2].

Chen et. al. [37] proposed a cloud-based privacy-aware role based access control (CPRBAC) model. This scheme allows for more granular controllability, traceability of data, and authorized access to cloud resources, where the underlying assumption is that the PHR/EHR scheme is cloud-based.

Ekblaw et. al. [7] utilize blockchain technology to create MedRec, a decentralized medical record management system allowing patients to access and control their records
across varied providers. Built upon the permissionless Ethereum platform, the system uses a series of smart contracts to manage EHRs; registrar contracts (RCs) map user identification strings to their Ethereum address, patient-provider relationship contracts (PPRs) allow patients to define very specific rules for which providers can access which parts of their data, and summary contracts (SCs) unite all pieces of the patient’s medical record in one piece by storing references to the off-chain medical files and references to all of the patient’s PPRs. However, the MedRec system does not outline any auditing capabilities and provides patients with extremely fine-grained control of their records. Any provider who accesses a patient’s record must be given explicit access by the patient with the creation of a PPR, and the patient can fine-tune access down to singular queries. This approach is no doubt patient centric, but also lacks flexibility and practicality.

Yue et. al. developed Healthcare Data Gateways which uses a purpose-centric access control model for accessing health data stored on the blockchain. $R$-users are defined as users seeking access to raw healthcare data while $p$-users denote users who wish to retrieve results from data achieved through secure multi-party computation. The system does not address practical healthcare delivery workflow or patient-specific access control as it takes a more research-oriented focus.

Xia et. al. [11] introduce the usage of a permissioned blockchain for sharing medical records yet again take a research-oriented focus. Their system relies on permitting data users and owners to access EHRs from a shared repository after their identities and cryptographic keys have been verified, yet no access rules exist once entry is permitted into the closed blockchain. The system does make use of blockchain’s immutability and autonomy by recording all requests and accesses of data as transaction on the blockchain.
Also, their focus on integrating identity management with the blockchain, an aspect that is outside the scope of my work, offers a promising solution to be included with future EHR blockchain solutions. Their later work, MedShare [14], further emphasizes the role of smart contracts to monitor data usage and automatically revoke access if data permissions or rules are violated. This approach adopts transparency of data usage, yet does not specifically address patient control or healthcare delivery workflow.

Dubrovitskaya et. al. [31] also utilize a permissioned blockchain with Hyperledger to manage access to cloud-stored medical records. However, the system does not go into any detail on access policies, and the framework is presented as being institution-specific. That is, the network only exists of trusted peers within the same institution, which therefore does not address any interoperability concerns. Rather, their focus was on allowing patients to impose very fine-grained access control policy and efficient data sharing, and then applying this work towards a Radiation Oncology clinic.

My work builds upon the above-mentioned systems by assuming a cloud-based EHR/PHR system, employing a role-based access scheme, addressing the balance between patient privacy and provider accessibility, and utilizing a permissioned blockchain with smart contracts for immutability and automation. With the underlying assumption from prior work that we are moving towards a patient-centric, decentralized EHR/PHR system, I directly address the challenge of increasing patient control while maintaining flexible provider access for healthcare delivery via blockchain technology.
4 Survey Investigation

My work approaches the issue of patient-centric access control in two manners, first by conducting a user survey and second by developing a blockchain network prototype to address the spectrum of patient control preferences and automate practical, existing access policy for providers on a distributed scale. This chapter will summarize the goals and results of the survey investigation and the following chapter will walk through my blockchain prototype solution.

With the many cases of recent literature advocating for patient-centric access policies, a natural question arises: do patients really want this level of control? Do they fear for their privacy and even understand the ways in which their medical information is currently being accessed? To investigate the plausibility of patient-centric access policy from the patient perspective, I conducted a user survey to determine:

1) the level of control most patients want over the accessibility of their health records
2) the level of understanding most patients have over who must view their health records
3) the nature of patients’ concerns/desires over how personal health records are shared
4) if patients comprehend and have interest in an application allowing them to grant, block, and monitor access to their records based on currently used RBAC schemes

The survey begins with a demographics section, continues with a section evaluating patient familiarity with EHRs and their perceptions of privacy, then walks through mock-ups of a blockchain-backed application to determine patients’ understanding of and interest in using this access control solution. After first receiving approval from the Commit-
tte for the Protection of Human Subjects (CPHS) at Dartmouth College, I conducted this survey on the Amazon Mechanical Turk platform and restricted it to individuals within the United States only. This sample, as a result of being executed through the Amazon Mechanical Turk platform, is certainly biased. It is neither a representative sample of the nation nor of people who are likely to be concerned with their EHRs or their PHRs. On the other hand, to the extent that Mechanical Turk participants are more focused on technology than the average patient, and also to the extent that they have greater ability to understand the use of digital data than the average patient, this sample does provide useful insights into how a subset of the population thinks about control of their medical data. For the purposes of my research, this sample is useful in viewing how we might anticipate concerns about patient privacy and allocation of permission to view the data, especially as future adult populations become more and more techno-savvy/familiar with digital data.

A copy of my survey can be found in the Appendix.

4.1 Demographics

The survey was completed by sixty individuals on the Mechanical Turk platform. As mentioned above, Amazon Mechanical Turk users exhibit biases, thus a full randomized survey of patients around the nation would be a valuable comparison and goal of future work. Of the sixty individuals, there is a heavy skew towards younger people. However, these younger people are those who stand to benefit from future improvements in EHRs. The exact distribution of age is shown in Figure 2a below. Gender distribution is relatively even yet skewed towards males, and the exact distribution is shown in Figure 2b below.
Most respondents have higher degrees of education, shown in Figure 2c below, and only see a medical professional 1-5 times a year.

While this demographic is clearly not representative of the entire American population, they do represent a significant subgroup of younger, educated individuals who have grown up in the Internet age, are familiar with privacy concerns, and have relatively infrequent healthcare issues. This survey was only intended to help inform the blockchain prototype and understand if patient-centric access control is a universal desire amongst the public.

4.2 Results

I will address notable results according to the four above-mentioned main goals of the survey investigation. First, regarding the level of control most patients want over the accessibility of their health records, a few key lessons were learned:

1) When provided an explanation of patient-centric proposals and prompted if they want to be responsible for granting each individual provider explicit access to their records, 55% of respondents preferred medical professionals to handle the management of access policies.

2) All but 5% of survey-takers want to be notified when someone accesses their health records, yet half are comfortable if they are only notified if someone not explicitly permitted by them accesses their records. This could be any medical professional from nurse to insurance biller. As previously mentioned, this could be problematic if the sheer number of accesses alarm patients, yet, within my scheme, this notification would
What is your age?

(a) Breakdown of survey respondents’ ages. The population skews strongly on the younger side, however this population would age into the primary beneficiaries of healthcare in the next 20-30 years.

What is your gender?

(b) Breakdown of survey respondents’ gender. The population skews male but not drastically so.

What is the highest degree that you hold?

(c) Breakdown of survey respondents’ levels of education. The population tends to be have higher levels of education.

How often do you see some form of medical provider (general practitioner, dentist, counselor, etc.)?

(d) Frequency of seeing a provider distribution amongst survey respondents. The sample, likely as a result of its youth, skews towards infrequent healthcare encounters.

Figure 2: This figure shows a series of demographic breakdowns of my survey respondents. Unfortunately, this sample, as a result of being executed through the Amazon Mechanical Turk platform, is certainly biased. It is neither a representative sample of the nation nor of people who are likely to be concerned with their EHRs or their PHRs. On the other hand, to the extent that Mechanical Turk participants are more focused on technology than the average patient, and also to the extent that they have greater ability to understand the use of digital data than the average patient, this sample does provide useful insights into how a subset of the population thinks about control of their medical data. For the purposes of this research, this sample is useful in viewing how we might anticipate concerns about patient privacy and allocation of permission to view the data, especially as future adult populations become more and more techno-savvy and familiar with digital data.
only be from a new role, i.e. outpatient nurses as Hospital X or surgery claims processors at Insurance Company Y—this scheme is discussed in detail in Chapter 5. This is useful insight indicating that patients do want to know who is viewing their information (something that the immutable history of blockchain transactions can accomplish well), and that notifications should still exist in some capacity.

3) When asked whether they would be more lenient about the access policies for their medical records if they could see exactly when and who accessed specific parts of their records, 0% answered “no”, 16% answered “not sure”, 45% answered “maybe”, and 38% answered “yes” as shown in Figure 3 below. Thus, transparency helps promote flexibility in theory. It would be a valuable sociological experiment to observe how patients would actually behave in reality. For my research, I assume this transparency would help achieve flexibility, although this may be contested.

4) When asked about the types of tasks they would like to have control over, 81% would like to be able to specifically grant access to an individual provider (such as physician, a specific insurer, etc.), 62% would like to be able to specifically block access from an individual provider, and 88% would like to be able to view exactly who has accessed their records. Only 23% and 14% of respondents would like to be able to add and delete from their own health records specifically. This seems to support the lessons learned from Google Health and Microsoft HealthVault—patients prefer to leave the curation of their medical records to healthcare providers.

Next, regarding the level of understanding most patients have over who must view their health records, another set of notable results stand out:
If you could see exactly when and who accessed specific parts of your medical records, would you be more lenient about the access policies for your medical records?

Figure 3: This figure depicts the breakdown of responses indicating patient leniency in access policies if access events are transparent. It must be noted that there may be a significant difference here in what patients say they would do versus what they actually would do in a real situation. For the purposes of my research, I assume transparency would help promote leniency, especially with my nudge theory based access scheme described in Chapter 5.5. However, this may be contested.

1) 78% of respondents understand that healthcare professionals other than their main physician, such as nurses and insurance billers, must view their records, yet only 15% believe that more than 5 individuals may view their records on any given healthcare visit. There is clearly of lack of understanding amongst patients regarding how their medical records are used by providers. This does suggest that relying on patients to grant access entirely would be quite problematic.

2) About 75% of respondents believe that fewer than 10 individuals actually have access to their records regardless of whether or not they actively look at them. In reality, role groups at large healthcare organizations can be very expansive, and theoretically hundreds of providers could have access to any given health record.

Regarding the nature of patients’ concerns over how personal health records are shared, this survey also yielded important results:
1) When asked how concerned they are about who is accessing their medical records, 38% of respondents are not at all concerned, 33% are a little concerned, and only 12% are very concerned. These results are shown in Figure 4 below. Perhaps the wariness of patients, and therefore the incentives for patient-centric models, seem to be much weaker than prior literature portrays, at least in this sample.

2) When asked about the nature of their concerns, the most indicated fears are for someone exploiting their information for profit (38%) and someone judging their personal information (31%).

3) When asked if they are actively concerned about the number of people who may potentially have access to their records, just 26% responded affirmatively. The full breakdown is shown in Figure 5 below.

4) Only 9% say that they value the privacy of their records over the quality of their care. If this is a representative sentiment, then this largely supports my model of access control, discussed in Chapter 5.

Finally, regarding whether or not patients comprehend and have interest in an application allowing them to grant, block, and monitor access to their records based on currently used RBAC schemes, results were overall positive. This section of the survey, included in the Appendix, walks patients through an access scenario where the underlying rules are governed by my RBAC scheme, described in detail in Chapter 5.5. Specifically results show that:

1) Over 60% of respondents understand the RBAC scheme correctly from the application
How concerned are you about who is accessing your EHR medical records?

Figure 4: This figure depicts the breakdown of respondents’ level of concern over who can access their medical records. Notably, less than 12% of sample respondents report being very concerned about who is accessing their records. This result is a large basis for my decision to default to an organizational role based access scheme in my blockchain network prototype, discussed in Chapter 5.5. If patients are not heavily concerned, perhaps we will observe a more optimal outcome in terms of flexible access rules by giving patients the choice to have more control, yet defaulting to a more flexible access scheme. Thus, the majority of unconcerned patients would not alter their access rules.

Are you concerned by the number of people who may have access to your records?

Figure 5: This figure depicts the breakdown of respondents’ level of concern over the amount of people who may have access to their medical records. Notably, most patients do not understand the number of individual providers who actually access their records, yet still, about 40% of respondents are concerned about this. This suggests that it could be worrisome for patients to have transparency into who is viewing their records—the sheer number of accesses may alarm them. However, since the majority seem not to be concern, and because of the potential to attach rationale for accesses upon further development of my solution, I decide to include transparency in my blockchain system by allowing patients to view the immutable history of access requests to their records.
walk-through and indicate that they are comfortable with this kind of access policy.

2) When asked if this application system seems more secure, just as secure, or less secure than what they believe about how their records are currently accessed by healthcare organizations, 95% of respondents chose equally or more secure. The RBAC scheme that I implement and convey in the survey is, in reality, equal to those currently used by healthcare organizations, yet I add transparency and enhance patient control by allowing users to specifically grant or deny access to individuals.

3) 40% of respondents said they were very likely and 33% of respondents said they were extremely likely to use an application like this. Only 6% of respondents said they were unlikely to use an application like this. Also, by way of blockchain’s distributed maintenance and automation, my system would not require patients to be actively using this application if they had no interest in monitoring access to their health records.

4) 10 individuals left additional comments indicating their interest in the concept and their desire to have more control and transparency over who accesses their records. Concerns raised by some respondents in the comments include emergency situations, real-life access scenarios where doctors share passwords, and poor security allowing rules to be broken. While my system does not address how providers would authenticate themselves as their correct participant identity in the network, the distributed consensus and immutable history of blockchain would ensure that programmed rules would be securely enforced.
4.3 Survey Conclusions

The above results have informed my conclusions regarding the four main goals of the survey. First, in line with what we have learned from the commercial failures of Google Health and Microsoft HealthVault, patients are less interested in contributing to their records and more interested in having additional control and transparency over who is accessing them. These results support Spil and Klein’s conclusions that patients prefer their records to be curated by their providers, yet still have privacy concerns [9]. Also, the exact level of control patients desire is quite varied—flexibility is essential. One thing is clear: very few patients are highly concerned about who is accessing their medical records. Most patients do not want to be the ones primarily responsible for managing access to their records, but they do want to be able to grant, block, and view accesses at their will. Furthermore, while most patients understand that many different kinds of healthcare providers may access their records, most do not understand that, as a result of organization-wide RBAC access schemes, several providers actually do have access yet do not look based on HIPAA protocols. Greater transparency over who is accessing their records would make patients more comfortable with flexible access rules. Finally, most respondents feel safer with and are interested in an access control application that allows them to monitor access to their medical records and have greater control over granting/blocking access. However, there is not full understanding among respondents of who has access—some still believe they are granting access to each provider individually rather than by role—so the application would have to include informative pop-ups to address some patients’ lack of RBAC understanding.
5 Blockchain-backed access control system prototype

My work also investigates the balance of patient control and rule flexibility by developing a blockchain network prototype. This prototype aims to address the spectrum of patient control preferences shown in the user survey and automate practical, existing access policy for providers on a distributed scale. This section will walk through the implementation of the prototype by aligning blockchain’s features with EHR access problems, outlining the technologies used, explaining the access model, smart contracts, and user capabilities, and finally commenting on issues during development. For my solution, I implemented a role-based access control (RBAC) scheme (discussed in Chapter 5.5), a blockchain ledger modeled as access states and transactions modeled as access control events (discussed in Chapter 5.6), smart contracts to enforce access rules (discussed in Chapter 5.7), and added some additional programmability (discussed in Chapter 5.8).

5.1 Blockchain features and the EHR access problem

Blockchain has garnered much attention in recent years for its potential to revolutionize a variety of industries and workflows. This is largely because of its universal distributed ledger, immutable transaction history, cryptographically secure enforcement of programmed rules, and overall ability to facilitate behavior amongst participants who do not fully trust one another. After investigating blockchain’s features and potential applications, it became clear to me that blockchain offers a potential solution for problems involving 1) many participants with variable roles/needs, 2) a need for secure, trusted logs of past behavior, and 3) the ability to apply universal, automatable rules. While, as dis-
cussed in Chapter 3, blockchain is being investigated for EHR systems as a whole, I found blockchain’s features to be particularly applicable to managing and controlling access to EHR files stored off chain. As shown by prior work and my own survey investigation, patients desire greater control and transparency over who is accessing their medical records. However, patients’ levels of concern regarding improper accesses to their records spread a wide range, and very few are strongly concerned about how providers are accessing their information. Patients indicate, and are lawfully granted via HIPAA, a desire to know who is accessing their medical records [40]. Thus, I believe that blockchain could mediate trust and provide a secure history of such access events. Also, the healthcare sphere involves many varied participants—organizations, insurers, physicians, nurses, patients, etc.—and these participants can be grouped according to their varying roles and needs. We also know from conversations with providers that healthcare delivery depends on easy access to necessary information, and current healthcare workflows require several providers to have access to a single patients’ medical records. Thus, there is potential for automating access rules based on healthcare workflows. I do this by modeling rules after those currently used by many healthcare organizations, explained in detail in Chapter 5.5. Additionally, with the support of the U.S. Government’s Office of the National Coordinator, there is a tremendous push towards having universal, holistic patient records and full interoperability amongst providers [32, 43]. This provides grounds for the assumption that my blockchain-backed access model could be applied to a universal, interoperable EHR system.

Blockchain’s universal distributed ledger directly aligns with the push towards universal, holistic, patient-owned medical records. In this way, I assume each medical file could be stored on a cloud service and then I adapt the distributed ledger to store the
current access rules associated with that file. Additionally, the secure, programmable enforcement of rules allows for current healthcare access protocols to easily be adapted into a blockchain network. I write smart contracts, stored on each node of the blockchain network, that enforce my RBAC model based off of these current protocols (see Chapter 5.5). Also, the immutable transaction history ensures that any time access is requested, it is recorded permanently in the blockchain. As a result, patients can view accesses to their records—this offers added transparency desired by patients.

The automated, self-enforcing flow of the network allows patients to be as greatly or as weakly involved as they want. Thus, by utilizing blockchain’s automation with smart contracts, and adding in additional access “transactions” available to patients (blocking access, granting access, viewing access), I address the spectrum of patient concerns regarding improper access to their records and offer them greater control if they desire it. I discuss a primary behavioral theory behind this idea, nudge theory, more in Section 5.5. Finally, the cryptographic authentication of participants to a permissioned blockchain network ensures that participants are known and acting within their respective realms of control. I model participants based on general roles in this network, and assign capabilities within the network based on those roles. Despite not fully trusting each other, patients and providers can exchange access to health information and easily identify when access has been abused. My use of blockchain thus allows patients to take greater ownership over who accesses their medical records and keeps providers honest about only accessing necessary information. In my system, access rules are, by default, the same as in current healthcare organizations, yet patients gain added transparency with the immutable history and the ability to slightly alter those rules by submitting access “transactions”. As
a result of the ability to apply these features to EHR access control, blockchain stands as a strong candidate for helping address the modern EHR access concerns of both patients and providers.

5.2 Assumptions

In line with prior work and current literature, my work assumes a universal, holistic patient medical record system. Specifically, this means that there is only one version of each medical record file that is accessed and curated by many healthcare organizations, rather than each organization storing their own versions of patient EHR files internally. Patients own their individual files of medical history, shown in Figure 6, and access rights can differ amongst the separate files. For example, patients’ mental health history may be separated from their general care history and dental history. These files are encrypted and stored on a cloud service, as seen in [2], and referenced by a unique record identifier. These files are created and curated by medical providers rather than the patients themselves. The goal of my blockchain network is to maintain flexibility by automating RBAC-based access rules for medical providers to these files and to offer patients greater control by allowing them to monitor, explicitly grant, and explicitly block access to any of their files. This scheme of assuming universal, patient-owned, cloud-stored EHR files is also used in several examples of prior work, notably [2, 7, 8, 11].

My solution focuses only on implementing a flexible access control blockchain network; it does not address securely retrieving medical record files once access is approved. However, there are several prior examples of securely retrieving and decrypting files stored
Figure 6: I assume this scheme of universal medical record storage in my work. Separate medical records each belong to a single patient, are encrypted and stored on a cloud service. This is the same scheme used in prior literature, specifically [2, 7, 8, 11]. My work does not address actual retrieval of these records, however [2] addresses this in depth.

on the cloud. For example, Ekonomou et. al propose a token-based access system to their cloud-stored EHR files [2]. For time and scope purposes, I do not address this concern and leave the fetching and decryption of medical record files to future work. Furthermore, while identities in my network are assigned a public-private key pair and must authenticate within the network, I do not address how individual providers authenticate themselves within their organizations. Thus, my work assumes that participants are assigned the proper roles from their organizations and tests functionality based on this assumption. The tasks of properly assigning participants roles and authenticating them based on those roles are therefore also outside the scope of my work and remain subjects for future work. Finally, for the sake of a prototype, I design my RBAC scheme based off role structures most commonly used by large healthcare organizations [18, 22, 23, 24, 44], yet I acknowledge that several other access schemes may be used in practice. The RBAC scheme used is explained in further detail in Chapter 5.5, and the ability for organizations to further
customize their access protocols is addressed in Chapter 8 as future work.

5.3 Requirements

In specifically addressing the issue of access control for a distributed, universal EHR/PHR system, I outline a few key requirements for a solution. First, the solution must be flexible. It must be able to easily adapt to the changing roles of medical providers and changing relationships of patients and their care providers. Second, the solution must offer patients greater control over who may access their medical records. This falls in line with the push towards patient-centrism, as patients should be easily able to identify who has viewed their records and adjust access accordingly. However, they should not be relied upon to grant explicit access to every provider who may require it. Third, the solution must be easily adopted and customizable. Organizations have several different kinds of role structures and implement different access schemes. This customization should be easy for individual organizations to replicate in my EHR access control network, and the network should be able to be incorporated to their existing workflows relatively smoothly. Fourth, the solution must be trusted and secure. Patients should be certain that no access event goes unreported, that the reported access state of their records is accurate, and that their access rules are being properly followed. Finally, the solution must be practical. Healthcare requires timely and efficient spreading of information to maximize quality of care. Therefore, an EHR access control network must allow providers to learn what they need to know when they need to know it. These five requirements guide my technological and structural choices for implementation.
5.4 Technologies Used

I build my solution upon the Hyperledger framework introduced in Chapter 2, and specifically use the Hyperledger Composer architecture to implement an access control blockchain network [62]. Hyperledger is a project of open source blockchains and related tools, started in December 2015 by the Linux Foundation, to support the collaborative development of blockchain-based distributed ledgers [51]. The Hyperledger Fabric is a permissioned blockchain architecture that I chose for flexibility and performance reasons. Specifically, unlike a permissionless network, participants in an EHR access control network cannot be anonymous—identities are essential to the solution. Furthermore, a permissioned network has no systemic dependency on a native cryptocurrency or need for slow and costly proof-of-work to verify transactions as seen in Bitcoin [53]. Without the need for a cryptocurrency backbone, the Hyperledger Fabric offers a flexible platform on which to build a blockchain network. Also, without the need for proof-of-work or proof-of-stake consensus, Hyperledger can execute transactions much more efficiently than permissionless networks. Finally, Hyperledger Composer, with its own command line interface (CLI), data modeling language, and network testing platform, made it a very favorable platform for developing and iterating a flexible, novel blockchain network [62]. Other options included permissionless platforms, most notably Ethereum, which, as explained above, are not very plausible for the EHR access problem due to their anonymous participants and cryptocurrency dependency [6]. In terms of permissioned platforms, I ruled out other options, notably Multichain and HydraChain, due to the lack of development activity and documentation. Hyperledger is by far the most active in terms of open source
development (as measured by its GitHub statistics such as stars and forks). Thus, Hyperledger’s support, documentation, growing familiarity, and easy adoption by developers made it appealing to work with. Overall, Hyperledger stood out to me as the best option for developing a blockchain network with the flexibility, security, and practicality required of an EHR access control system. This is especially true considering how I sought to remodel the ledger as storage for access states and transactions as access events—these tasks required frequent testing, remodeling, and flexible development options.

To further improve flexibility and achieve the customizability, easy adoption, and increased patient control required of my solution, I specifically built upon the Hyperledger Composer framework [62]. Hyperledger Composer was first introduced in June of 2017 and is a set of collaboration tools for building blockchain business networks built with JavaScript. The architecture helps simplify the development of a proof of concept blockchain business network by leveraging modern tools including node.js, npm, a command line interface (CLI), and plug-ins with popular editors. Hyperledger Composer offered a modern and efficient platform on which to develop and test my EHR access control network with a variety of participants and functionalities. Furthermore, Hyperledger Composer is a constant work in progress—new features are being added monthly—and it is targeted specifically towards business developers. This makes it an excellent platform to use in addressing the required customization of different healthcare organizations. The framework allows developers to quickly test their networks and create decentralized applications; this could be extended to internal development teams and allow them to specify access policies for their own unique organizations. Thus, the Hyperledger Composer framework offers further flexibility, customizability, and workflow adaptability. At the time of development, I am using
Hyperledger Composer version 0.16.0. As an open source work in progress, Composer did have some compatibility bugs I encountered which are further discussed in issues. As of January 2018, Composer does not supposed Node version 9; I am using Node version 8.9.1, npm version 5.6.0, and nvm version 0.33.0.

5.5 Access Model

The access model I use in this solution is based off of the RBAC model used most commonly in large healthcare organizations [18, 22, 23, 24, 44]. However, as a result of the development structure and technologies used, roles can very easily be changed and customized. I evaluate the functionality of this model in Chapter 6. It can best be described as a hierarchical role structure, where the largest participant is an organization, then the specific role groups within those organizations, then down to the individual participants within those role groups. Roles are based on organization, job function, and assignment within the organization. For example, Hospital X is a large organization with several employees grouped into roles: Outpatient Physician, Inpatient Physician, Medical Student, Inpatient Nurse, Outpatient Nurse, Physician’s Assistant, Insurance Biller, and Hospital Administrator. Individuals can assume more than one role, i.e. Dr. Jane Doe may be an inpatient physician and a hospital administrator and thus assume the access rights of both roles, however she must be explicitly added to those role groups by the administrator of her organization. By default, when a provider requires access, the patient must grant initial access to that provider’s role group. Figure 7 shows a mock-up example of this used in my survey investigation. However, once that access is granted, access is extended to all
other providers at that organization within the same role. To illustrate, if Sarah, a patient at Hospital X, goes to see her Outpatient Physician Dr. Johnson, then she must grant Dr. Johnson access to her general medical history record. After that initial granting, all other outpatient physicians at Hospital X have access to Sarah’s general history record. If Dr. Smith, Dr. Johnson’s colleague and also an outpatient physician at Hospital X, is consulted for Sarah’s case and requests access to her records, then she should be allowed access as long as Sarah has not explicitly blocked her. Further details on the smart contract logic used are discussed in Chapter 5.7 below, and the default flow of access rights in this solution is shown in Figure 8. Overall, whenever a provider with a different role is requesting access to a certain record file, the patient who owns that record must be the one to grant that access. However, once that initial access is granted, access is automatically extended to all others in that same role. Additionally, the Hyperledger Composer platform makes it very easy to encode additional rules to customize an organization’s access protocols, i.e. automatically extending access to active inpatient nurses when access is granted to inpatient physicians.

One caution that I am wary of is organizations’ ability to structure roles that are too broad. Since access is automatically extended to all individuals within a role group, an organization could easily assign all providers to a single role. However, I then look practically at the healthcare industry—it is highly regulated and an essential part of American government services. It seems to me highly unlikely that this relationship will ever fade, as the government’s primary role is the welfare of American citizens. I thus assume that the proper regulation and accountability would exist to ensure that organizations are following reasonable role structures based on responsibilities. Furthermore, this could be no worse
than current access protocols, which are internally devised by organizations and currently face the same kind of government regulation. Thus, the goal of my work is to suggest a practical access system for universal, distributed patient records and I therefore assume healthcare organizations would structure their roles accordingly. However, it is certainly worth noting the potential hazard of improper role assignment.

Finally, I would like to discuss a behavioral motivation for the role structure I implemented, and specifically how blockchain helps me to achieve this. Nobel Prize-winning Behavioral Economist Richard Thaler made nudge theory popular in his 2008 book by the same title, *Nudge*. The idea is simple; in the words of Thaler, “a nudge is any aspect of the choice architecture that alters people’s behavior in a predictable way without forbidding any options or significantly changing their incentives. To count as a mere nudge, the intervention must be easy and cheap to avoid. Nudges are not mandates. Putting fruit at eye level counts as a nudge. Banning junk food does not” [59]. The classic example of a nudge policy is an opt-out organ donor program—it is socially optimal for more citizens to be organ donors, and this result is achieved by making the default option to be an organ donor, yet people are easily able to opt-out by choosing not to be. In the case of EHR access policy, it is socially optimal from a provider’s standpoint to have flexible, easy access to patient records. Thus, I choose to make my RBAC scheme, where the entire role group obtains access automatically if any individual member of the group is granted access, the default rule of this blockchain network. However, I easily allow for the “opt-out” by adding transactions, available to any Patient participant, that grant explicit access and block specific access from individual providers. This allows patients who are most concerned about their privacy or who have a specific desire to add in more granular access
rules to have that ability, yet, in line with nudge theory, the default option should lead to a more socially optimal outcome—patients feel as though they have the choice, but likely will not go the extra step to create more granular access rules unless they have stronger desires to do so. This design choice should, in theory, help maintain flexibility and practical access policy while still providing patients the control they desire and legally own.

Blockchain specifically helps achieve this with smart contract automation; by remodeling the ledger to store access control states, further discussed in Chapter 5.6 below, and modeling access decisions and events as transactions, I am able to securely, programmatically enforce these default rules and fuse them with additional rules specified by patient transactions. Every access decision and event is permanently stored in the blockchain history, and participants must be cryptographically authenticated to the network. Thus, while any system could enforce such rules, blockchain adds an extra level of security and trust. The absolute only way to interact with the ledger is by going through the smart contracts, or chaincode in Hyperledger, and thus any attempt to change the access rules or request access will be made visible to the file owner. Furthermore, since blockchain operates as a distributed data system—it is not stored in a single location but rather spread across all nodes of the network—all participants can agree upon and trust the behavior exhibited in the network. Hyperledger employs a Practical Byzantine Fault Tolerance (PBFT) consensus mechanism to verify and publish transactions to the blockchain [56]. The chaincode on each participating node processes each transaction and those outputs must be verified across the network for the transaction to be executed and stored on chain. Thus, rather than have vulnerable, expensive, inefficient storage of EHR files and access rules performed separately by individual organizations, blockchain allows for secure, trusted,
Figure 7: An illustration of a patient granting default access to providers—this image was used in my user survey investigation. Of course, upon further development and actual user testing, the options available and user interface may change, yet the idea is to default to my simple RBAC scheme, where access is granted to the provider’s entire role group, and patients can make this access more granular if they choose. immutable storage and enforcement of access rules, where the provenance of such rules is entirely clear.

5.6 Implementation Structure

This subsection will outline the structure of my blockchain EHR access network solution. To reiterate, in my solution, I implemented an RBAC scheme (discussed in Chapter 5.5 above), a distributed ledger modeled as the current access state and transactions to interact with this ledger (discussed in this section), smart contracts to enforce access rules by handling the transactions (discussed in Chapter 5.7), and added some additional programmability (discussed in Chapter 5.8). The blockchain network itself consists of four main parts: the distributed ledger, assets, transactions, and participants. The distributed
Figure 8: This figure shows an example of my system’s default role-based access flow. Jane must be the one to grant initial access to each organization-role, or RoleOrg participant in my blockchain network. Once Jane grants this initial access, the entire role group gains access to her record. The downside of this scheme is that Jane is still responsible for these initial grantings (steps 1, 4), yet this does not differ than current organizational standards of signing HIPAA forms when arriving for treatment to ensure that her medical data can be shared amongst the organization. Additionally, healthcare organizations are easily able to customize their access schemes in my system; Hospital X could have a rule that groups Outpatient Physicians and Outpatient Nurses. In this example, Sam, Helen, Sarah, James, Judy, and Bill gain access to Jane’s record file with Jane only granting two separate access requests. Also notably, Jane can see exactly who has accessed her file and when they have accessed it with blockchain’s immutable history—this promotes an added level of trust.
ledger stores the current state of the network. In Bitcoin, the ledger stores each account number and the number of bitcoins, or assets, associated with those accounts [30]. Transactions allow for the assets to move from one account to the other. So, for example, assume the current state of the Bitcoin network has three accounts: Alex, Bob, and Casey. The ledger says that Alex has 10 coins, Bob has 5 coins, and Casey has 3 coins. Bob needs to pay Casey 1 coin for dinner, so he initiates a transaction to transfer the 1 coin to Casey. After the network approves the transaction, the ledger indicates the new state of the network: Alex has 10 coins, Bob has 4 coins, and Casey has 4 coins. In the case of this EHR access network, the ledger represents the current state of access rules. To represent access rules on the ledger, I decided to create an Access asset—this is the equivalent of bitcoin in the previous example. Thus, in this system, the ledger is just a key-value store of each Access asset. Access can be granted or denied through specific transactions executed by logic written in smart contracts, which in turn update the data stored in each Access asset. Consequently, the only way to interact with the ledger—that is, change the access state—is to submit a transaction that is processed by the chaincode (smart contracts).

The structure of the Access asset is shown in Figure 10 and an example of an Access asset stored on the ledger is shown in Figure 11. Each individual record file, identified by the file ID, corresponds to an Access asset. The asset contains a list of entities that have been granted access to the file, a list of entities that have been blocked from accessing the file, an integer that counts the number of accesses to the file, and a reference to the network participant, a patient, that owns the file. With these attributes, the network ensures that any change to access rules and any access request are recorded permanently in the chain—this is the rationale for including the accessCount attribute. If there is a successful
access request, this must change the ledger by upping the *accessCount*, thus updating the entire network of the event. The same is true for changing access policy, either within the *accessList* or the *blockedList*. As of now, these lists can contain any form of *ProviderEntity*, discussed further below, which could be an entire role group or just a single individual provider.

It is useful to conceptualize this *Access* asset in terms of traditional system security frameworks. For an access control matrix [60], there are two important objects to distinguish: the *Access* asset itself, stored in the distributed blockchain ledger, and the actual EHR file, stored off-chain in the cloud. Figure 9 distinguishes which participants can do what to which objects in my system. As mentioned, the two main objects are the *Access* asset stored on chain, and the EHR record file stored off-chain. The data contained in the *Access* asset defines the ability of providers to read the EHR record file.

Transactions initiated by network participants allow for the state of the access rules to change. This network currently has five different kinds of transactions shown in Figure 12: assigning a role to a provider, removing a role from a provider, granting access, requesting access, and blocking access. The network also has four different kinds of participants shown in Figure 12: Patients, Providers, Hospital Administrators, and “RoleOrgs”. To mirror current organizational practices, Hospital Administrators, *ProviderAdmin* in code, are the only participants allowed to assign a role to or remove a role from individual *Provider* participants, and they can accomplish this by submitting an *AddRoleOrg* or *RevokeRoleOrg* transaction to the network. The abstract participant *ProviderEntity* encompasses the *Provider, RoleOrg, and ProviderAdmin* participants. Providers are all individual healthcare providers—physicians, nurses, assistants, billers, etc.—identified by a
<table>
<thead>
<tr>
<th>File Owner (a Patient participant)</th>
<th>Access asset</th>
<th>EHR file</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Read, Write:</strong> through AccessGrant and AccessBlock transactions, which alter the data stored in the asset</td>
<td><strong>Read:</strong> Patients have access to their medical files; however, there is potential to have some files, such as psychiatric history, be kept hidden from the actual file owners themselves</td>
<td></td>
</tr>
</tbody>
</table>

| Providers (a ProviderEntity participant) | **Write:** in terms of requesting access, this can update the data stored in the asset; providers cannot change components of the blockedList or the accessList | **Write, Read:** if they are not blocked in the blockedList and they are members of a role(s) contained in the accessList |

Figure 9: This access matrix distinguishes which participants can do what to which objects in my system. The two main objects are the Access asset stored on chain, and the EHR record file stored off-chain. The data contained in the Access asset defines the ability of providers to read the EHR record file. The two main classes of participants, Patients and Providers only interact with the data in the Access asset itself by submitting transactions to the blockchain network, which, once validated via the consensus mechanism, alter the data stored in the asset. Patients that are not the file owner have no rights to either the Access asset or the EHR record file.

Figure 10: This figure shows the structure of an Access asset, written in Hyperledger Composer modeling language. Changes to this asset can only be made by the chaincode, explained in Chapter 5.7, executing transactions sent in the network. The blockedList and accessList can contain role groups, or RoleOrg participants, or individual providers. Only the file owner can submit transactions that alter these lists. When a Provider participant requests access, it is granted only if that provider’s individual identifier or role group identifier is not in the blockedList and is in the accessList. Upon verifying these conditions, the access is granted and the accessCount attribute is incremented.
Figure 11: This figure displays an *Access* asset stored on the ledger. In this example, only one role group, *RoleOrg 2002*, has been granted access to the file. This *RoleOrg* contains three providers, who thus all have access to the file. Only the file owner, *Patient* participant 0001, has the ability to view the contents of this *Access* asset.

unique identifier. The *Provider* participant contains a reference to the *RoleOrg(s)* to which that provider belongs. A *RoleOrg* pairs a role and organization and represents the most essential piece of the RBAC scheme I employ. As seen in Figure 10, the Access asset lists granted and blocked access in terms of *ProviderEntity*. Thus, both *RoleOrgs* and individual *Providers* can be contained in the list, and, by default, access will flow based on roles (discussed in Chapter 5.5). Patients then have the option to grant or block access to individual providers. Thus, by defaulting to granting access based on *RoleOrgs*, this design mimics current healthcare RBAC protocols, yet it also offers those patients who are more concerned about their privacy greater control with individual *Providers*. This addresses the spectrum of patient desires observed in my survey investigation.

Patients are the only ones able to grant initial access to or block access from their record files for a *RoleOrg* or individual *Provider* by submitting *AccessGrant* or *Access-
Block transactions respectively. Furthermore, only Providers can submit a RequestAccess transaction, and the transaction will only be processed if the requesting Provider is also the currently authenticated participant. This is to prevent a malicious actor from attempting to pose as a different provider when requesting access to a file. Figure 13 displays the relationship between transactions and participants.

![Diagram of transactions and participants](image)

Figure 12: This figure shows the structure both of the transactions and the participants I modeled for my network. The ledger, and thus the state of access, can only be altered through the chaincode’s handling of these transactions. Transactions are also restricted to certain participants. Figure 13 displays the relationship between transactions and participants.

5.7 Smart contracts

Automated programs, or smart contracts, form the logical backbone of my access control network. Hyperledger refers to these contracts as “chaincode,” and, in my imple-
Figure 13: This figure categorizes the relationship between transactions and participants in my blockchain network. The chaincode, discussed in Chapter 5.7, enforces these relationships by verifying the identity of participants, each of whom are cryptographically authenticated to the network with a public-private key pair.

Table 1.1: Transactions and Participants with the ability to initiate this transaction

<table>
<thead>
<tr>
<th>Transactions</th>
<th>Participants with the ability to initiate this transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddRoleOrg, RevokeRoleOrg</td>
<td>ProviderAdmin</td>
</tr>
<tr>
<td>AccessGrant, AccessBlock</td>
<td>Patient (but also must be the Access asset’s corresponding fileOwner)</td>
</tr>
<tr>
<td>AccessRequest</td>
<td>Provider</td>
</tr>
</tbody>
</table>

The logic used in this case is presented below in Figure 14, and the actual code being executed is displayed in Figure 15. I evaluate the functionality of all my chaincode in Chapter 6.1. The chaincode handling AccessGrant and AccessBlock transactions are shown in Figure 17.

The chaincode also allows for added security to ensure that the right participants are submitting transactions. It is the chaincode that retrieves the current participant, authenticated to the network by their unique private/public key pair, and validates that
Figure 14: This figure depicts the logic flow of granting access upon receiving an \textit{AccessRequest} transaction. Access is only granted if the \textit{requestingProvider} of the transaction, who must be cryptographically authenticated to the network to submit the transaction, is both not in the \textit{Access} asset’s blocked list (either by role or individually) and is in the asset’s access list (again, either by role or individually).

that type of participant is allowed to submit the transaction. For example, only hospital administrators can submit \textit{RevokeRoleOrg} or \textit{AddRoleOrg} transactions, and only the \textit{Provider} currently authenticated to the network can submit a valid \textit{AccessRequest} transaction. The chaincode can also emit events when a certain condition is met; in this case, I create an \textit{AccessEvent} whenever an \textit{AccessRequest} transaction is processed. Although any transaction would also be shown in the immutable blockchain’s transaction history, an \textit{AccessEvent}, shown in Figure 16, directly reports the success or failure of access request and all other details associated with the transaction. Consequently, notifications can be emitted to a file owner about an access attempt based on the creation of an \textit{AccessEvent} in chaincode. See the Appendix for specific implementation of chaincode for handling an \textit{AccessRequest} transaction.

My chaincode implementation allows for automated role based access fused with additional rules submitted by patient file owners. It, when joined with the entire blockchain network, offers the trusted security, programmability, transparency, and flexible control
Figure 15: This figure displays the chaincode responsible for handling an AccessRequest transaction. After verifying the requesting provider's identity, the code ensures that the requesting provider is not blocked by fetching the associated Access asset and looping through its blockedList. The code then does the same for the asset's accessList, this time checking to see if the requesting provider is a member of any allowed role groups, or, in terms of my network setup, RoleOrgs. Finally, the code emits a special event, an AccessEvent, shown in Figure 16 below, which expresses the result of the request via the success variable—it is 1 upon a successful access, and 0 if access is denied. This event can be further developed as a notification in an application. After processing the transaction, it is permanently recorded in the blockchain and the file owner can view both the AccessRequest transaction and the transaction's result.
Figure 16: An AccessEvent event modeled in Hyperledger Composer, created by chaincode whenever handling an AccessRequest transaction. The `success` attribute is a binary variable indicating if access was granted (1) or denied (0). In future application development, this event can be subscribed to and emit notifications to file owners.

```javascript
event AccessEvent {
  // event attributes
  --> Access fileAccessed
  --> Provider requestingProvider
  o Integer success
}
```

Figure 17: This figure shows the chaincode used to handle `AccessGrant` and `AccessBlock` requests respectively. In handling these transactions, the chaincode updates the `Access` asset stored in the distributed ledger, and thus changes the state of access. Only the file owner can submit these transactions, and they can be submitted for either entire role groups or individual providers.

```javascript
/**
 * A transaction processor function description
 * @param {org.acme.biznet.AccessGrant} accessGrant - the access to be granted
 * @transaction
 */

function handleAccessGrant(accessGrant) {
  var fileOwner = accessGrant.fileAccess.fileOwner;
  var currentParticipant = getCurrentParticipant();
  if (currentParticipant.getFullyQualifiedName() !==
      fileOwner.getFullyQualifiedName()) {
      throw new Error('Access can only be granted to this file by the owner of this file');
      return;
  }

  accessGrant.fileAccess.accessList.push(accessGrant.accessEntity);

  return getAssetRegistry('org.acme.biznet.Access').
    then(function (assetRegistry) {
        return assetRegistry.update(accessGrant.fileAccess);
    });
}

/**
 * A transaction processor function description
 * @param {org.acme.biznet.AccessBlock} accessBlock - the access asset to be blocked
 * @transaction
 */

function handleAccessBlock(accessBlock) {
  var fileOwner = accessBlock.fileAccess.fileOwner;
  var currentParticipant = getCurrentParticipant();
  if (currentParticipant.getFullyQualifiedName() !==
      fileOwner.getFullyQualifiedName()) {
      throw new Error('Access can only be blocked to this file by the owner of this file');
      return;
  }

  accessBlock.fileAccess.blockedList.push(accessBlock.blockedEntity);

  return getAssetRegistry('org.acme.biznet.Access').
    then(function (assetRegistry) {
        return assetRegistry.update(accessBlock.fileAccess);
    });
}  ```
necessary of an EHR access solution. I evaluate the functionality of my chaincode in Chapter 6.1.

5.8 Additional Programmability

To further test the potential for this solution, I investigated the practicality of the network being incorporated in a developer-friendly and user-friendly application. Due to Hyperledger Composer’s focus on solving business problems, it is made to compatible with many modern JavaScript tools. This was a significant reason for my choice in using the platform, as it is made easier for developers to adopt, expand, and develop upon a blockchain business network. I used two main tools to further expand upon the EHR access blockchain network. First, I used Loopback to generate Rest APIs for interacting with the distributed ledger. This allows internal organization developers to further customize applications either for internal use or for improved engagement with patients. These Rest APIs can also be used in a Node.js application allowing patients to interact with the ledger, view their records, grant access, block access, and view their access histories. A screenshot of these calls are shown in Figure 18.

Furthermore, I used Yeoman, a tool for scaffolding modern web applications, to help generate a skeleton Node.js application. Further developing this application is the primary subject of future work, as it would offer patients and providers a clean user interface for granting, blocking, requesting, and viewing access to their medical records. For now, it can interact with my blockchain network via API calls and return data from the ledger, but has not been developed as a user-facing application. The success of integrating with
Figure 18: This figure displays the REST APIs I created using the Loopback tool for programmatically interacting with the EHR access blockchain network. With these API calls available, this blockchain could be easily incorporated into a variety of customized applications, which suggests this network could be more easily adopted into current healthcare workflows.

these tools for modern development shows the potential for this EHR access control network to be easily built upon and adopted in the future. A full diagram of my EHR access system including future application development is shown in Figure 19 below, and Figure 20 shows a mock-up of the future user-facing application, used in my survey investigation. This application would allow patients to view accesses to their records and submit access transactions to the blockchain network.

5.9 Issues

I encountered a few issues during development that are worth noting. First, due to the novelty of the Hyperledger Composer framework, I often encountered bugs related to compatibility and dependency issues. There is also very little documentation on these
Figure 19: This figure displays the full design of my EHR access control system. It integrates my blockchain network with a user-facing application and cloud-stored EHR files. The blockchain access control network adds a trusted, secure way to enforce automated access rules, customized by patients, for a distributed EHR system used by many healthcare organizations. With its immutable history, patients can easily view accesses to their records and adjust rules accordingly.

Figure 20: This figure displays the mock-up of my blockchain network’s user-facing application, used in my survey investigation. This application would interact with the blockchain network via API calls, shown in Figure 18 above, and allow patients to view accesses to their records and submit AccessGrant and AccessBlock transactions if they desire. Similarly, providers would have a separate set of options allowing them to search for patient medical records, request access, and retrieve those medical record files.
errors, as each bug tends to be relatively new. While I developed with the Hyperledger Composer version 16 stack, the network failed upon upgrading to the newly released version 19. Previous versions of the network (I created four different versions throughout development, each one packaged as a separate “business network archive” (.bna) file) could not be deployed on the newer architecture. Consequently I have rolled back to version 16 and continued work there. More documentation and stability in the framework should appear naturally over time, but it makes current development difficult if encountering unknown errors.

Furthermore, authenticating to the network with separate identities posed issues as I had more than one identity, known as a business cards in Composer, stored locally. Transferring the .bna file to a separate workstation, creating an initial identity, and deploying the network all work smoothly, yet deploying new versions of the network with old identities often fails with unclear errors. Again, documentation should be improving for the Composer framework, yet developers should be aware that, although Composer implies this should not be necessary, identities must be created fresh when deploying a new version of the network.

6 Evaluation

To evaluate the success of my solution in balancing patient control and practical access policy for EHRs, I consider two factors: functionality and practicality. Functionality encompasses the success of my blockchain network in creating and enforcing access rules consistent with current healthcare workflow. Practicality encompasses the success of my
blockchain network in being usable, desirable, and adopted by patients and providers alike. I evaluate my solution on these two factors below.

6.1 Functional Analysis

If my system is to be a functional solution for balancing EHR access rules, then it must replicate access rules of current healthcare organizations, properly enforce these rules, and still offer patients greater control than they previously had. By pairing organizations and roles with the RoleOrg participant, the network is able to mimic the role structure used in healthcare organizations. I test to ensure that access is granted by RoleOrg or individual Provider IDs only, and to ensure that no participant outside the allowed roles are granted access by the network. I also ensure that the added patient control functions properly. One specific example of these tests, shown in Figures 21-23, outlines a file accessible by Outpatient Physicians at Good Samaritan Hospital, yet one physician, Doctor Smith, has been explicitly blocked through an AccessBlock transaction submitted by the file owner. Thus, initial access is granted just as it would be at most large healthcare organizations, by role to outpatient physicians at the organization, yet the patient chooses to specifically block an individual for personal reasons. Therefore, access should be granted to all outpatient physicians at Good Samaritan Hospital yet be denied to Doctor Smith. This functions properly, as does granting access, blocking access, assigning a provider to a new role, and removing a provider from a role.

In terms of authentication and identity management, the network succeeds in only allowing EHR file owners to grant, block, and view accesses to their records. Furthermore,
Figure 21: This figure shows an *Access* asset, file ID 3001 owned by *Patient* 0001, stored in the ledger. In the *accessList*, there is one *RoleOrg* ID 2002, that contains 3 individual providers. One of those providers is ID 1002, Dr. Smith, who is also specifically blocked by the file owner as evidenced by his placement in the *blockedList*. Thus, every member of Dr. Smith’s role should have access, but Dr. Smith himself should be denied access to the file. Such a scenario would arise if a patient knew that a coworker or friend at the organization would have access to the file, and that patient wanted to specifically block this from happening. This is an example where the patient, not the organization, is the best auditor of proper access to his/her records.

```json
3001
{
   "$class": "org.acme.biznet.Access",
   "fileID": "3001",
   "accessList": [
      {
         "$class": "org.acme.biznet.RoleOrg",
         "organization": "Good Samaritan Hospital",
         "role": "Outpatient Physician",
         "providers": [
            "resource:org.acme.biznet.Provider#1001",
            "resource:org.acme.biznet.Provider#1002",
            "resource:org.acme.biznet.Provider#1003"
         ],
         "entityID": "2002"
      }
   ],
   "accessCount": 1,
   "blockedList": [
      {
         "$class": "org.acme.biznet.Provider",
         "firstName": "Doctor",
         "lastName": "Smith",
         "roleOrgs": [
            "resource:org.acme.biznet.RoleOrg#2002"
         ],
         "entityID": "1002"
      }
   ],
   "fileOwner": "resource:org.acme.biznet.Patient#0001"
}
```

Figure 22: This figure shows the *AccessRequest* transaction being submitted by Dr. Smith for a record file he is blocked from (Figure 21). Figure 23 confirms that this request is properly denied, despite his role being granted access. Thus, the chaincode is able to fuse default RBAC rules with more granular rules specified by patient file owners.
Figure 23: This figure shows the result of the AccessRequest transaction, shown above in Figure 22, submitted by Dr. Smith in the immutable history of the blockchain. When the chaincode processed the transaction, it emitted a special AccessEvent notification where the success variable = 0, indicating that the request was denied. This event appears when the file owner, Patient 0001, queries the blockchain for access requests to his/her records, thus adding transparency to who is accessing records when.
Figure 24: This figure shows a simple attempt to grant access, via an `AccessGrant` transaction, that is not submitted by the file owner. The chaincode processes this by retrieving the requested `Access` asset, checking the file owner associated with that asset, and verifying that the currently authenticated participant is that same file owner. Thus, file owners are the only ones able to add granular access rules for their records—the same is true for `AccessBlock` transactions.

```json
{
    "$class": "org.acme.biznet.AccessGrant",
    "fileAccess": "resource:org.acme.biznet.Access#3001",
    "accessEntity": "resource:org.acme.biznet.RoleOrg#2892"
}
```
it prevents users from submitting transactions on behalf of other participants, thus ensuring that access rules are being properly followed. Finally, hospital administrators, registered to the network as ProviderAdmin participants, are the only users able to adjust the roles of other providers. This mirrors the organizational structure currently in place at medical institutions. An attempt to grant access while being authenticated to the network as a provider, not the file owner, is shown in Figure 24, and an attempt to request access as a different provider than specified in the request is shown in Figure 25; the specific chaincode implementation of this is included in the Appendix.

To fully analyze the functionality of this system, I propose a testing methodology as future work. This system is functional if, when a patient visits two or more separate healthcare organizations:

1) The state of access displayed in the distributed ledger aligns entirely with the state of access exhibited at each individual organization.

2) Medical providers during those visits are able to quickly obtain the patient’s medical information necessary for their treatment.

Executing this test would require running scenarios of patient visits at two healthcare organizations, having providers use both their organization’s internal EHR system and my external EHR system, then comparing the times taken to access the necessary file and the state of access rules for both systems. Implementing this test and thus gaining a complete analysis of my system’s functionality is outside the scope of my thesis work, but this remains necessary future work for evaluating the potential of my EHR access control system as a valid solution. For now, I can only ensure that my system properly executes
Figure 25: This figure shows an attempt to request access by posing as a different provider, notably one who does have access to the requested file associated with Access asset 3001. The chaincode processes this by ensuring that the requesting provider contained in the JSON data, or, in the case of an application, the API call translated to JSON data, is the same as the participant currently authenticated to the blockchain network. Of course, multiple participants could be authenticated to the network at once, yet accounts transactions are tied to the exact authenticated account from which they originate. Thus, the data in the request must match that same authenticated account. Here, the request fails since the requesting provider identified in the transaction is not the same as the provider authenticated to the network. Notably, this failure is not registered in the permanent history of the blockchain, it is just denied as a failed transaction, rather than accepted as a valid transaction to which access is denied. As such, a limitation of my blockchain network is that malicious attempts will fail, but fail quietly. In a full application, this could be accounted for with error logs managed by an administrative regulator, likely the same administrator who must assign Provider participants to appropriate RoleOrg groups.
and enforces the RBAC based rules I designed to mimic those currently used in healthcare organizations, yet, with the flexibility of Hyperledger Composer, these rules can be easily adjusted upon insights from real tests in healthcare organizations.

6.2 Practical Analysis

If my system is to be a practical solution for balancing EHR access rules, then it must allow providers access to files they need when they need it, be desired by patients, and be easily adopted by providers. Practicality must be evaluated from two sides: patients and providers. My user survey investigates practicality from a patient perspective, and the testing methodology discussed above in Chapter 6.1 would evaluate important aspects of provider practicality. First, it is clear that patients desire more control and transparency over who views their medical records. As mentioned earlier, only 6% of patients said they were unlikely to use a system like the one I propose. Also, by relying on nudge theory and using smart contract automation mirroring the RBAC schemes of organizations, patients do not have to be actively involved with the access rules of their medical records—there is defined default behavior. Greater control is there if they want it, but otherwise they do not have to take any action. On the other side, the goal of this system is to mimic the access protocols currently used internally on an external, distributed system. Therefore, if the current RBAC schemes work well internally, then this EHR access system should still be practical from a provider’s perspective.

However, there is one key difference to note: patients must still grant the first initial access to a role group in an organization. While organizations could customize what
constitutes a role group, they would still require the patients to sign off on a new role
group gaining access. Patients already do this when they first visit a healthcare organi-
zation—HIPAA requires that patients explicitly grant the organization the ability to share
their records as needed—yet once signed in person, those organizations are able to adjust
which providers and roles can access the records internally [40]. Consequently, one poten-
tial barrier to practicality for this system is the need for patients to explicitly grant an
organization’s role group initial access to their records. Evaluating the full implications of
this limitation on practicality remains an aspect of future work to be incorporated with
tests in real hospital settings. From a patient perspective, user survey results and prior lit-
erature indicate that patients desire this engagement and would be willing to grant access
as needed—only 9% of my survey’s respondents value the privacy of their records over the
quality of their care. However, from discussions with healthcare experts, patients may re-
quire additional information about why a specific role or provider is requesting access [23,
24, 25]. As providers are already pressed for time and burdened with regulations, a truly
practical system should incorporate a way to very easily or even automatically express
why access is required. In a full solution built on top of my blockchain EHR access net-
work, this added explanatory feature could significantly mitigate the potential limitation
of patients granting initial access to RoleOrg participants. Overall, since patients desire
more control and this system is intended to mimic existing healthcare access policies, there
is strong reason to believe it is practical and adoptable. However, EHRs are in a state of
constant evolution; a complete evaluation of practicality must analyze the severity of po-
tential limitations in America’s changing healthcare landscape.
6.3 Limitations

My blockchain-based EHR access system has a few notable limitations that must be considered in evaluating its potential as a valid solution. First, as discussed in the above practical analysis, patients must still provide initial access to each different organization-role pair. While organizations could customize the structure of their roles to allow for more flexibility, there may be issues with automating workflows where access must be granted from physician to nurse to biller if patients are not actively granting access when requested. Ideally, patients grant initial access to all necessary roles at the request of organizations upon registration.

Furthermore, this system does not address EHR records owned by the organizations themselves; some information recorded about a patient, especially mental health information, must not be accessible to the patient. While this concern is outside the scope of this project, this EHR access network could still account for it. Another asset, *PrivateAsset*, could refer to files referencing a *Patient* participant but be inaccessible to the corresponding patient individual. The *Patient* could still grant, block, and view access to this file, yet the chaincode would enforce that the patient could not request access to this file. In this way, even important medical information hidden from the patient could be made interoperable amongst providers while still offering the patient greater control over how that information is accessed.

Finally, this network does not address emergency situations in which the patient is not able to grant initial access for providers. A *break-glass* protocol is an essential part of any EHR access system—a requirement of a solution is allowing providers to view the
information they need exactly when they need it. To address this limitation, I propose an additional kind of transaction, *BreakGlassRequest*, which could be submitted by any provider to gain access to a file. However, when processing this transaction, the chaincode would emit an *EmergencyEvent* to all participants in the network, thus ensuring there is accountability for any provider issuing this request. Prior work on break-glass protocols are outside the scope of my work, yet any plan of implementation could easily be incorporated into the flexible Hyperledger chaincode and network model.

### 6.4 Results

With everything I have presented, a natural question arises: has my work solved a problem facing EHR access control? Prior literature and my own survey investigation both support patients’ desires to have more control over who is accessing their records. Also, government initiatives, along with recent work on blockchain-backed EHRs, promote interoperability and patient-owned, universal, holistic health records [7, 8, 10, 32]. In doing this, the literature has pushed for a patient-centric EHR system, where patients completely own their records and must grant specific granular access to any provider needing it. For example, MedRec [7], the primary example of prior work on a blockchain-backed EHR system, requires patients to grant access to individual providers, and only for very specific data queries within their records. I found this direction of patient centrism to be largely unpractical for healthcare delivery. My survey investigation confirms that most patients do not understand who and how many providers must view their health data, and it also confirms that the majority of patients are not very concerned about who may be
accessing their data—they trust it is being used properly by providers. However, they indicate strong interest in an application that would allow them to easily view who has accessed their records (a right already granted to them by HIPAA).

Thus, I saw a problem: assuming we achieve universal, patient-owned records, patient-centric access management will be impractical for providers to deliver care (physicians, nurses, assistants, administrators, billers, etc. must all view health records for varying purposes) [22, 23, 24, 25]. I saw a solution in recreating general individual organization RBAC schemes on an external, universal system. Based on nudge theory, I could default access to these rules yet still allow patients the additional control to grant access and block access to individual providers (Chapter 5.5). Patients would also gain transparency into who views their records. Since this problem involves many different participants, has rules that could be automated and customized, and has a need for a secure, immutable history of behavior, blockchain was a good candidate for implementing this solution (Chapter 5.1).

As discussed above in 6.1-2, without an end-to-end system deployed in a case study hospital setting, I cannot make conclusions on the effectiveness of my solution. However, my blockchain network prototype succeeds in defaulting to my simple organization-role based access control scheme, and is able to fuse these rules with additional blocked access and granted access rules specified by patients. Additionally, all activity in the network is permanently recorded in the immutable blockchain—I, posing as a patient, can query the blockchain for a specific EHR file and view all past access requests to that file, along with whether or not they were granted or denied access. The network can be integrated via REST APIs with a Node.js application, suggesting that it could be easily adopted and customized by healthcare organizations. Considering the conflicting demands of patients
and providers for EHR access—privacy vs. flexibility—I find blockchain, manipulated to model access states, automate an organizational RBAC scheme, and provide an immutable history of behavior in the network, to be a very plausible solution for balancing patient desires and provider needs. This idea is, to my knowledge, the first example of blockchain’s use for less patient-centric, nudge theory-based EHR access control, an idea that could align access control interests as academics, the government, and the healthcare industry make strides towards interoperable, universal patient records.

7 Future Work

As a time-constrained investigation, there are some focused items of future work for this project. Primarily, this blockchain-based EHR system must be evaluated in a real-world organizational setting. To do this, the system must be completed end-to-end. As of right now, only the blockchain network and skeleton application has been implemented (see Figure 19 for a diagram on the complete system). As a result, a key item of future work includes building out the application into a user-friendly interface for both patients and providers. Mock-ups of a full application were used in the user survey and are shown in the Appendix. Furthermore, complementary systems discussed above in Chapter 5.2—implementing the token-based fetching of encrypted files stored on the cloud and integrating my blockchain network with the authentication systems of healthcare organizations—remain necessary pieces of a complete distributed EHR access system. With a complete system, a proper analysis on practicality and functionality could be performed in a hospital setting with real patients, conditions, and healthcare workflows. Additionally, by customizing
the chaincode and RoleOrg participants to a very specific healthcare organization as a case study, I could evaluate the level of ease with which this system could be adopted and integrated into the American healthcare industry. Until then, my work remains largely theoretical; it contributes to the literature regarding blockchain’s potential for EHR records and patient-centric EHR systems by proposing a more practical way to improve patient control and transparency.

8 Conclusion

My work began with an investigation of blockchain and an identification of its key features for potentially solving problems. Due to its immutability, automated smart contracts, and ability to promote trust amongst varied participants, I identified it as a strong candidate to address the access control problems of EHR records. After initially analyzing the current state of EHR access and recent work applying blockchain to the EHR realm, a general push towards patient-centric health record management became clear. I hypothesize that, assuming we achieve a universal, patient-owned EHR system, patient-centric access will be impractical for healthcare delivery. As provider accessibility to health records presents a major barrier to the practicality and success of universal, interoperable, patient-owned health records, my work investigates the potential of a blockchain network to balance patient control and provider accessibility with a two-fold approach. First, I conduct a user survey to identify patient concerns and determine the level of control patients would like over their health information. Second, I implement a blockchain network prototype to address the spectrum of patient control preferences and automate practical, existing access
policy for providers on a distributed scale. My survey investigation, although biased by a more tech-savvy sample on Amazon Mechanical Turk, confirms that 1) patients largely do not understand how their data is used by various providers and 2) patients desire more control and transparency over who is accessing their records.

My solution is based on a default organization-role based access scheme, where patients can easily submit access “transactions” to the network to specifically grant or block access from certain providers. My blockchain solution also allows patients to view exactly who has requested access to their EHR files. Considering the conflicting demands of patients and providers for EHR access—privacy vs. flexibility—I find blockchain, manipulated to model access states, automate an organizational RBAC scheme, and provide an immutable history of behavior in the network, to be a very plausible solution for balancing patient desires and provider needs. This idea is, to my knowledge, the first example of blockchain’s use for less patient-centric, nudge theory-based EHR access control, an idea that could align access control interests as academics, the government, and the healthcare industry make strides towards interoperable, universal patient records.

9 Acknowledgements

I have several important acknowledgments to make, as I owe many people thanks for their help throughout this thesis research. First, I would like to thank my family and friends who were always patient in me discussing these topics with them and gathering more feedback. Specifically, I would like to thank my parents, Dr. Helen Koselka and Dr. Ed Horton, who helped both in terms of offering insights from the healthcare sphere and
connecting me with other colleagues in their fields. I would also like to thank my thesis advisor, Professor Sean Smith, for his guidance throughout the many parts of this process, including solidifying my initial topic, finding helpful connections and resources, and reviewing draft papers. I also am very appreciative of Dr. Ross Koppel for his thoughtful assistance in educating me on survey techniques and the state of electronic health records, and also for reviewing my survey investigation. Additional thanks go to Dr. Andrew Gettinger for his insights into electronic health records and patient concerns, which helped solidify my research topic, and to Professor Denise Anthony for addressing my questions and concerns from a sociological angle. Also, thank you to the medical informatics team at Tri-Health in Cincinnati, Ohio, who frequently answered questions regarding their experiences and workflows. Finally, thank you to my friend and fellow researcher Jessie Anderson for reviewing my work, offering insight, discussing these topics, and supporting this process.

10 References


“Individuals’ Right Under HIPAA to Access their Health Information 45 CFR § 164.524.”


11 Appendix

This appendix contains my survey investigation questionnaire and my main smart contracts, written in JavaScript, and available at https://github.com/elenahorton/EHRAccess along with the rest of my code.
Patients and Electronic Health Records

This survey will walk through some questions about your background, medical care history, and beliefs about your electronic health records. It will then continue with some questions about a healthcare application. All responses are completely anonymous.

* Required

1. Please enter your MTurk Worker ID *

2. How old are you?
Mark only one oval.
- Under 21
- 21-29
- 30-39
- 40-49
- 50-59
- 60-69
- 70-79
- Over 80

3. What is your gender?
Mark only one oval.
- Female
- Male
- Transgender
- Prefer not to say
- Other: __________

4. What is the highest degree you hold?
Mark only one oval.
- No high school diploma
- High school diploma
- Some college
- Bachelor's degree
- Master's degree
- Doctoral degree
5. How often do you see some form of medical provider (general practitioner, dentist, counselor, etc.)?
   Mark only one oval.
   ☐ Less than once a year
   ☐ About 1-5 times a year
   ☐ About 6-10 times a year
   ☐ About 11-20 times a year
   ☐ More than 20 times a year

6. Have you ever been hospitalized?
   Mark only one oval.
   ☐ Yes, for more than 3 days
   ☐ Yes, for more than a day
   ☐ Yes, for less than a day
   ☐ No

7. At about how many different medical offices or organizations (including dentist, physical therapy, counselor, general practitioner’s office, etc.) have you been a patient?
   Mark only one oval.
   ☐ None
   ☐ 1-2
   ☐ 3-5
   ☐ 6-10
   ☐ More than 10

8. If applicable, why have you seen more than one medical provider? Check all that apply.
   Check all that apply.
   ☐ Different medical specialties
   ☐ Found a new provider after being dissatisfied with another
   ☐ I moved
   ☐ My provider moved or retired
   ☐ I required treatment while I was away from home
   ☐ Emergency situation required nearest treatment center
   ☐ Other: ____________________________________________

9. How difficult in the past has it been for you to switch providers?
   Mark only one oval.
   ☐ I’ve never switched providers
   ☐ Not difficult at all
   ☐ A little difficult
   ☐ Somewhat difficult
   ☐ Annoyingly difficult
10. If it was in some way difficult for you to switch providers, please briefly explain why it was difficult.

11. Who do you share your medical information with? Check all that apply.

Check all that apply.

- Parents
- Partner
- Spouse
- Children
- Friends
- Coworkers
- Employers
- Other:

12. Do you feel that any of the medical conditions you have are ‘embarrassing’?

Mark only one oval.

- Yes
- No
- Somewhat

Regarding Electronic Health Records

Electronic Health Records are digital files of your personal health information acquired by doctors from your visits. Most healthcare organizations store these on-site and have access policies that determine which medical professionals can access your information.

13. Are you familiar with the term “Electronic Health Record” or “EHR”?

Mark only one oval.

- Yes
- No

14. How have you been exposed to your electronic health record (EHR)? Check all that apply.

Check all that apply.

- I saw my doctor entering or reading information on it when I was in the office
- Not sure, I don't know what an EHR is
- I logged into view it through an online portal
- I have discussed it with my physician
- I received a printed copy of a ‘visit summary’ with some of my medical information
- Other: ________________________________________________
15. Have you done any of the following? Check all that apply.  

   Check all that apply.
   - Logged into an online portal to view my health records
   - Emailed/called my provider
   - Asked questions of nurses, physician assistants, etc. regarding my conditions
   - Done outside research (internet searching, reading, etc.) on my conditions
   - Maintained some form of a “health log” for myself

16. How concerned are you about who is accessing your EHR medical records?  

   Mark only one oval.
   - Not concerned
   - A little concerned
   - Somewhat concerned
   - Very concerned

17. Why are you concerned about people accessing your medical records? Check all that apply.  

   Check all that apply.
   - I'm not concerned
   - Someone exploiting my information for profit
   - Someone judging my personal information
   - Someone I know might find out a health issue that I want to keep private
   - Someone using my health information against me/family member/friend
   - Other: ____________________________

18. Who do you think has had access to your electronic health records in the past?  

   Check all that apply.
   - Physicians
   - Nurses
   - Insurance/Medical Billers
   - Students
   - Administration
   - Family/Friends
   - Employers
   - Other: ____________________________
19. For one visit to the hospital, about how many people do you think actually view your electronic health records?

Mark only one oval.

☐ Less than 2
☐ 3 to 5
☐ 6 to 10
☐ 11 to 20
☐ More than 20

20. Are you actively concerned by the number of people who may have looked at your health records?

Mark only one oval.

☐ Yes
☐ No
☐ Somewhat
☐ Not sure

21. For one visit to the hospital, about how many people do you think can potentially access your medical records?

Mark only one oval.

☐ Less than 2
☐ 3 to 5
☐ 6 to 10
☐ 11 to 20
☐ More than 20

22. Are you concerned by the number of people who may have access to your records?

Mark only one oval.

☐ Yes
☐ No
☐ Somewhat concerned
☐ Not sure

Some proposals suggest that patients should own full access rights to their medical records, which would require the patients themselves to directly grant individual access to any medical professional necessary. This could be multiple nurses on a team or multiple doctors if they are collaborating on treatment, and it would require you to actively manage your health records.
23. **Do you want to be responsible for granting each individual provider explicit access to your records?**
   *Mark only one oval.*
   - [ ] No, I prefer to let medical professionals handle this
   - [ ] Yes, I want to select which (if any) clinicians can see my record

24. **For one visit to a healthcare provider, about how much time do you think you'd be willing to spend managing access to your health records?**
   *Mark only one oval.*
   - [ ] I don't want to manage access to my health records at all
   - [ ] Less than 2 minutes
   - [ ] 3-5 minutes
   - [ ] 6-10 minutes
   - [ ] 11-15 minutes
   - [ ] 16-30 minutes
   - [ ] As much time as it takes

25. **From your own experience, would you be fine with any of the following people accessing your records for your treatment without explicit permission? Check those with which you would be fine with:**
   *Check all that apply.*
   - [ ] My physician
   - [ ] My physician's colleagues
   - [ ] Hospital administrators
   - [ ] My physician's assistants
   - [ ] Medical students
   - [ ] Another consulted physician with a different specialty
   - [ ] My nurses
   - [ ] My health insurance
   - [ ] My pharmacist

26. **To what degree do you agree with this statement: It's ok for me to withhold some of my health information from my providers if I don't think it's important and prefer to keep it private.**
   *Mark only one oval.*
   - [ ] 1
   - [ ] 2
   - [ ] 3
   - [ ] 4
   - [ ] 5
   | Strongly disagree | 1 | 2 | 3 | 4 | 5 | Strongly agree |

27. **What do you value more: the privacy of your medical information or the quality of your care?**
   *Mark only one oval.*
   - [ ] Privacy of my records
   - [ ] Quality of my care
   - [ ] Value both equally
28. **Would you like to be notified whenever someone accesses your health records?**
   *Mark only one oval.*
   - [ ] Yes, regardless
   - [ ] Yes, but only if it's not a clinician I already know
   - [ ] No

29. **If you could see exactly when and who accessed specific parts of your medical records, would you be more lenient about the access policies for your medical records?**
   *Mark only one oval.*
   - [ ] Yes
   - [ ] Maybe
   - [ ] No
   - [ ] Not sure

30. **Have you ever wanted to share your EHR health information with a trusted third party source for a second opinion (family, friend who is a physician, etc.)?**
   *Mark only one oval.*
   - [ ] Yes, I have wanted to
   - [ ] Maybe, it would have been nice
   - [ ] No

31. **If you controlled your electronic health records, what would you like to be able to do? Check all that apply.**
   *Check all that apply.*
   - [ ] Grant access to a specific person
   - [ ] Block access for a specific person
   - [ ] View who has accessed my records
   - [ ] Add to my own health records
   - [ ] Delete from my own health records
   - [ ] Other: ____________________________________________

**Healthcare Application Walk-through**
32. From this application home page, what do you believe is the relationship between Dr. Smith and Dr. Johnson?

Mark only one oval.

- No known relationship
- Student/Teacher
- Colleagues
- Other:

33. Assuming this relationship, are you comfortable with Dr. Johnson accessing this medical record?

Mark only one oval.

- Yes
- No
- Maybe
34. Do you prefer to think of your medical history as one complete unit or as separate components (such as physical history, mental health, dental, etc., similar to as shown on the application screen above)?

Mark only one oval.

- One complete medical history
- Separate files of my medical history
- Somewhere in between
35. **Have you ever withheld information from your provider for privacy concerns?**
   *Mark only one oval.*
   - Yes
   - No
   - Not sure

36. **If you could monitor and adjust access for separate components of your medical history, would you have been less likely to withhold any information from your provider?**
   *Mark only one oval.*
   - I've never withheld information
   - Yes
   - Maybe
   - No
You have 5 Records

- General Medical History
  Record ID: fjdiosa894289sdhias

- Medication History
  Record ID: 3289fdhsad3jdbsa

- Procedural History
  Record ID: gjfdlaw38384xadd

- Mental Health History
  Record ID: wqui4920dsk133

- Dental History
  Record ID: dsjioaf31903ajjd1

Access to ‘General Medical History’
Record ID: fjdiosa894289sdhias

Currently granted to:

- American Healthcare Providers
  Roles: Outpatient Physician, Outpatient Nurse

- Good Samaritan Hospital
  Roles: Inpatient Physician, Inpatient Nurse

- Family General Practitioners
  Roles: Outpatient Physician, Outpatient Nurse

- Dr. Jessie Doe from Bethesda Hospital
  Roles: Individual Access

Currently blocked:

- Dr. John Smith from Family General Practitioners
  ID #: j683290jja13

Actions

- View Record
- View Access List
- View Access History

- Revoke Access
- Grant New Access
- Add/Remove Blocked Access
- View Access History
37. By looking at the application screen shown above, who has access to the patient's 'General Medical History' record? Check all that apply

Check all that apply:

- All physicians from Bethesda Hospital
- Dr. Jessie Doe from Bethesda Hospital
- Everyone at Family General Practitioners
- Only one specific doctor at Family General Practitioners
- All outpatient doctors at Family General Practitioners
- All outpatient nurses at Family General Practitioners

38. If Dr. John Smith, an outpatient physician, was not specifically blocked by the patient, do you think he would be able to access the record in this system?

Check all that apply:

- Yes
- No

39. Briefly explain why you chose yes or no:

The following series of actions grants new access to this medical record:
40. What do you believe the 'default standard access' referred to?
Mark only one oval.

- Only Dr. Alice Fisher has access
- Individuals with the same role as Dr. Fisher at the same organization have access
- Only Dr. Fisher and her known colleagues have access
- All individuals in Dr. Fisher's organization have access
41. Would you be comfortable with this method of granting access?
   *Mark only one oval.*
   - Yes
   - No
   - Not sure

42. From your current understanding of hospitals, who do you think would have access to this record if the patient went for an outpatient visit?
   *Mark only one oval.*
   - Every employee of the healthcare organization
   - Only the patient's physician and assistants
   - Only the patient's physician
   - Only the outpatient nurses and doctors at the organization

43. Does this application system seem more secure, just as secure, or less secure than what you believe about how your records are currently accessed by healthcare organizations?
   *Mark only one oval.*
   
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Much less secure</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Much more secure</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

44. How likely would you be to use an EHR access management application like this?
   *Mark only one oval.*
   
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not at all likely</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Very likely</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

45. Thank you for taking this survey! Please write any additional comments below:

   ___________________________________________
   ___________________________________________
   ___________________________________________
   ___________________________________________
'use strict';
/**
 * Transaction processor functions
 */

/**
 * A transaction processor function description
 *
 * @param {org.acme.biznet.AccessRequest} accessRequest - the access request to be tested, then
granted or denied
 * @transaction
 */

function handleAccessRequest(accessRequest) {

    var me = getCurrentParticipant();
    console.log('@debug: **** REQUESTING ACCESS: ' + me.getIdentifier() + ' requesting access of'
' + accessRequest.fileAccess);

    if(!me) {
        throw new Error('A participant/certificate mapping does not exist.');
    }
    if (me.getIdentifier() !== accessRequest.requestingProvider.entityID) {
        throw new Error('Requesting provider is not the current participant, request denied.');
    }

    var len = accessRequest.fileAccess.accessList.length;
    var roleOrgIDs = accessRequest.requestingProvider.roleOrgs;
    var lenRoles = accessRequest.requestingProvider.roleOrgs.length;
    var success = 0;
    var blocked = 0;

    // first ensure that the requesting entity is not blocked
    for (var b = 0; b < accessRequest.fileAccess.blockedList.length; b++) {
        if (accessRequest.requestingProvider.entityID ==
accessRequest.fileAccess.blockedList[b].entityID) {
            blocked = 1;
            break;
        }
    }

    var i = 0;
    while (i < len && success == 0 && blocked == 0) {
        for(var j=0; j< lenRoles; j++) {
            var allowedEntity = accessRequest.fileAccess.accessList[i].entityID;
            var requestingRoleOrg = accessRequest.requestingProvider.roleOrgs[j].entityID;
            if (allowedEntity == requestingRoleOrg || allowedEntity ==
accessRequest.requestingProvider.entityID) {
                accessRequest.fileAccess.accessCount++;
                success = 1;
                break;
            }
        i++;
    }
}

    return getAssetRegistry('org.acme.biznet.Access')
        .then(function (assetRegistry) {
```javascript
var accessNotification = getFactory().newEvent('org.acme.biznet', 'AccessEvent');
accessNotification.fileAccessed = accessRequest.fileAccess;
accessNotification.requestingProvider = accessRequest.requestingProvider;
accessNotification.success = success;
emit(accessNotification);
return assetRegistry.update(accessRequest.fileAccess);
};

/** A transaction processor function description
 * @param {org.acme.biznet.AccessGrant} accessGrant - the access to be granted
 * @transaction
 */
function handleAccessGrant(accessGrant) {
    var fileOwner = accessGrant.fileAccess.fileOwner;
    var currentParticipant = getCurrentParticipant();
    if (currentParticipant.getFullyQualifiedIdentifier() !== fileOwner.getFullyQualifiedIdentifier()) {
        throw new Error('Access can only be granted to this file by the owner of this file');
        return;
    }
    accessGrant.fileAccess.accessList.push(accessGrant.accessEntity);
    return getAssetRegistry('org.acme.biznet.Access').then(function (assetRegistry) {
        return assetRegistry.update(accessGrant.fileAccess);
    });
}

/** A transaction processor function description
 * @param {org.acme.biznet.AccessBlock} accessBlock - the access asset to be blocked
 * @transaction
 */
function handleAccessBlock(accessBlock) {
    var fileOwner = accessBlock.fileAccess.fileOwner;
    var currentParticipant = getCurrentParticipant();
    if (currentParticipant.getFullyQualifiedIdentifier() !== fileOwner.getFullyQualifiedIdentifier()) {
        throw new Error('Access can only be blocked to this file by the owner of this file');
        return;
    }
    accessBlock.fileAccess.blockedList.push(accessBlock.blockedEntity);
    return getAssetRegistry('org.acme.biznet.Access').then(function (assetRegistry) {
        return assetRegistry.update(accessBlock.fileAccess);
    });
}

/** A transaction processor function description
 * Only provider admins can add or revoke roles to and from providers
 */
```
* @param {org.acme.biznet.AddRoleOrg} addingRoleOrg—the role org transaction to be added to a provider
* @transaction
*/

function handleAddingRoleOrg(addingRoleOrg) {
    var me = getCurrentParticipant();
    console.log('@debug: **** ADDING ROLEORG: ' + me.getIdentifier() + ' adding access of ' + addingRoleOrg.provider + ' to ' + addingRoleOrg.roleOrg);

    if(!me) {
        throw new Error('A participant/certificate mapping does not exist.');
    }
    if (me.getFullyQualifiedType() != "org.acme.biznet.ProviderAdmin") {
        console.log('@debug: requesting provider: ' + me.getIdentifier() + ' is not a provider admin, denying transaction');
        throw new Error('Current participant is not an admin, cannot add or revoke roles to/from providers.');</n
    return
}

addingRoleOrg.provider.roleOrgs.push(addingRoleOrg.roleOrg);
addingRoleOrg.roleOrg.providers.push(addingRoleOrg.provider);

var updateList1 = getParticipantRegistry('org.acme.biznet.Provider').then(function (participantRegistry) {
    return participantRegistry.update(addingRoleOrg.provider);
});

var updateList2 = getParticipantRegistry('org.acme.biznet.RoleOrg').then(function (participantRegistry) {
    return participantRegistry.update(addingRoleOrg.roleOrg);
});

return updateList1 && updateList2;

/** A transaction processor function description
 * Only provider admins can add or revoke roles to and from providers
 * @param {org.acme.biznet.RevokeRoleOrg} revokingRoleOrg—the role org transaction to be revoked from a provider
 * @transaction
 */

function handleRevokingRoleOrg(revokingRoleOrg) {
    var me = getCurrentParticipant();
    console.log('@debug: **** REVOKING ROLEORG: ' + me.getIdentifier() + ' revoking access of ' + revokingRoleOrg.provider + ' from ' + revokingRoleOrg.roleOrg);

    if(!me) {
        throw new Error('A participant/certificate mapping does not exist.');
    }
    if (me.getFullyQualifiedType() != "org.acme.biznet.ProviderAdmin") {
        console.log('@debug: requesting provider: ' + me.getIdentifier() + ' is not a provider admin, denying transaction');
    }
throw new Error('Current participant is not an admin, cannot add or revoke roles to/from providers.');
return

var index = revokingRoleOrg.provider.roleOrgs.indexOf(revokingRoleOrg.roleOrg);
if (index !== -1) {
    revokingRoleOrg.provider.roleOrgs.splice(index, 1);
}

var provIndex = revokingRoleOrg.roleOrg.providers.indexOf(revokingRoleOrg.provider);
if (provIndex !== -1) {
    revokingRoleOrg.roleOrg.providers.splice(provIndex, 1);
}

var updateList1 = getParticipantRegistry('org.acme.biznet.Provider')
    .then(function (participantRegistry) {
        return participantRegistry.update(revokingRoleOrg.provider);
    });

var updateList2 = getParticipantRegistry('org.acme.biznet.RoleOrg')
    .then(function (participantRegistry) {
        return participantRegistry.update(revokingRoleOrg.roleOrg);
    });

return updateList1 && updateList2;